Dr. B.R. AMBEDKAR
OPEN UNIVERSITY

MBA-6
Information Management and Computers

CONTENTS

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

Course Design Committee

Prof. G. Ram Reddy *(Chairman)*
Vice-Chancellor
IGNOU

Prof. B.S. Sharma *(Convener)*
Pro-Vice-Chancellor
IGNOU

Prof. Ashok Chandra
Educational Adviser (Technical)
Department of Education
Ministry of Human Resource Development

Mr. V.K. Dar
Secretary
Department of Public Enterprises

Mr. R.S. Gupta
National Productivity Council

Mr. S.P. Singh
Director-Personnel
Bharat Heavy Electricals Ltd.

Prof. A. Dasgupta
Professor Emeritus
Faculty of Management Studies
University of Delhi

Prof. R.N. Haldipur
Director
Institute of Rural Management

Prof. M.V. Pylee
Director General, AIDE

Dr. (Mrs.) V. Patil
Director
Jamnalal Bajaj Institute of Management

Prof. M.M. Anand
Dean
Faculty of Management Studies
University of Delhi

Mr. P.S. Chaudhri
Member Staff
Railway Board

Dr. Ashok Ganguli
Chairman
Hindustan Lever Ltd.

Prof. J. Philip
Director
Indian Institute of Management
Bangalore

Prof. Y. Ranga Rao
Department of Commerce and Management Studies
Andhra University

Prof. D.R. Prasad
Osmania University

Course Preparation Team

Prof. S. Ankolekar
Indian Institute of Management
Ahmedabad.

Dr. Utpal K. Banerjee
All India Management Association
NewDelhi

Prof. S.C. Bhatnagar
Indian Institute of Management
Ahmedabad

Prof. M.P. Gupta
University of Delhi

Prof. B.H. Jajoo
Indian Institute of Management
Ahmedabad

Dr. P. Kumar
British Council Division
New Delhi

Prof. N.R. Patel
Indian Institute of Management
Ahmedabad

Prof. T.P. Rama Rao
Indian Institute of Management
Ahmedabad

Dr. D.K. Banwet
Indian Institute of Technology, Delhi

Prof. K.V. Ramani
Indian Institute of Management, Ahmedabad

Prof. B.S. Sharma
Pro-Vice Chancellor
IGNOU

Prof. Rakesh Khurana
Director
School of Management Studies, IGNOU

Dr. B.B. Khanna
School of Management Studies, IGNOU

Dr. G. Sambasiva Rao
Editor, IGNOU

Mr. Anil Yadav
School of Management Studies, IGNOU

Dr. Arun Kanda
Department of Technical Engineering
IIT, New Delhi.

Prof. Vijay B. Aggarwal
Department of Computer Science
University of Delhi

Dr. Lalit M. Johri
Faculty of Management Studies
University of Delhi

BLOCK 1 COMPUTERS IN MANAGEMENT

This block attempts to give you an understanding of a computer and its diverse applications in management.

First unit introduces you to the role of computer-aided decision-making in management of an organisation through proper handling of information system, and its advantages and limitations.

Unit 2 discusses the evolution of computers and then its components i.e., Hardware and Software. It also gives an idea about the classification of computers.

Unit 3 deals with phenomenal growth of micro-computers. It is followed by an explanation of a typical micro-computer, its hardware and the most common personal computer applications. Major types of software used for management applications are then described.

Fourth and the last unit of this block begins with a description of the basic concept of a Spreadsheet. It then discusses features of a specific Spreadsheet package called LOTUS 1-2-3 and how problems can be solved/modelled using this software package. The capability of modelling and sensitivity analysis are exemplified through an illustration. The major features are listed and a few important ones are explained. Some advantages of using Spreadsheet package are then highlighted.

UNIT 1 ROLE OF COMPUTERS IN MANAGEMENT

Objectives

After going through this unit you should be able to:

- appreciate the significance of computerisation for efficient management decision-making at the corporate level
- understand the role of Computer-aided decision-making in management of an organisation and advantages and limitations thereof.

Structure

1.1 INTRODUCTION

Prof. H.A. Simon views the computer as the fourth great breakthrough in history to aid man in his thinking process and decision-making ability. The first was the invention of writing which gave man a memory in performing mental tasks. The remaining two events prior to the computer were the devising of the Arabic number system with its zero and positional notation, and the invention of analytic geometry and calculus, which permitted the solution of complex problems in scientific theory. Now the electronic digital computers combine the advantages and attributes of all these breakthroughs and make them available for decision-making and management of organisations.

1.2 NEED OF INFORMATION HANDLING

Management Information System (MIS) can be defined, according to Joel E. Ross, as a communication process wherein information (input) is recorded, stored, processed and retrieved for decisions (output) regarding the managerial process of planning, organising and controlling. If we now define decision-making as the process of selecting from among alternatives a course of action to achieve an objective, the link between information and decision becomes clear. Indeed, decision-making and information processing are so inter-dependent that they become inseparable, if not identical, in practice.

Computerised MIS cannot technically make a decision but it can yield processed data and follow instructions to the extent of its capacity. For example, the computer can be properly instructed to compare inventory levels with programmed decision-rules on re-order level and re-order quantity, and generate purchase requisition, purchase enquiry and purchase order. This can resemble an automatic control of purchase documents, as being done for TISCO (Tata Iron and Steel Co. Ltd.) over a long period.

The modern role of MIS for managerial decision-making in a complex organisation has been compared to that of a military commander. Commanders often adopt a strategy built by direct observation of partial situations. This is the style used by the managers who track operations by periodic communications with remote sales depots, plant divisions and other offices. For instance, the central marketing organisation of the Steel Authority of India Limited (SAIL) has to keep track of around 50 sales depots spread all over India for marketing decision-making.

1.3 LEVELS OF INFORMATION HANDLING

In a modern complex organisation, the levels of information handling can be divided as decision support system, management information system, transaction processing system, and office (and other) automation system.

At the apex, the top level managers may need decision support system (DSS). This would be an inter-active system that provides the user-manager with easy access to decision-making models and data in order to support semi-structured and non-structured decision-making tasks. Inputs for DSS can be some processed data, and mostly management-originated data along with some unique models. The DSS would involve queries and responses, operations research models, and simulation. The output from DSS would be special reports to resolve difficult questions and replies to management queries.

At the middle management level (if there exists one), MIS would deal with an organised set of procedures to provide information for middle managers to support their operations and decision-making within the organisation. At this level, inputs for MIS would be both processed and raw-data and some management-originated data, along with preprogrammed models. The MIS process would involve report generation data management, simple models and statistical methods. The outputs from MIS would be filtered and screened for semi-routine decisions and replies to simple management queries.

At the shop-floor management level, transaction processing system (TPS) is a computer-based system that would capture, classify, store, maintain, update and retrieve simple transaction data for record keeping and for feeding MIS and DSS. The TPS would have transaction data as inputs. The processing for TPS would involve classification, codification, sorting, merging, adding, deleting and updating. Outputs for TPS would be detailed reports relating to routine decisions and processed data.

At the clerical level, office and other automation control system can be in operation. Office automation system (OAS) is simple in an automated office having multiple functions, where the integrated and computer-aided system allows many office activities to be performed with electronic equipment. The OAS would have inputs such as appointments, documents, addresses, etc. The OAS processing would be scheduling word-processor, data storage and retrieval. Outputs from OAS would be schedules, memoranda, bulk mail and administrative reports.

Here we would mainly be concerned with MIS at the corporate level.

Activity A

Prepare a brief report on the extent of Decision Suport, Transaction Processing and Office Automation Systems, prevailing in your organisation, in context of the description given above.

1.4 ADVANTAGES OF COMPUTERISATION

The advantages associated with computer-based managerial decision-making can be the following: (1) response time is greatly reduced; (2) very large data are stored for information and decision-making; (3) accuracy of information is considerably improved, thereby improving the quality of the decision; (4) problems are handled more easily by using various operation research models; (5) the cost involved in the decision-making process is reduced; (6) more secrecy is observed as compared to manual file system; (7) ability to take quick decision improves considerably as the time for retrieval of information is very fast; (8) paper work is reduced to the minimum as all the information is stored in the computer itself; (9) lots of information are stored for future reference; (10) chances of leakage of clasified information are reduced; (11) accuracy in manipulation is increased very much; and (12) time spent in various decision-making activities is reduced to a minimum.

Emanating from the above, the following benefits for a commercial organisation can be attributed to computerisation: (1) the availability of accurate forecasts within 1 per cent of net income; (2) the preparation of short-term profit plans and long-term projection; (3) the provision of pre-plan information in budget preparation; (4) the calculation of variances between budgeted and actual results; (5) the triggering of revised forecasts if not proceeding in accordance with plans; (6) the early warning system for monitoring activities and the signalling of necessary reactive plans; (7) the indication of income and cash flow by following alternate investment strategies; (8) the assistance to the planning of new facilities and a host of special strides; and (9) the accomplishment of the preceding items at a great speed.

While TPS has been in use over several decades, OAS is coming into practice only now in a number of organisations. The TPS has brought its own benefits for speedy execution, accurate performance and quite often confidential handling. Such benefits will become evident if one considers a couple of very common TPS applications.

The first is examination result processing which the bulk of Indian universities are doing on computer today, either with inhouse systems or with hired service bureaus. The massive nature of such processing can be visualised by looking at one State alone, namely, U.P., where 13 lakh candidates go through high school stream and 6 lakhs through the intermediate stream in any single year. The processing and publication of their results in time would not have been possible without computerisation. Besides, it is possible to maintain some confidentiality on computer processing. Another application is computerised electricity billing adopted by several State Electricity Boards in India. Under the computerised system, every meter for light and fan, or for power is invariably supported by a billing raised by computer and every such bill is again invariably despatched by computer centre. Both these actions guarantee improvement over the manual system where there is usually little certainty of bills being raised or being actually despatched due to adoption of foul means. In a single State undertaking like SPSEB it was estimated that the computerisation in the Western U.P. district for electricity billing had resulted in a considerable amount of additional revenue.

Advantages of MIS can be manifold bacause of the aid to higher level decision-making. Once the planning, monitoring, reviewing and control process are facilitated, the benefits can literally multiply several times, over and above the mere shop-floor or clerical TPS applications.

Activity B

Based on the various uses of transaction processing system in different organisations for speedy and accurate execution, suggest some other uses, which may be taken up by your organisation.

....

....

....

....

....

1.5 APPROACH TO COMPUTERISATION

The first important stage to organising MIS at the corporate level is to build up comprehensive data-base from the clerical systems. Valid data should be initially classified and codes attached to each data-set. Thereafter data-base should be constantly updated. The analogy to a reference library system is almost uncanny, where books have to be classified according to the subjects (e.g. reference, economics, management, etc.) and then codes attached to each book (e.g. 001 for reference, 338 for economics, 658 for management, etc.). Thereafter the books need constant updating through cataloguing and indexing. A library, however, is not as amenable to easy cross-reference among a vast number of books, as a computerised data-base is. With classification, codification and updating, a computerised data-base can help the user with almost instant retrieval of any amount of cross-classified and cross-revised data, thus helping tremendously the decision-making process.

The second important stage to MIS at corporate level is to decide on the principles of evaluating the raw data for decision-making. For this purpose, the four principles that can be unhesitatingly recommended are: **selection, pattern, linkage** and **overview**. The first principle of selection looks at a screened segment of data which can focus attention on variances from standards, deviations from norms, fluctuations from targets and differences from budgets. It is presumed that whatever data are related to the initially fixed standards, norms, targets and budgets they are, to that extent, not required to be looked at any further. But whatever are not conforming to the steady state are worth looking at for decision-making purposes. The second principle of **pattern** is to look at the collection of data and to derive insight by virtue of management ratios, trends, correlations and forecasts. Essentially this is a principle of gaining insight into the given mass of data. The third principle of **linkage** is a way of looking at a number of widely dispersed data-sets and to formulate a coherent picture. The last principle of **overview** is to derive a total picture which cuts across a number of control parameters and sums up the managerial position.

The third stage of MIS at corporate level is to realise the above four principles in actual practice. The first principle of **selection** can be implemented by generating exception-based reports. This requires the safe-keeping of classified, codified and updated data on the computer and retrieving only specially meaningful reports on the basis of exception. The second principle of **pattern** can be implemented by using mathematical modelling and statistical analysis. Such analytical approach requires the data-sets to be treated with mathematical models and statistical methods in order to derive meaningful indicators for decision-making.

The third principle of **linkage** can be implemented by inter-relating different data-sets from disparate files or data-bases. The inter-relationships would provide again available insight across the board. The fourth principle of **overview** can be implemented by aggregating data. Such a process of aggregation can connect together the classified and codified data for purposes of deriving a managerial insight into the total span of operations.

1.6. STRATEGIC ISSUES OF COMPUTER-AIDED DECISION-MAKING

Transaction processing system using computers have played a relatively limited role as a management tool. This has been so because decision-making has not been their central theme. Instead, they have been **specially-oriented** for on-going clerical needs in personnel (pay roll), book-keeping (accounting), technical data (capital projects) or specific functional areas (materials). Alternatively, they have been **project-oriented,** used to manage a specific programme of limited time and scope, such as, examination result processing, or, they have been **problem-oriented** for emergency and random retrieval of information to meet a crisis situation of limited duration and scope, such as, coal-blend analysis for the incoming coal wagons with varying ash and volatile content for all the steel plants.

According to Robert Anderson, the coporate MIS should assist such material functions as (a) manufacturing, marketing and other real-time operations; (b) futuristic improvement and problem-solving, instead of historical reports of the past actions; (c) necessary corrective action rather than book-keeping; and (d) monitoring of outside conditions affecting the organisational plans.

Joel E. Ross identifies the reasons for corporate MIS as the same for planning in general. It should offset uncertainty, improve economy of operations, focus on the objectives and provide a device for control of operations. Such an approach is radically different from the patch-work approach of the transaction processing system. What follows is identification of some of the strategic issues identified by Ross and others, and their suggested solutions in the Indian context.

Communication Gap

One of the reasons for the over-emphasis on the transaction processing system is the communication gap between the computer professional and the user-manager of the system. In India, far too many organisations have become used to separate EDP departments, now increasingly called computer services departments. According to Robert Townsend, the computer professionals keep on "trying to make it look tough, which is not easy. They are building a mystique priesthood, a trend of mumbo-jumbo ritual to keep you from knowing what they are doing". Because of the training interest and peer pressure, Ross suggests that there is a compulsive tendency for the computer professional to generate massive data-bases, to install display devices and glittering data-communications techniques, and to install newer and grander design. This phenomenon is quite common in India and only serves the purpose of empirebuilding and not improved management.

There is a familiar situation where the computer professional is engaged in developing the computer-aided decision-making but is not able to communicate to the user-manager. The information that he needs is called for, but the user cannot adequately express them as he has not been accustomed to a rigorous self-analysis. Thereafter, the computer professional works out a plan based on his own understanding of the user-needs, to convert them into the flow-charts and programming. In the process, the information needs themselves get altered. When the programmer codifies and implements the system, his own interpretation gets incorporated, thus further changing the user-needs. All these end up by frustrating the user-manager. This can be called "ten-minute syndrome" where sufficient time has not been spent between the user-manager and the computer professional to get all the needs clearly conveyed and understood.

A situation arose where during examination processing grace marks had to be allocated by way of moderation. Computer professionals allocated grace marks to all students which resulted in glaring anomalies where some top ranking students secured more than 100 per cent marks by virtue of additional grace marks. Obviously, the Controller of Examination had not explained properly the mystique of grace marks to the computer professionals!

Reliance on Service Bureaus or Computer Vendors

Quite often, Indian user-manager is approached by computer vendor who brain washes the management into buying his system, indicating that the system has all the solutions to the managerial problems. The end-result is that either the user gets a system which is too large for him with a lot of computer "fat" or he gets inadequate computing power for his needs.

Ross suggests that there should not be any technical romance with the computer vendor but a return on investment (ROI) approach to expenditure. Further, the user-manager should operate with master plan, rather than react to the vendor's suggestions. There have been cases where an organisation had appointed a service bureau for a large sum of money to develop a corporate MIS. After spending an year as well as a couple of lakhs of rupees, the user-organisation was thoroughly dissatisfied with the recommendations of the service bureau and did not implement it.

Lack of Master Plan

The bulk of computer failures are due to the lack of master plans to which hardware acquisition, software development and individual MIS design can be related. Without such a plan, "islands of mechanisation" result with little integration between separate systems.

The TISCO studied the interface of various systems like production planning and control system, financial control system, and sales invoicing and order processing system. It was observed that if individual systems were developed without regard to their mutual interfaces, the result would be an absence of communication between the systems and the incompatibility of the systems would prevail throughout the company. This was prevented by building up sufficient linkages among these systems and developing an integrated approach according to a master plan. A similar approach was also adopted by TELCO with encouraging results.

Organisation of the MIS Function

Since clerical systems came first involving accounting, pay roll, inventory returns and similar financial jobs, the transaction processing system developed around all of them. Following the normal principle of assigning a service activity by "familiarity", the historical trend in India has been to assign the computer to the Controller of Finance or Chief Accountant. This has been the case in the TISCO, TELCO, Indian Railways and elsewhere. Only now the situation is being reversed and in HAL, BHEL etc., MIS function has been placed under the user-manager.

With more distributed processing becoming possible, the trend has been to place computer-aided decision-making where it belongs, mainly under the user-manager with his own computing power. Already, the personal computers (PCs) have made this trend possible in practice, with individual data-bases available to the users. Similarly, terminals are available to most important users to share central computing power. In both cases, all PCs as well as terminals, the control of the computer-aided activity has to remain with the user-manager.

Lack of Good Management System

It is imperative for successful corporate MIS on computer that there is good planning and control within the framework of an efficient organisational structure. No degree of sophistication with computers can cure the basic ill of chaotic data management.

There have been many organisations where computerisation has not brought any tangible improvements because there has been no systematic handling of data or attention paid to the data management. In such cases, there would have been considerable gain by first conducting a good Organisation and Method (O&M) study. MIS has to be built on top by a management system which should include the organisational arrangements, the structure and procedures for adequate planning and control, the clear establishment of objective, and all other manifestations of good organisation in management.

It is interesting to note that good computer professionals know their craft but are simply not oriented to managerial jobs. In other words, the broad-based skills, which are necessary to function both in the computer room and in meeting with user-manager for the MIS, are conspicuous by their absence. This phenomenon has been known globally and that is why computer professionals are often called 'machine-mesmerised', where they are more loyal to their profession than to their organisation!

Managerial Participation

The single most critical problem in effective computer utilisation is the need for understanding and support from top management. In a State like Andhra Pradesh, the Chief Minister's personal encouragement has paved the way to bring computerised MIS gradually to all the districts, apart from their wide-spread use in State departments and in the Secretariat.

Even after top management support is ensured, it is necessary that there is user participation in the design phase on corporate MIS so as to avoid subsequent extensive and time-consuming re-work. This can be called "overnight syndrome" where users spell out their needs and expect the computer professionals to deliver the outputs immediately thereafter. Converting jobs eventually for computerisation needs a stabilisation period, which is all too easily forgotten.

An example of the above is the case of Uttar Pradesh State Electricity Board which went for computerised electricity billing for the Western districts. In the design phase, the Board authorities dictated that only Agra and Ghaziabad should have a pilot project before its extension to all other districts. Also, manual financial ledgers would continue as a parallel run alongwith the computerised financial ledgers for at least two billing cycles so as to generate enough confidence in the computerised ledgers. The result was a smooth introduction of computerisation. On the other hand, another State Electricity Board went through a similar computerisation process without the benefit of any pilot project or parallel run on the grounds that the results were needed fast and they were spending money on computers in any case. The consequence was a disaster when many erroneous bills came and in some cases none at all. There were questions in the Parliament and an enquiry committee under the Central Electricity Authority had to investigate the mistakes. The result was to cancel the existing computerisation, to start de novo, after having spent several lakhs of rupees which went waste.

It makes good sense, as has been experienced in the NTPC, when the user-manager picks up a minimum familiarity with the MIS at the beginning. From the point of view of the organisation, corporate MIS is as much a vital part of the operation as marketing operations and finance are today. Indian Airlines, too, discovered that managers had to be involved in order to get better and more effective information systems by virtue of their participation. A similar approach is being followed in LIC.

Failure to Identify Information Needs

A clear identification of information needs is fundamental and necessary to go for design of a corporate MIS. Recently, a Central Government department spent lavish sums on hardware and software to perpetuate the existing 53 MIS reports and to build a sophisticated data-bank without first determining the real information needs of management. It is often forgotten that only that information should go into the corporate MIS which can increase the perception of managers in critical areas such as problems, alternatives, opportunities and plans.

At least there have been two States in India which built up comprehensive land-record data-banks without examining what the user needed from these data-banks. The enormous costs for building their data-banks were later found to be unjustified and both these States have since discontinued their efforts in this area. On the other hand. another State (Uttar Pradesh) has begun on a limited basis to build up corporate MIS for 200 parameters for a couple of districts and, after ascertaining the user-needs, it is proposed to extend the MIS further to all the districts.

It is the user-manager who is to provide the specification for what he wants out of his corporate MIS. If the manager fails to do so, the computer professional by default would provide his own objectives and his own information needs. These would seldom meet the needs of the user-manager.

Poor Systems Prior to Computerisation

It has been observed that computerisations of a poor system will merely increase inefficiency at an accelerating rate. The user-manager gets irrelevant or bad information faster and the bad decisions are made sooner!

Hindustan Zinc Ltd., for instance, planned to upgrade and improve their transaction-processing system in a methodical manner. Such clerical systems as ledger accounting were to be upgraded to financial planning; invoicing to sales analysis; inventory accounting to inventory management; and production reports to production planning and control. Well established procedures helped them to make a smooth transition.

Overlooking Human Acceptance

A new MIS quite often meets resistance from the user-organisation because people do not accept what they do not understand. Such reasons for resistance have to be analysed and a new attitude brought into overcome it. Ross identified the reasons as threat to the status of the salesmen; threat to the ego of the managers; economic threat to the clerical persons (fear of job loss), insecurity for the managers having personal powers and political base; loss of autonomy and control for the production managers and engineers; and frayed and inter-personal relations for all others.

A number of public and private sector organisations such as BHEL, Hindustan Zinc Ltd., NTPC, Bhadrachalam Paper Board, etc. have started a process of systematic programme of training and user-education. It is imperative that such education begins at the top level for computer appreciation, at the middle management for specific computer applications in their own domains, and at the working level for direct involvement in input and output quality control. It is good to see the bulk of Indian organisations going through such an elaborate process of computer initiation as there is no short-cut to it.

Activity C

Go through the text in para 1.6, recall some of your own experiences and comments based thereon.

....

....

....

....

....

....

....

....

....

....

1.7. SUMMARY

We have given a brief account of a purposive way of using computer as a management tool. There are two eloquent reasons why such a tool is more feasible to operate today than ever before. The first reason is that computing power in any organisation is becoming increasingly available on a distributed basis through an array of terminals and/or through desk-top computers available for personal use. Availability of both terminals and personal computer is almost a comon phenomenon today.

The second reason is that the computers were used to be driven earlier by the professionals and were housed exclusively in the EDP departments. From that atmosphere, the computers have currently become very much user-friendly when the barriers of EDP department have broken down and computer languages are becoming more and more English-like. From this stage on, it is very likely that computers would show a trend to become user-driven. Thus, the transition from professional-driven to user-friendly to user-driven computers is a near reality now!

Further we have described some of the strategic issues which have bedevilled introduction of computer-aided decision-making in the forms of corporate MIS in many organisations in India and abroad. Some general principles have been used to illustrate the situation and a number of Indian examples have been cited by way of deriving appropriate lessons. It is vital to keep reviewing the situation so that such strategic issues are not lost sight of in the country's current emphasis on computerisation. It is important to avoid frustration before one aims at achieving elation on successful computerisation!

1.8 SELF-ASSESSMENT EXERCISES

1. Employees in modern Organisation are-called 'knowledge workers'. Justify.

2. Handling of information provides the basic ingredient in the decision-making process. Explain.

UNIT 2 INTRODUCTION TO COMPUTERS

Objectives

The objectives of this unit are to understand:

- initially what actually a computer is and its evolution
- two major components of a computer system viz., Hardware and Software
- classification of computers.

Structure

2.1 EVOLUTION OF COMPUTERS

Computers have brought about a revolution in the field of computing as a result of technological advancement in the field of electronics.

"Abacus", the first mathematical device used to facilitate arithmetical computation, was invented by the ancient Chinese before the birth of Christ. Abacus used beads strung of wires to aid arithmetical computations and is believed to have been in use till recently.

The first mechanical 'Computer' called 'Analytical Engine' designed by Charles Babbage betwee. 1830 and 1850 marked the birth of computer age. This was the first mechanical computer capable of performing basic arithmetical functions. Charles Babbage designed his analytical engine around five components, namely,

a STORE to hold numbers,

an ARITHMETIC UNIT (which he called the 'Mill') to perform arithmetical operations,

a CONTROL UNIT to control and coordinate various activities in the correct sequence,

an INPUT device to transfer both numbers and instructions into the computer, and

an OUTPUT device to display the results of computations.

Remarkably even today's computers are built around the same concepts. However, modern electronic computers provide three major **advantages:**

- enabling the computer to operate at electronic speeds (an electron travels approximately 1 foot in 1 billionth of a second),
- providing tremendous reliability,
- making the computer a general purpose machine.

The first commercial computer called UNIVAC (Universal Automatic Computer) using vacuum tube circuits was used by the U.S. Census Bureau in 1951. Computers using vacuum tube circuits belonged to **First Generation Computers.** Invention of the transistor by Bell Labs in U.S.A. gave birth to **Second Generation Computers** which used transistors instead of vacuum tubes. An example of a Second Generation Computer is the IBM 1401. **Third Generation Computers** were introduced in the mid-1960s with the invention of tiny integrated circuits on silicon chips. IBM released its 360 series computers with integrated chips of 28/1000 each sq. carrying more than 1000 circuits. Large scale integrated chips became available in 1969. Computers which used large scale integrated chips can be be called **Fourth Generation Computers** even though there is no general agreement on what constitutes a Fourth Generation Computer.

Invention of the micro-processor in 1972 has changed the computing scene dramatically. A micro-processor when interfaced with memory and input/output units becomes a micro-computer. A micro-computer is very small in size (can be actually placed on a table top) but is very powerful and provides a user-friendly atmosphere for managers to interact with it. The first business micro-computer called APPLE II was released in U.S.A. in 1977. A micro-computer which is reasonably powerful can be bought today for as low as Rs. 20,000.

Recent advances in the field of artificial intelligence are being used to design a "thinking computer". These computers can be called **Fifth Generation Computers.** A special feature anticipated in these computers is their ability to respond to spoken words. These computers are expected to be in the market by the end of this decade.

Activity A

Describe briefly 'generation of computers'. Find out which generation of computer, your organisation has, in case these computers are there in your organisation.

....
....
....
....
....
....
....
....
....
....
....
....
....

2.2. COMPUTER HARDWARE

Physical units of computer system constitutes a computer's hardware. When you look at a computer system you are actually looking at the computer hardware.

A computer, like the human brain, receives information, stores it, processes it, and displays results. A computer receives information from **input devices,** stores it in **memory,** processes this information in the **central processing unit** and displays the results of processing in a useful form through **output** devices. A computer's memory like the human's is limited. Hence it stores most critical information in its **main memory** and less critical and less frequently used information in its **secondary memory.** Below we describe each of these parts in detail.

Central Processing Unit

The Central processing unit is the most important component of a computer's hardware. It could even be called 'the computer' by itself. It has an **arithmetic logic unit** (ALU) and a **control unit.**

The ALU, as the name indicates, performs all the arithmetical and logical operations. Examples of arithmetical operations are: addition, division, multiplication, etc. Examples of logical operations are: (1) Is A = B, where A and B are both numeric/alphanumeric? (2) Is a given character equal to M for male or F for female?, etc. All the arithmetical and logical operations are performed in the CPU in special storage areas called **registers.** The size of the register is a very important consideration in determining the speed of processing. Register size refers to the amount of information that can be held in a register at a time for processing. The larger the register size, the faster will be the speed of processing. A CPU's processing power is measured in million instructions per second (**MIPS**). The speed of CPU was measured in milliseconds (one 1000th of a second) on first generation computers, in micro-seconds (one millionth of a second) on second generation computers, in nano-seconds (one billionth of a second) on third and fourth generation computers, and is expected to be measured in pico-seconds (one 1000th of a nano-second) in the future fifth generation computers.

The control unit, as the name indicates, control and co-ordinates all the operations of a CPU. It ensures that the required information is transferred between the main memory, the arithmetic logic unit, input devices and output devices in the required and desired sequence. It also contains the logic circuits and storage needed for the control of multiple input and output devices. When a programme begins an input operation, the control unit identifies the input devices and sets up electronic data path for the data and instructions to enter the CPU. Subsequently the control unit executes the various operations in CPU in the desired manner. Upon completion of a job, the control unit identifies the required output devices and sets up data paths for supplying the output information to the desired output devices.

Main Memory

Main memory stores a variety of critical information required for processing by the CPU. Just how does it store information?

A computer works by electricity. Hence a binary number system which uses only two digits, namely, 0 and 1 is a very convenient way to represent information inside a computer. We could use the symbol 1 to represent the presence of an electrical pulse and the symbol 0 to represent the absence of it. Information in a computer consists of data (numerial and non-numerical) and instructions which are made up of a large number of characters, namely, decimal numbers 0 to 9, alphabets A to Z, arithmetical operators like (+), (-), etc., relational operators like (=), (<), etc., and many other special characters like (,), (;), etc.

With two binary digits, we can represent four different characters, namely, 00, 10, 01 and 11. With three digits we can represent eight different characters, namely, 000, 100, 010, 001, 110, 101, 011 and 111. Computers use eight binary digits (**bits**) to represent information internally. This allows upto $2^8 = 256$ different characters to be represented uniquely. A collection of eight bits is called a **byte.** One byte is used to represent one character internally. Most computers use two bytes or four bytes to represent numbers (positive and negative) internally.

The memory unit stores all the information in memory cells, also called **memory locations,** in binary digits. Each memory location has a unique **address.** The contents of the desired memory locations are provided to the central processing unit by referring to the address of the memory location. The amount of information that can be held in the main memory is known as **memory capacity.** The capacity of the main memory is measured in kilo bytes (KB) or mega bytes (MB). One kilo byte stands for 2^{10} which is approximately 1000 bytes. A mega byte stands for 2^{20} which is approximately one million bytes.

Earlier computers used **magnetic core memory.** However all modern computers use **semi-conductor memory.** Semi-conductor memory is faster and cheaper than magnetic core memory. Semi-conductor memory is also available in a small board. These characteristics have made semi-conductor memory more popular and attractive. The only drawback of semi-conductor memory is that it is **volatile,** that is it loses its contents in the event of power failure. However, it is not a serious drawback and can be easily overcome by having back-up power units.

Semi-conductor memory is known as **RAM** (Random Access Memory). This means that any part of the memory can be accessed for reading and writing. A magnetic core memory is also a random access memory. However the terminology RAM has been used in the literature with the invention of semi-conductor memory. Another part of main memory is **ROM,** (Read Only Memory). ROM allows its contents **to be read only** and does not allow users to store their programmes. Usually ROM contains utility programmes supplied by the manufacturers frequently used by the users.

RAM capacity is a useful indicator to compare the main memory capacity of various computers. Earlier computers, provided a memory capacity of 16 KB, but a memory capacity of 512 KB is very common in today's micro-computers.

Secondary Memory

Secondary memory is essential to any computer system because of the limited main memory. There are two types of secondary memory available, **serial access memory** and **random access memory.** A serial access memory provides only a serial access to retrieve information stored in it. A random access memory provides a random access to retrieve information stored in it. It is helpful to think in terms of a cassette tape providing serial access memory and a LP record providing a random access memory. Computers use magnetic tapes to provide serial access memory and magnetic disks to provide random access memory.

Magnetic Tape

A magnetic tape is a very compact medium for storing a large amount of data. A standard tape reel is 2400 feet long, 1/2 inch wide and can store up to anywhere from 20 MB to 150 MB depending on the recording density it permits. A magnetic tape is mounted on a **tape drive** which has a read/write head.

The **density** of a magnetic tape refers to the amount of information stored in one inch of its tape length and is measured in bits per inch (bpi). Standard tapes come with 800 bpi or 1600 bpi, even though tapes with 6250 bpi are also common. A text book of 600 pages with 50 lines per page and 40 letters per line can be accommodated in less than 20 feet of a tape of density 6250 bpi. Imagine how much data can be stored in a 2400 feet tape?

The major advantages of a tape is its **economical storage** of large volumes of data and a **quick transfer** of its contents to CPU. However it has a major disadvantage namely it permits **only a serial access of data.** This is because the read/write head is fixed and immovable. As a result the tape has to be moved physically over the read/write head till we come to the desired location to access the required data.

Magnetic Disk

Magnetic disk was invented to overcome the sequential processing requirement of magnetic tape data. It is helpful to visualise a magnetic disk as a LP record. An access mechanism moves the read/write head to the desired surface on the disk to provide random access of data. **A disk pack** is a collection of individual disks stored vertically one atop the other and mounted on a **disk drive.** The disk drive rotates the disk pack at a constant speed. There is enough room in between the spinning disks to allow access arms with read/write heads to move to any storage location. Each disk in a disk pack has two recording surfaces, except the first and last disks which have only one recording surface. Thus a disk pack with 10 disks provides 18 recording surfaces. Such a disk pack also has 9 access arms. Each access arm supports two read/write heads, one read/write head to access the lower recording surface of the top disk and the other read/write head to access the upper recording surface of the bottom disk. Each recording surface is divided into a series of concentric circles called **tracks** and each track is further divided into **sectors.** The capacity of each sector whether it is on the inner most track or outer most track is a constant. One important characteristic of a disk drive is the **access time** which represents the time needed to access the desired record from a disk. Access time consists of two components, namely, **seek time** and **rotational delay.** Seek time represents the time required to seek the desired track on the desired recording surface. This involves a movement of the access arm, which supports the read/write head. Rotational delay represents the time required for the rotating disk to position the desired sector under the read/write head. Access time is measured in milli seconds. An average seek time of 25 milli seconds and an average rotational delay of 10 milli seconds resulting in an average access time of 35 milli seconds are common. Disk packs are classified into three types, namely, **removable disks, fixed disks** and **winchester disks.**

A removable disk pack, as the name indicates, can be trasported and replaced. A fixed disk pack on the other hand, cannot be removed or transported or replaced. However, it provides a quicker access time. **A disk storage capacity** of 600-1000 MB is very common. Winchester disks are a special type of hard disks and are supported mainly by micro-computers. A winchester disk provides a storage capacity in the range of 10 MB - 80 MB.

Diskettes and winchester disks provide random access memory on micro-computers. A diskette is a smaller disk of 51/4 inch in diameter. It provides a **storage capacity** of 360 KB and an **average access time** of 180 milli seconds.

Input and Output Devices

The input or output units of a computer system establish the communication between the system and its users. A large number of input and output units are available in the market. A few of them are discussed in this note.

Punched card is the most ancient (few decades old) input medium. Instruction and data punched on cards are transmitted into a computer through a **card reader.** A printed card has survived a whole century because of its simplicity. But it is on its way out now. Punched cards are being replaced by terminals which provide direct data entry.

Terminals have become very popular interactive input and output units. A terminal, when connected to a CPU, sends data and instructions directly into the computer. Terminals can be classified into two types, namely, **hard copy terminals** and **soft copy terminals.** A hard copy terminal provides a print out on paper whereas a soft copy terminal provides a visual display on a screen. A soft copy terminal is also known as a **CRT** (Cathode Ray Tube) terminal.

Key-to-tape, key-to-disk and **key-to-diskette** are stand alone data entry stations. These units usually have a small processor attached to a key board and a visual display unit. The processor checks for the accuracy of data at the time of entry. The screen displays what is being entered. These facilities are highly desirable for a data processing centre. Hence most of the data processing centres are switching over from punched card input to data entry stations.

A graphic display terminal displays information in both character and graphic forms. These are extensively used for **CAD/CAM** (computer aided design/computer aided manufacture) applications.

A plotter is used to obtain printed copies of graphic outputs. It is used by architects and designers to produce blueprints of their design on paper.

Printers are purely output devices. Printers are classified into **line printers** and **dot matrix printers.** A line printer prints one line of information simultaneously, whereas a dot matrix printer prints one character at a time. A dot matrix printer uses a 5 X 7 or 7 X 9 dot matrix formation to represent each character. Dot matrix printers therefore provide a broken appearance to the characters, whereas a line printer provides a better quality output. A line printer has a speed of about **2000 lines per minute** whereas a dot matrix printer has a lower speed of approximately **100 characters per second.**

Daisy wheel printers and **letter quality printers** are also character printers. But they produce excellent quality output. However, they are more expensive and slower than dot matrix printers. All the printers discussed so far are called impact printers as they make physical contact with paper while printing. A nonimpact printer on the other hand transfers information to paper without any physical contact. Examples of nonimpact printers are **laser printers, xerographic printers, electrostatic printers,** etc.

Most recent trend for data input is towards **source data automation.** The equipments used for source data automation capture data as a by-product of a business activity thereby completed eliminating manual input of data. Some examples are:

1. **Magnetic ink character recognition** devices are used by the banking industry to read the account numbers on cheques directly and do the necessary processing.

2. **Optical mark recognition devices** can sense marks on computer readable papers.

This kind of device is used by academic and testing institutions to grade aptitude tests where candidates mark the correct alternatives on a special sheet of paper. These answer sheets are then directly read by the optical mark recognition devices and the information sent to a computer for processing.

Activity B

Draw a flow diagram about functioning of computer Hardware.

2.3 COMPUTER SOFTWARE

Computer software consists of sets of programmed instructions which enable the hardware units to perform. Programming a computer to perform has always been a very difficult task. The first electronic computer was programmed using wired panels. Today's computers are programmed using software.

A computer software can be broadly classified into two categories - **System Software** and **Application Software.** System software is a set of instructions to the machine hardware to interpret and execute application software. An application software is a set of programming instructions for specific applications like payroll accounting, inventory control, etc. A system software consists of **language translators** (called compilers and interpreters), **operating systems, utilities** and **special purpose** software.

Language translators

A language translator is a system software which will translate a computer programme written by humans into a machine understandable form.

The most elemental form of programming uses only the binary digits 0, 1 which is directly understood by the electronic circuits. A programme written using only binary digits is called a **machine language programme.**

Assembly language provided a significant improvement over machine language. Assembly language programmes are written using mnemonic codes like ADD, STORE, etc. rather than their machine language representations in binary digits. Therefore programming in assembly language is easier. However, it needs to be translated into machine language codes. This translation is done by an **assembler.** Both machine language and assembly language programmes are **machine dependent.** This means that a programme written for one machine cannot be used in another machine. **High level languages,** which are closer to English overcame the drawback of machine dependence. A few high level languages are **FORTRAN, BASIC, PASCAL, COBOL** etc. These languages relieve the programmers from being machine specific. However a programme written in a high level language needs to be translated into machine language codes before execution. This translation is done either through **compilers** or through **interpreters.** A compiler is a translator which reads an entire programme written in a high level language and converts it into machine language codes. An interpreter on the otherhand, is a translator which interprets statement by statement, any programme written in a high level language. An interpreter is a very effective tool for programme development as it checks for errors statement by statement in an interactive mode. This allows the programmer to correct the errors statement by statement as he enters them. A compiler on the otherhand, will request the user to enter the entire programming statement and then it will check for errors. Basic language provides a compiler and an interpreter. Other languages such as Fortran, Cobol and Pascal had only compilers initially. However, interpreters for these high level languages are also becoming available, partly due to the popularity enjoyed by micro-computers in the field of computing.

Operating Systems

An operating system is the most important system software and is required to operate a computer system. An operating system manages a computer's resources very effective, takes care of scheduling multiple jobs for execution and manages the flow of data and instructions between the input/output units and the main memory.

Operating systems became a part of computer software with the second generation computers. Since then operating systems have undergone several revisions and modifications in order to achieve a better utilisation of computer resources. Advances in the field of computer hardware have also helped the development of more efficient operating systems.

The first operating system called **batch processing** (serial) operating system was developed for the second generation computers. This operating system executes jobs serially one after another from a batch of jobs submitted for execution. The central processing unit is kept busy only during the processing cycle of a job and it idles during the input and output operations.

The above drawback of idling the CPU was overcome with the introduction of **overlapped processing.** For example why not take up input operation of job 2 and the output operation of job 1 simultaneously? This and similar considerations gave rise to the concept of **multi-programming.** A multi-programming operating system handles multiple jobs simultaneously by overlapping the input, output and processing cycles of various jobs.

This operating system was introduced along with third generation computers and is still very popular. It has replaced the earlier batch processing (serial) operating system. With multi-programming a CPU's utilisation is increased and hence jobs get executed faster on an average. However a multi-programming operating system is more complex than a batch processing operating system and hence it requires more powerful hardware to support it.

Other types of operating systems which are popular today are **multi-processing** operating systems and **real time** operating systems. A multi-processing operating system uses multiple CPU's to process multiple jobs. A real time operating system is a very different type of operating system, because it is used for different type of applications, i.e., real time applications such as airlines reservations, process control, etc.

Utilities

Utility programmes are those which are very often requested by many application programmes. A few examples are:

1 SORT/MERGE for sorting large volumes of data and merging them into a single sorted list.

2 Transfer programmes for transforming contents from one medium to another, e.g., disk to tape, tape to disk, etc.

Special Purpose Software

Special purpose programmes are those intended to extend the capability of operating systems to provide specialised services to application programmes.

A few examples are:

1 Spreadsheet software like LOTUS, VISICALC etc.

2 Data Management software like dBASE III, DBMS etc.

2.4 CLASSIFICATION OF COMPUTERS

In the 1970s computers were classified into three categories, namely, super computers, large computers and mini computers. But the invention of the micro-processor which gave birth to micro-computers in the late 70s has drastically changed the computing scene. A wide range of computers are available today in the market ranging from a **personal computer to a super computer.**

A personal computer (PC) is a micro-computer which has its entire central processing unit on a single micro-processor chip. A PC provides a main memory capacity upto 640 kilo bytes and a processing power of 0.5 MIPS. It supports one of 2 diskette drives of 360 KB each for random access secondary memory. A PC usually comes with a standard key board, a visual display unit and a dot matrix printer. A PC is usually operated by a single user.

A super micro-computer has its entire central processing unit on a very powerful micro-processor, provides a main memory capacity of upto 16 mega bytes, has a processing power of 2 to 3 MIPS, supports winchester disks of capacity 20 to 80 mega bytes and allows simultaneously access to multiple users. Today's super micro is actually replacing yesterday's **mini** computers which provide almost the same computing power if not less; but based on a different technology.

A large computer provides a main memory capacity of upto 32 mega bytes, a processing power of 10 MIPS and supports a large number of disks and tapes. Disk capacities range from 100 MB to 1000 MB. It also allows simultaneous access to multiple users and supports a large number of input and output devices.

A super computer is the largest computer system available in the market. It provides a main memory capacity of upto 64 mega bytes and a processing speed of 20 MIPS. It also supports a large number of disks and tapes of enormous capacity for providing random access and serial access secondary memory. A super computer also allows multiple users to interact with the CPU simultaneously through multiple types of input devices. It also provides outputs in a variety of forms and supports a large number of output devices.

Personal computers are used by managers for their own computing needs. Availability of a large number of user-friendly software packages on PCs have made them effective tools to support managerial decision-making. PCs have also encouraged **distributed data processing.** A super micro is used for commercial data processing and managerial applications. A large computer is necessary for very large data processing and data management applications. A super computer is primarily used for highly scientific and research purposes.

2.5 SUMMARY

This unit began with a brief commentary on the evolution of computers. We then discussed in detail the two major components of a computer system, viz., **Computer Hardware** and **Computer Software.** A few important characteristics which measure the power of a computer's hardware and software have also been mentioned. Finally we provided a classification of computers ranging from a personal computer to a super computer.

2.6 SELF-STUDY EXERCISES

1 a) What is computer hardware?

b) What is computer software?

Mention two hardware and 2 Software components of a computer system.

2 Explain the following terms in one or two sentences each.

a) Central Processing Unit

b) Secondary Memory

c) Operating Systems

d) Language Translators

3 Explain the difference between Random Access Memory and Serial Access Memory.

4 Explain the advantages and disadvantages of Core Memory and ROM Memory.

5 Explain the distinction between a batch processing (serial) operating system and a multi-programming operating system.

6 What are the differences between a compiler and an interpreter?

7 Choose the most appropriate alternative for the following questions.

A With 5 bits it is possible to represent

a) 5^2 differnet symbols

b) 2^5 different symbols

c) 2 X 5 different symbols

d) 5 different symbols

e) None of the above

B Which statement is true

a) Assembly language programmes are machine independent.

b) Machine language programmes are machine independent.

c) High level languages are machine independent.

d) High level languages are machine dependent.

e) All programmes are machine independent.

C CPU fetches information from main memory by referring to the

a) Variable name

b) Variable value

c) Location address

d) Location name

e) None of the above

D Computers can be classified in the following hierarchical order.

a) PC, Super Micro, Large, Super Computer

b) PC, Large, Super Micro, Super Computer

c) large, Super Micro, Super Computer, PC

d) Super Micro, PC, Large, Super Computer

e) None of the above

Answer Key

Question Nos. A - (b), B - (c), C - (c), D - (a)

UNIT 3 PERSONAL COMPUTER AND ITS USES

Objectives

The objectives of this unit are to give you an understanding:

- of what a personal computer is and how it is used by managers
- the most common types of software used for management applications on personal computers.

Structure

3.1 INTRODUCTION

Computers were first used in management and business applications in centralised data processing departments. These departments were manned by computer specialists who were the only persons allowed access to the machines. Later terminals were provided to users for certain applications. However, cost, complexity, size and the need for special environment control did not permit computer users to have computers at their work places. All this has changed with the advent of micro-computers.

3.2 MICRO-COMPUTERS

Micro-computers are computers which have their entire Central Processing Unit (CPU) on a single integrated circuit 'chip' (called a micro-processor). Today powerful micro-computers are available which cost as little as Rs. 20,000. These micro-computers can do many of the jobs earlier done by large computers which cost many times this amount. In addition, because they are small enough to sit on a desk and do not require extensive environmental control, they can be used in most reasonably dust-free environments. Field tests on machines used in projects such as the Computer Literacy Project for Secondary Schools of the Education Ministry have shown this. This ready availability for users has enabled a large number of entirely new applications to be undertaken. It has also increased user interest in computing since users can be directly in control of their data unlike the situation with centralised computing. Due to these factors and the fact that micro-computers can be used at home and schools for education and entertainment there has been a spectacular increase in their numbers since their first arrival as hobby kits in the U.S. Several million micro-computers are used in management applications around the world today. In India the projections are that by 1995 a few millions of these machines will be installed.

3.3 HARDWARE

Micro-computers are structurally similar to large computers in that they have the same basic components - CPU main memory, secondary memory and input/output devices. The most popular secondary memory devices are floppy disk drives and fixed disks.

Floppy disks store between 140 K Bytes to 1 Mega Byte = 1024 K Bytes (MB) while hard disks store 10 MB to 40 MB. The main memory is made from several chips. These may be ROM or RAM chips. ROM (Read Only Memory) chips contain instructions permanently written in the memory. The contents of these chips cannot be altered. RAM (Random Access Memory) chips provide the memory that can be read from, written into and cleared. Most micro-computers have between 16 K bytes and 1 MB memory. The typical word-size of micro-computers when they were introduced in business in the late seventies was B-bits. Today, the most common word-size is 16-bits. It is virtually certain that in the next few years 32-bit micro-computers will be very common although today their numbers are somewhat limited due to cost and scarcity of software. Dotmatrix and daisy-wheel printers are popular printing devices on micro-computers.

With the proliferation of micro-computers, the concept of distributed computing has emerged. In distributed computing, a number of computers are dispersed in different parts of an organisation and inter-connected using special cables, telephone lines or other telecommunication media. This allows data and programmes to be shared and also provides for some new types of applications such as electronic mail. To support such networking special electronic devices such as modems and network inter-face cards are often fitted into micro-computers.

As with larger computers, micro-computers have operating systems to enable the user to easily interact with the machine. These operating systems are similar to those on large machines except that they are much simpler. This is because the majority of micro-computers are not multi-terminal machines - only one user has access to the machine at any time. In this situation the machine is the **personal computer** of the user. Personal computers are used for a variety of tasks. The most common applications are:

Word Processing
Financial Analysis
Data Base Access
Graphics
Accounting

3.4 APPLICATIONS SOFTWARE

Systems software available on micro-computers includes compilers and interpreters for popular programming languages like BASIC, COBOL and FORTRAN. However, because of their large numbers and personal use, powerful applications software which does not require significant programming knowledge has been developed to run on personal computers. The most important categories of this-software, which enables a user to autonomously utilise a personal computer are:

Data Base Management Software
Spreadsheet Software
Word Processing Software
Graphic Software
Data Communication Software

Data Base Management

Software packages called Data Base Management Systems (DBMS) had been developed long before micro-computers to handle the problems of maintaining and integrating large volumes of data on large machines. The most important distinguishing feature of a DBMS package on personal computers is that it provides a very high level language interface which can be learned by a user who is not a computer programmer. In a few hours, it is possible to acquire enough skill to use the basic features of DBMS packages like dBASE III which is the best selling software currently in this class.

The dBASE III package has attained its current popularity mainly for the following reasons:

1 It is easy to use and is simple. Its basic features can be learned in a couple of hours.

2 It provides a very high level language interface which is command oriented. Some people consider it a fourth generation language. In lay man's terms, fourth generation language means it is superior to standard programming languages like BASIC, COBOL, FOR TRAN and PASCAL; and it simplifies and sometimes even avoids programming in a conventional sense.

3 Small business information systems can be easily implemented in a few days using this package.

4 This package can be used as a tool to prototype large applications. Prototyping is useful for saving costs in implementing large applicaitons. Parts of a large system may be implemented through dBASE III package to finalise specifications of what users want through the envisaged application and how these parts could possibly be implemented in their final form.

Like other DBMS packages, dBASE III provides features for:

1 Creating data files on a computer.

2 Maintaining these data files by way of providing functions for adding, deleting, editing and updating a given set of data items.

3 Generating reports based on the data files created through dBASE III.

4. Querying on those data files.

To develop an understanding of the features provided by dBASE III package, let us take a specific application and see how it could be implemented in dBASE III. Suppose we are operating a small business house which has 1000 customers. We are interested in maintaining a list of these customers and using this list for our day-to-day business operations. Obviously we will have to create a file which contains data about these 1000 customers. In this file there will be one record for each customer. Thus, we will have a file containing 1000 records.

Bytes
↓ compose
Words
↓ compose
Fields
↓ compose
Records
↓ compse
Files

Fig. 3.4

1 (a) Describe the hardware of a typical personal computer system.

Each record will have to contain details about a customer. What constitutes 'the necessary details' will depend mostly on what use we want to make of this data-base. For the sake of simplicity, let us say we will have the following information for each customer.

1	Customer Identification code	CUSTID	(G 1029)
2	Customer Name	CUSTNAME	(V.K. GANDHI)
3	Address	ADDRESS	(NARAYAN CHAMBERS, ASHRAM ROAD)
4	City	CITY	(AHMEDABAD)
5	State	STATE	(GUJARAT)
6	Pincode	PINCODE	(380009)
7	Phone Number	PHONE	(77334)
8	Amount Due	AMOUNTDUE	(5249.25)
9	Credit limit	CRDLMT	(10000.00)

Each data item listed above will form a **field** in the record structure for the customer. When we create a customer file through dBASE III package, it will first want us to specify the record structure for the file.

For a customer file we want each record to have nine fields: one each for the items listed above. In dBASE III, it is necessary to give names to each of these nine fields so that they can be later referred to using these names. The field names which we have chosen are shown in the third column above. Apart from the name, we have to also indicate what type of data will be stored in the field and in certain cases number of characters to be reserved to hold the data for that field. We will want AMOUNTDUE and CRDLMT to hold numeric data and all other fields to hold text or character data.

Now let us have a quick look at a dialogue in dBASE III for creating and using such a file.

When we start the dBASE III programme, it prompts by dot (.). Also when it finishes executing a given command, it writes dot in the first column of next line on the screen. Interaction with dBASE III package consists mainly of issuing appropriate commands against this dot prompt.

Each dBASE III command starts with a key word (or verb) followed by certain parameters. The command for creating a file for customer data would be

..CREATE CUSTLIST

Note the keyword CREATE which is typed by the user against the dot prompt produced by dBASE III. CREATE command wants a parameter which will be used as a name for the file so created. The user can choose any name that he desires within the syntactic rules specified by the package. We shall henceforth omit such minor syntactic details of the package. We shall use the symbol ↵ to indicate enter or return key of the keyboard.

In response to CREATE command, dBASE III will provide a screen where we can enter the above nine field names alongwith their type characteristics.

Note that with this command will have only specified the structure of the record for a file which is to contain customer data. Therefore we should have some way of **loading** the customer records in the CUSTLIST file. For this purpose we will have to type the following command:

..APPEND ←

With this command we get a screen which provides a blank customer record. As the name implies, this command will append a record to our CUSTLIST file. Thus, it can be used whenever we want to add one more customer to customer list.

In the screen provided with above command, we now type the necessary details of a customer. The names of the fields are displayed on the screen to facilitate entry of data for a customer for which we wish to create a record in CUSTLIST file. After we type in all the required details about the customer and type ↵ we get another blank record in which we can enter data about next customer and so on. When we type ↵ right at the beginning of a blank record we get back to dot prompt where we can issue a new dBASE III command.

Now suppose we want to close our dBASE III session and come next day to query this customer file. We must first exit from dBASE III. The command for this is

..QUIT

Next day when we start the dBASE III programme, we must first indicate which file we want to use, otherwise dBASE III will not know which file we are interested in. The command for this is

..USE CUSTLIST

Now let us say we want to query on the customer data for extracting different types of information. Here is a sample list to illustrate how easy it is.

TYPE	TO GET
● LIST FOR AMOUNTDUE > 5000	Details of all customers for whom amount due is greater than 5000.
● LIST PHONE FOR CUSTID = 'G1029'	Phone number of a customer whose Identification number is 'G1029'
● LIST CUSTNAME, ADDRESS FOR CITY = 'BOMBAY'	Name and address of all customers from Bombay.
● LIST FOR AMOUNT DUE > CRDLMT	Details of all customers who have crossed their credit limit.
● LIST FOR 'SHAH' $ CUSTNAME	Customers who have a character string SHAH in their names. Note that this will list not only SHAH but also customers like SHAHNAZ.

We have seen that just with five commands we can do a lot with dBASE III package. Below are a few more commands which further illustrate the simplicity of dBASE III package.

TYPE	TO GET
● AVERAGE AMOUNTDUE	Will provide on the screen average amount that is due from the customers.
● COUNT FOR AMOUNT DUE > CRDLMT	Will show on screen a number indicating how many customers have crossed their credit limit.
● SUM AMOUNTDUE FOR CITY = 'BOMBAY'	Will give how much amount is due from customers from Bombay.

One has to learn only few additional features to see how sales and receipt transactions can be used to update amount due for each customer. The dBASE III package also has REPORT command for producing user specified reports.

With the command verbs, dBASE III also provides conventional programming features like IF - THEN - ELSE, WHILE - DO. A complete set of dBASE III programmes can be designed to implement a medium sized application in a couple of weeks.

Word Processing

Word Processing software is designed to enable the user to prepare typed documents. Thus, in contrast to data processing where the focus is generally on numerical data, in word processing the main concern is with text. In the early days of computing a disparaging remark that computer professionals employed to refer to an application system which did very little computing was to say that the system used the computer as an 'expensive typewriter'. With dropping costs and the increased productivity due to word-processing, today the micro-computer can be used as a 'cheap -typewriter'!

In order to prepare a document using a word-processing package, it has initially to be typed into the computer's memory from the keyboard. The main productivity improvement comes from the ease and rapidity with which the document can be modified. Only when the finished version is ready it is necessary to put it on paper. Where there are several drafts of a letter or report, or where extensive text from past reports is to be used the productivity gains can be very large. Typical situations where high gains are possible are: law firms, contractors, newspaper offices, banks and government offices. Studies in the U.S. have shown that the manpower required to prepare a document can be reduced from 40 to 80 per cent over using an ordinary typewriter. In addition, the document can be sent at electronic speeds over a local or long-distance network to other computers easily because it is in computer-readable form. Thus, word-processing provides a ready interface to electronic mail.

A typical word-processing package (WPP) has the following features:

Automatic wrap-around

With a typewriter, the typist has to watch for the end of the line and press the carriage return key to start a new line. With a WPP typist simply continues typing, the computer automatically starts a fresh line when a line is filled up. Only at the end of a paragraph, it is necessary to perform a carriage return.

Cursor Control

All WPP display a bright movable area the size of one character on the screen. This is called the cursor. It can be moved around the screen by pressing keys. The cursor enables the WPP user to identify an area of text where he wants to make changes.

Deletion

Characters, words, sentences, paragraphs and entire pages can be deleted by using the cursor with just one key-stroke. The text following a deletion will automatically get re-adjusted to fill the gap created by deletion.

Insertion

Just like deletion except that characters, words, sentences, etc. can be inserted anywhere in the text.

Replacement

Typed characters can over-write characters in document (this is like a combined deletion and insertion).

Movement of blocks

A block of coninuous text can be moved from any point to any other point in the document.

Copying of blocks

A block of continuous text can be copied from one document to another or from one part of a document to another.

Formatting

Spacing, margins, right and left justification, page numbering can be set and changed at any time.

Besides the above standard features a number of advanced features are available on sophisticated WPPs. Some of these are:

Mail Merge

A common need in many offices is to send the same letter with changes for name and address to a number of parties. This feature enables the computer to process a file of names and addresses and merge it with a standard letter creating letters for each party.

Search and Replace

A string of characters can be replaced anywhere in the document by another string of letters. This is useful, for example, when the spelling of a person's name has to be corrected in several places.

Spelling checker

The WPP can be instructed to check spellings and point out where errors may have been made. It would also suggest what the correct spelling might be. Some WPPs also enable the user to add words to its dictionary that the user commonly uses but which are not in its dictionary. Specialised technical terms are examples of this situation.

Thesaurus

The WPPs can be requested to supply words that are similar in meaning to a particular word (synonyms) in a document.

Multiple fonts

Different styles such as italics and bold as well as different sizes and shapes of characters can be typed (provided the printer has the requisite features). Mathematical symbols, Greek letters are also available.

Electronic Spreadsheets

Electronic Spreadsheet software hailed as the single most important reason for management use of micro-computers. It is doubtful if the business world would be buying micro-computers in the volume that it does if there were no electronic spreadsheet packages. The fundamental idea of value in such packages is the concept of an 'electronic' spreadsheet. A spreadsheet is simply a sheet of paper with rows and columns in which one can enter data in the form of numbers and text. A balance sheet is a spreadsheet, a price list is a spreadsheet, in fact most managerial reports are spreadsheets. An electronic spreadsheet is like a paper spreadsheet except that:

1 It is much faster and easier to make modifications to it and to make both electronic and paper copies of it.

2 At a given time one can only see a part of the whole sheet. To see parts which are not visible, we need to 'scroll in' those parts. It is like seeing the spread sheet through a window.

3 In addition to allowing numbers and text it allows formulas (such as Contribution = Revenue - Variable Cost) to be entered into the work-sheet. This enables very rapid recalculation to be done under changed assumptions.

The most important capability that a spreadsheet offers is that of a straight-forward, rapid and unobtrusive sensitivity analysis. Once a spread sheet has been set up, it is very easy to answer 'what-if' questions. For example, if one has a spreadsheet reflecting all the cash-flows and other aspects of an investment, we can readily re-compute the impact of errors in our sales prediction on the return on investment.

Lotus 1-2-3 is the most popular spreadsheet software package today. The Lotus 1-2-3 spread sheet (called a worksheet) has rows entered numbered 1,2,3 upto 2048. It has 256 columns - from A, B, C, ..., Z through AA, AB etc. upto IV. Newer versions can handle more rows and columns. The width of the columns can be different and can be adjusted to suit the application. At any time 20 rows and (typically) 8 columns are visible on the computer-screen. To view other parts of the worksheet, the 'cursor' key has to be manipulated. The 'cursor' is a bright rectangle of light which is one row by one column in size. By using up, down, left and right arrow keys the cursor can be moved around on the screen. When the cursor is at the right extreme, pressing the right arrow key will 'scroll' the worksheet one column to the right, i.e., the left column(s) will disappear and a new column (the column to the immediate right of the previous right-most visible column) will appear. By repeating this action as often as required, we can move the window to the right. Similarly, by using the other arrow keys, we can move the window left, up and down as well.

To enter a number of text, we need to move the cursor to the desired row-column position (called a 'cell'). Next, the data is entered by pressing the appropriate typewriter keys. Finally, when the 'Return' key is pressed, the data will get entered in the cell and be displayed on the screen. To enter formulas a similar procedure is followed. The only difference is that instead of data, a formula is entered. A formula indicates how the value of the chosen cell depends on other cells of the worksheet. For example, if cell A5 contains the revenue figure and cell B8 contains the variable cost, and the cursor is at cell E15, entering '+A5 - B8' will ensure that cell E15 will always show the contribution value. Lotus 1-2-3 has many for more advanced features. These will be covered in greater detail in a subsequent unit.

Spreadsheet models have a simplicity which makes them natural for users. They also find use in a surprising variety of applications. Although, undoubtedly, their largest use is in financial modelling, they find frequent use in areas like marketing, production, logistics and human resources.

Business Graphics Software

While spreadsheets and database packages are very useful in doing analysis, the output of such analysis is generally tables of numbers. The human brain is much more adept at picking up patterns from pictorial representations. If it is true, as the old adage says, that a picture is worth a thousand words, it is equally true that it is worth a thousand numbers. Computer accessible data can readily be converted to graphic form on the screen as well as on paper using dot-matrix printers or plotters. With plotters it is possible to have different colours (four colours is very common).

Typical business graphics software enables data to be plotted as:

Line Charts

Bar Charts

Pie Charts.

Sophisticated business graphics software provides for three dimensional display and maps.

Data Communications Software

In large organisations there is often a central computer for data processing which does routine data processing. Sometimes a manager would like to get some of this data for processing on his personal computer. To facilitate this, data-communications software has been developed which runs on the personal computers to make it look like a terminal to another computer. Using this 'terminal emulation' facility the manager can access data from his computer. Selected data can be 'downloaded' into the personal computer's memory and stored in files on disks. Subsequently, he can analyse these files using DBMS or spreadsheet software. An example of this kind of use would be a situation where a main-frame data processing computer has files on sales of products broken down by month and by region. Downloading on a personal computer would enable the manager to analyse up-to-date data on product movement in different markets.

Statistical Packages

A number of easy-to-use packages which run on micro-computers to perform standard statistical analysis are available. Typical capabilities are frequency distributions, crosstabulations, tests for population means and proportions, analysis of variance, contingency table tests, regression and correlation anlaysis. More advanced packages would include forecasting modules, time-series analysis and non-parametric analysis.

Operations Research Packages

Inexpensive packages are available for standard operations research models such as linear programming, critical path analysis, resource scheduling, simulation, decision free analysis and network flows. While these packages are not as powerful as software on large machines, they are far more user-friendly and cost one-tenth to one-hundredth as much. They are powerful enough to solve problems of moderate size. For example practical linear programming models with less than 100 variables and around 50 constraints can be readily solved.

Integrated Software

Software that combines several of the capabilities of packages listed above are now available. For example, LOTUS 1-2-3 version 2 combines spreadsheet, database, graphics and statistical capabilities. Another example is FRAMEWORK which combines word-processing, spread sheet, database and graphics functions. The advantage of integrated software is that the user does not have to learn to handle a variety of disparate packages so that his learning time and effort is reduced. Another advantage is that the same data can readily be passed from one function to another. The disadvantage is that such software is expensive and makes greater demands on the computer resources.

Activity A

Prepare a brief report on functioning of the computer in your organisation, mentioning various software used. Name most commonly used language and five software packages and purpose thereof.

3.5 SUMMARY

A personal computer is a micro-computer system that is available to a user for his exclusive use. The use of personal computers in management is growing by leaps and bounds. This is due to their low cost, small size, and the availability of very user-friendly software that runs on these machines. The major types of software that account for the popularity of personal computers are: word-processing, database management, electronic spread sheets, graphics, data communications, statistics, operations-research and integrated software.

3.6 SELF-STUDY EXERCISES

1 Describe the hardware of a typical personal computer system.

2 What are the reasons for popularity of micro-computers?

3 What is meant by distributed processing?

4 Describe four important management uses of personal computers.

5 How is an electronic spread sheet superior to a paper spreadsheet?

6 Describe four functions of word-processing software that would increase the productivity of a typist?

7 Explain with examples how a data base management system provides for ad-hoc queries.

8 Illustrate with examples the common types of graphics displays provided by business graphics software packages.

9 What is meant by 'down-loading' of data?

10 What is integrated software? In what way is it advantageous in use to a manager?

3.7 FURTHER READINGS

Arthur, Naiman, 1985. *Introduction to Wordstar,* Sybex Inc: USA.

Alan, Simpson, 1984. *Understanding dBASE III,* Sybex Inc: USA.

Colin, Lewis. 1984. *Managing with Micros,* (2nd Edition), *The Economist Books,* Basil Blackwell: London

Dogolite, 1984. *Using Small Business Computers,* Prentice-Hall: Engelwood-Cliff.

Gilbest, Williams, 1985. *The ABC's of 1-2-3,* Sybex Inc: USA.

Jonathan, Sachs, 1983. *Your IBM PC Made Easy,* Mc-Graw Hill: New York.

Loebbcke and Vasarhelyi, 1986. *Microcomputers - Applications to Business Problems,* Irwin: Homewood.

Note: Several Sybex Publications are published by BPB Publications, New Delhi in India.

UNIT 4 SPREADSHEET SOFTWARE AND MANAGERIAL APPLICATIONS

Objectives

The Objectives of this unit is to give you an understanding:

- of the capability of spreadsheet software and
- its use in solving several types of managerial problems.

Structure

4.1 INTRODUCTION

Availability of spreadsheet along with word processing and data base management software has been responsible for the popularity of the personal computer with managers and administrators. Managers often work with tables. Sometimes they review the data in a table like in the case of a performance report which may be sent in by a subordinate or from the field. Managers also perform analysis on tables. Examples of tables being used for analysis are profit and loss statements, balance sheets, budgets, cash flows, etc.

The concept of a spreadsheet is akin to that of a table in which row and column titles and input data may be entered and some values may be derived on the basis of other values necessitating arithmetic computations. Spreadsheet is in fact a much larger table which is held on an electronic media rather than a sheet of paper. A spreadsheet is always viewed as being made up of a number of columns and rows. The intersection of a column and a row is called a **cell.** A cell is the basic unit with which a row or column heading may be associated, a value may be input or in which a computation may be performed. In Table I a profit and loss statement is presented for a company. To a manager the table presents the level of proposed activity during the year 1986-87, revenues that would be generated, both variable and fixed costs that would be incurred and the eventual profit that is expected to be earned. The table can be viewed as consisting of 8 columns and 20 rows and it could be said that the titles for each row are stored in column A and the corresponding values are stored in column B. It may be noted that the title of the table, each row label or a corresponding value can be associated with one specific cell. For example, the label 'Sales Qty.' can be associated with cell A5 and the corresponding value of 90,000 with cell B5. Notice that a cell is addressed as a column and a row. Table I will be referenced many times in the later text. It may be useful to create an exact copy of this table in a **separate** sheet.

The profit and loss statement in Table I is a typical example of a table with which managers would work. In projecting the profitability for the next year a manager would like to work with various assumptions regarding production level, prices and unit costs. He would, in fact, like to construct a scenario in which his expected profit is as much as possible. For this purpose he may wish to determine the consequent impact on profits if the production was raised to full capacity. He may like to ask similar questions regarding price, unit cost, advertising expenditures - items which are somewhat under his control. Spreadsheet provides an easy mechanism to do such analysis which is typically known as 'what if' analysis.

TABLE 1

	A	B	C	D	E
1		ALPHA CO.			
2		PROFIT AND LOSS STATEMENT 90-91			
3					
4	REVENUE			PARAMETERS	
5	Sales Qty.	90000		CAPACITY	150000.00
6	Unit Price	4		CAP UTIL.	0.60
7	Sales value	360000		UNIT MAT. COST	1.00
8	VAR. COST			UNIT. LAB. COST	1.10
9	Materials	90000		UNIT SHIP COST	0.04
10	Labour	99000		RATE OF COMM.	0.05
11	Shipping	3600			
12	Commission	18000			
13	FIXED COST				
14	Advertising	40000			
15	Fact. Overhead	88000			
16	Adm. Exp.	22000			
17					
18	CONTRIBUTION	149400			
19	TOTAL FIXED COST	150000			
20	PROFIT/LOSS	600			

Note: It may be advisable to make a copy of this table in a separate sheet for further reference.

Spreadsheet is a software which would allow you to create tables as in Figure 1 and to manipulate numbers in these tables to perform 'what if' analysis. Several spreadsheet packages are available in the market. Some of these are VISICALC, SUPERCALC, MULTIPLAN, LOTUS 1-2-3 etc. In this unit when we discuss the mechanics of creating and working with a spreadsheet, we would do so by discussing some elements of LOTUS 1-2-3. This package is available on IBM/PC compatible computers having at least 256 KB of memory. More than a million copies of this package have been sold. LOTUS 1-2-3 allows a user to work with tables which may be as large as 2,048 rows and 256 columns. Columns are numbered from A to Z, AA to AZ and so on up to IV. The rows are numbered from 1 to 2,048. When the LOTUS 1-2-3 programme is executed on a PC the worksheet screen that appears on the monitor is shown in Figure 1. Most spreadsheet packages have similar format. Notice that there is a thick border (on the monitor it will be in reverse video) within which the column and row numbers are indicated. The area to the bottom of the border is called the worksheet area in which one or many tables may be created and manipulated. Above the border there are three lines which are called the control panel. We will first understand how to operate in the worksheet area.

Although the total worksheet area is large (256 columns and 2048 rows) the area that can be seen at any time is small. One can see 20 rows and 8 to 9 columns. Of course, the number of columns that can be seen depends on the width of the column which can be defined by a user. Thus, if the width of the column is 1 character 72 columns can be seen and if it is 72 characters only one column can be seen. The first thing is therefore to learn to see the other parts of the worksheet which are hidden at any point of time. The screen in Figure I shows the cursor which is a lighted area covering exactly one cell. Currently, it is shown covering cell B2. This cursor can be moved about by pressing certain keys in the keyboard. It is extremely important to be familiar with the key board which is shown in Figure II (See in Appendix). The keys marked with arrows when depressed will result in the movement of the cursor one cell at a time. Thus, if the right hand arrow key is depressed five times the cursor will move to cell B2. Now, if the cursor is moved once more, rather than going over the board, it will remain in the same position. However, the next column J will appear on the screen and the first column A will disappear. It is as if a window, through which the worksheet was being viewed, has been shifted one column to the right. Similar shifts of this window can be done sideways or upwards and downwards to view any part of the worksheet. Obviously, tapping the key several times to move the worksheet through large areas can be laborious. There are page up and page down keys to move 20 rows or many columns (72 characters) at a time. There is also a key home which can bring the cursor to A1 from anywhere in the worksheet.

Three types of information can be stored in any cell in a worksheet. Labels (textual information) which is normally used to indicate table titles or row and column headings. Numbers which may be input to indicate values of different items in a table and formulas which will relate the value of a cell to the data stored in other cells.

Let us understand lebels, number and formula with reference to the Table I. Table heading, company name, 'Revenue', 'Unit Price', names of various cost elements, etc. are labels. You would notice that corresponding values shown in the table are either given or derived from other values in the table. For example, the value of unit sales is given as well as the value of unit price. However, the value in cell B7 showing the sales value is derived from Sales Qty and unit price. Normally, a person may use a calculator to perform such arithmetic. However, each time the prices of the units are changed the arithmetic would have to be performed again. In a spread sheet, the user would be able to write a formula for such derived values. For example, in cell B7 the formula would be B6 * B5 expressing revenue as a quantity obtained after multiplying the value in B5 with the values in B6. When cells addressed B5 and B6 are used in a formula it is immaterial as to what is the actual number in these cells. Whatever is the number in these cells at any given time will be used to derive the value of revenue. Therefore, if the value in B5 is changed from 90,000 to 1,50,000 the sales value in cell B7 will automatically become 1,60,000.

To put labels, numbers or formula into a cell the cursor must first be positioned in the specific cell. Remember that the cursor can be posistioned by manipulating the arrow keys. After positioning the cursor, textual matter or numbers may be typed in from the key board. The use of a computer key board is just like the typewriter key board. However, when a formula is typed in, to distinguish it from a number or text it must begin by a (+) sign. When a label, a number, or a formula is typed in, it will appear on the third line of the control panel. After the return key is pressed the matter will get stored in the cell (will appear in the cell and will disappear from the third line of the control panel). The first line of control panel always indicates the cell in which the cursor is positioned and its contents. If the contents are a formula, the formula expression would be shown. However, in the cell if the contents are a formula then their computer value is shown.

Let us see how the profit and loss table can be created. First, the cursor will be moved to cell C1 and 'ALPHA CO' will be keyed in followed by a 'return' and similarly PROFIT AND LOSS STATEMENT 91-92 will be stored in cell B2. Then the cursor will be moved from cells A4 to A20 and in each cell the respective row label will the typed followed by a return. The amount of information stored in a cell can be large (upto 240 characters) and will be displayed on the screen as long as the column width is adequate or the cells in the next columns are blank.

Similarly labels will be stored in D4 to D10 for various parameters whose values will be stored in corresponding cells from E4 to E10.

The input numbers which are 'given' and which do not vary in the short-run will be typed in next. These are capacity, various elements of fixed costs and unit costs. Each number will be keyed into the cell corresponding with the label. For example, the number 1,60,000 denoting capacity will be typed in cell E5. The screen will then appear as in Figure I.

Now the formulas can be typed in to compute quantities which depend on the value of other cells. Thus the following formulas will be typed in:

B6 : + E5 * E6 Sale Qty. = Capacity X expected cap utilisation

B7 : + B5 * B6 Sales Value = Sale Qty. * Unit price

B9 : + E7 * B5 Total material cost = Unit mat cost * Sales Qty.

B10 : + E8 * B5 Total labour cost = Unit labour cost * Sales Qty.

B11 : + E9 * B5 Shipping Cost = Unit shipping cost * Sales Qty.

B12 : + E10 * B7 Total Sales Commission = Rate of Commission X Sales Value

B18 : + B7 - B9 - B10 - B11 - B12 Contribution = Revenue - Variable Cost

B19 : + B14 + B15 + B16 Total Fixed Cost

B20 : + B18 - B19 Profit = Contribution - Fixed Costs

Once the above formulas are typed in, the cells will indicate the computed value of the formulas. Now the worksheet is ready to be used for what-if analysis.

For example, the Sales Manager of Alpha believes that a 15% reduction in price will fill the plant to capacity. Is this move desirable? This can be determined by changing the contents of cell E6 to 1 (reflecting full capacity utilisation) and the contents of B6 to 4*.85 (reflecting 15% reduction in price). As soon as contents of a cell are changed, the value of all cells which depend on the value of the cell in question will be **automatically** recalculated. (Unless the user wishes not to recalculate the values). Therefore, on changing E6 and B6 to the new values B5, B7, B9, B10, B11, B12, B18, B20 will be immediately recalculated. In small worksheets the recalculation is almost instantaneous. The new profit/loss can be read off from B20. This is the fantastic power of a spreadsheet.

Another scenario which the MD of the company would like to evaluate is to increase price by 25%, increase advertising by 12,000 and boost commission to 10% of sales value. He thinks Sale Qty. will increase by 50%. Such a scenario can be evaluated by plugging in the new numbers in appropriate cells and noting the recalculated profit in B20.

Figure I

Lotus Screen for Alpha Co.

B2: PROFIT AND LOSS STATEMENT 1990-91

	A	B	C	D	E	F
1			ALPHA CO.			
2			PROFIT AND LOSS STATEMENT 1990-91			
3						
4	REVENUE			PARAMETERS		
5	Sales Qty.		XXXXXXXX	CAPACITY	150000.00	
6	Unit Price		4	CAP UTIL.	0.60	
7	Sales value		XXXXXXXX	UNIT. MAT. COST	1.00	
8	VAR. COST			UNIT. LAB. COST	1.10	
9	Materials		XXXXXXXX	UNIT. SHIP COST	0.04	
10	Labour		XXXXXXXX	RATE OF COMM.	0.05	
11	Shipping		XXXXXXXX			
12	Commission		XXXXXXXX			
13	FIXED COST					
14	Advertising		40000			
15	Fact. Overhead		88000			
16	Adm. Exp.		22000			
17						
18	CONTRIBUTION		XXXXXXXX			
19	TOTAL FIXED COST		XXXXXXXX			
20	PROFIT/LOSS		XXXXXXXX			

XXXXXXXX Cells where formulas have to be put in

4.2 MAIN FEATURES OF LOTUS 1-2-3

Ability to build fairly complex models quickly

The model discussed above was a simple one. It modelled the profits as a function of various parameters like price, unit costs, capacity utilisation. Such a worksheet could be put up in 30-45 mts whereas manual calculations or programming the problem in a conventional language like BASIC could take much longer.

The model built earlier could be further refined. Currently the worksheet does not relate sales to price, commission or advertising expenditure. The relationship must exist (as when the MD expects 50% increase in proposition 2). However, it has not been modelled explicitly.

Such a relationship could be modelled in LOTUS 1-2-3 by expressing sales qty. in cell B5 as a function of price change, advertising expenditure and commission through formulas.

Alternatively a 3-way table could represent the sales quantity as a function of the three parameters. We have not discussed how such a table could be built and used but the appropriate commands will be described in Appendix.

Functions in LOTUS

There are certain computations that need to be done frequently by managers. LOTUS 1-2-3 has therefore predefined formulas for such computations stored as functions. These functions can be used as formulas or within a formula. For example, the total of a number of cells could be computed in any cell by using a function @ SUM (Cell range and/or cell addresses). Specifically for the worksheet discussed earlier, total fixed costs in B19 could have been expressed as @SUM (B14.. B16). A number of such functions for mathematical and statistical computations, trignometric functions, financial analysis like net present value that are available are listed in Appendix.

Mention must be made of functions that provide logical capability of choosing one of many options. If the tax incidence had to be computed for Alpha company in cell B21, given a tax rate stored in E11, then the formulae in B21 could have been: @ IF (B. 20> 0, B20*E11, O). Such a function first tests a condition - in the present case the condition is: is the profit greater than Q, i.e. is it a profit or a loss. If the condition is true then the cell (where the function is used) will take the value indicated immediately after the condition. In this case B20*E11 which is the incidence of tax. If the condition is false (Alpha has made a loss) then the cell will take the value indicated by the third expression in the bracket i.e., zero in this case. LOTUS has many such functions which are also listed in the Appendix.

Sensitivity Analysis

Ability to perform what-if analysis was illustrated in the earlier example. LOTUS 1-2-3 can also perform such analysis in an automatic way. For example, a separate table may be created for capacity utilisation varying from .4 to 1.0 in steps of .1 in cells E 14 to E20. It would then be possible to instruct the LOTUS system through a command called DATA TABLE (listed in appendix) to substitute each of these values one by one in cell E6, and to read the corresponding recalculated profit from B20 and to store the value in cells F14 to F20 automatically. Notice the convenience of doing the analysis automatically vis-a-vis doing it by changing cap utilisation in cell E6 manually and then recording the profit figure in a sheet separately or keying it manually into cells F14 to F20.

LOTUS 1-2-3 Commands

We have seen in the earlier parts of this chapter as to how a table can be created and manipulated using a spreadsheet package. A spreadsheet package also offers the user the flexibility of re-arranging the matter in a worksheet. Thus, columns and rows may be shifted (without having to rewrite formulas), certain cells may be erased, the screen could be split up into two vertical or horizontal windows, the format of numbers could be changed and the width of a column could be adjusted. All these and more operations can be performed by executing specific commands from a command menu. A user can also print the contents of a worksheet, create a file on a floppy for subsequent access and data can be read into a worksheet from other types of files.

The command menu like a restaurant menu offers various choices of which one may be executed (unlike in a restaurant) at a time. The command menu for LOTUS 1-2-3 is shown in Appendix. A command menu may be viewed on the screen by typing a slash. Once the command menu is up on the screen in the second and third lines of the control panel, the worksheet enters a menu mode. You cannot then manipulate the cursor in the worksheet area and therefore entering or changing data in the worksheet is temporarily suspended. To execute a command the first letter of the command may be typed or the cursor on the command menu can be pointed to the specific command (by moving right hand and left hand arrow keys). After pointing to the specific command, the return key must be depressed to execute the command. On execution the LOTUS systems will ask for other relevant inputs which need to be supplied from the key board. It is best to understand the execution of these commands on a PC by actually working with the software. However, in Appendix a new line explanation is given for each command to explain the facility that the command offers.

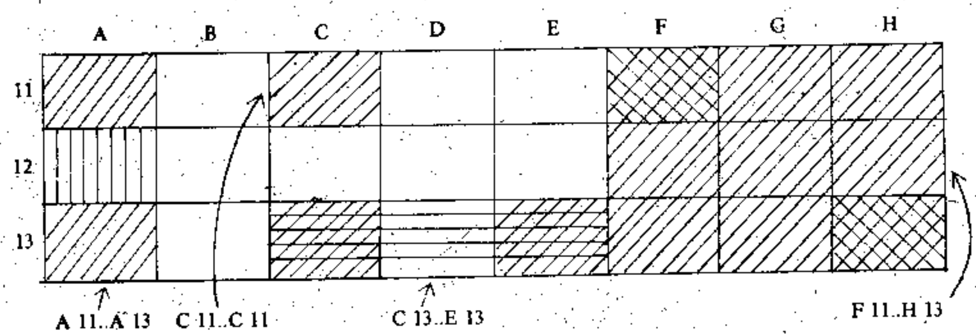
4.3 SOME IMPORTANT COMMANDS

We have chosen to discuss DATA and GRAPH commands in a little detail because these commands enable a LOTUS 1-2-3 user to do more than the spreadsheet analysis discussed earlier. The COPY command is discussed in detail because it greatly facilitates the building up of a worksheet in many situations.

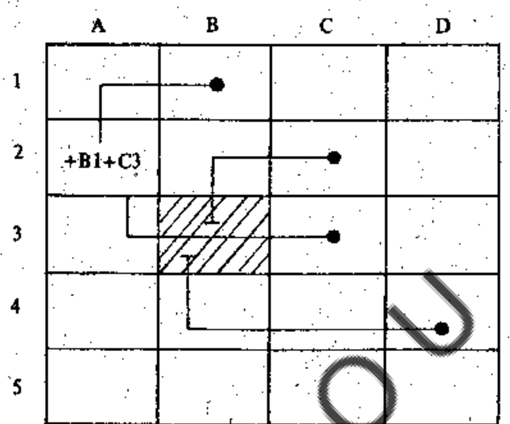
Copy

This command allows the user to copy the contents of a group of cells from one area of the worksheet into another area. For example, one may wish to create copies of labels and formulas in column A & B in the alpha worksheet or in other columns in E, F, G & H etc. where the profit and loss under different assumptions could be shown. First we will have to understand the mechanism of addressing group of cells.

In LOTUS 1-2-3 a group of cells is called a RANGE and is addressed by the left hand top corner cell and the right hand bottom corner cell. For example, the cells containing labels in column A of Figure I will be referred to as A4..A20. The figures below indicate the address for a column of cells, a row of cells, box of cells or a single cell.



When a COPY command is executed, the user must specify a FROM RANGE (which cells are to be copied) and a TO RANGE (to be copied into which cells). When the cells being copied contain labels and numbers an exact replica is created when the copy is made. However, in copying a formula, the contents of a cell may get altered. A formula like + B1 + C3 in cell A2 would be interpreted as adding in cell A2 the contents of a cell which is one row above and one column to the right to a cell which is one row below and 2 columns to the right. Thus the formula interprets the cell addresses in a relative manner. When the contents of cell A2 are copied into another cell say B3, the relative nature of the formula will be preserved by appropriately redefining the formula. Therefore, the formula in B3 will read C2 + D4.



However, if the formulas in A2 were stored as \$A\$2 + \$C\$3 (each row and column address prefaced by a \$) then such an address is called an 'absolute address'. When such formulas are copied , their form remains intact and the resulting formulas in B3 will continue to be \$A\$2 + \$C\$3.

In designing worksheets many times identical computation/operations need to be carried out on several cells. For example profit computations for one year in a column may be repeated in other columns for future years. COPY command applied to relative, absolute or mixed addresses (A\$2, \$A4) can be very useful.

Data Base Capacity

A LOTUS worksheet can also be used to create small data bases consisting of upto 1000 records and upto 30-40 (theoretically 256) fields in each record. The fields are defined in columns with the first row defining the names of each field. The data for each record is stored in successive rows. Such a data base can be defined in any area of the worksheet and can use all the features of a spreadsheet software.

One of the DATA commands will then allow a user to QUERY the data base i.e. select records which satisfy a criteria defined by the user. For example, if an employee data base was created, the user could select employees satisfying the following types of conditions.

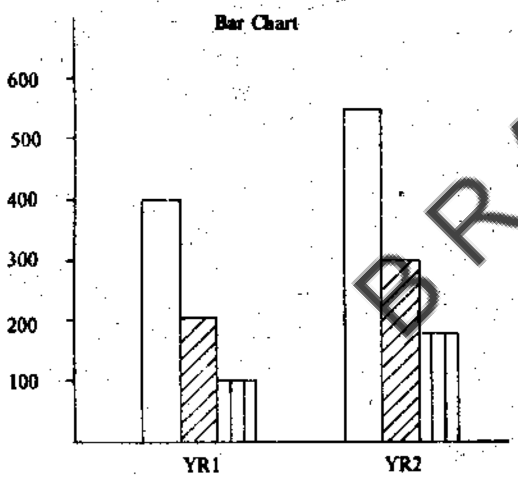
Salary	> 2000
Place	= Ahmedabad
Salary	> 2000 AND Age > 40
GRADE	= 3 OR GRADE = 5

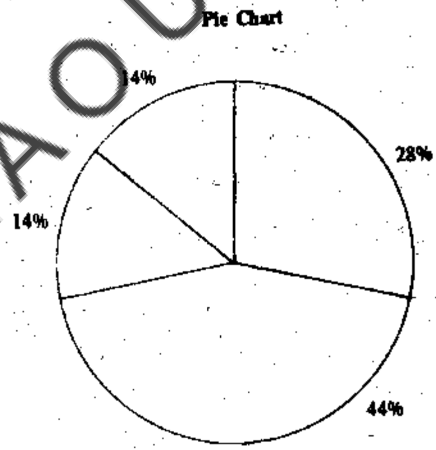
The user would have to specify such criterion in a specific format in a designated RANGE called CRITERION and will have to indicate to the system an INPUT RANGE specifying the location of the cells containing the data base. Through the use of data base functions in a cell, statistics like totals, variance, etc. could be computed for select records automatically.

A command called SORT also allows a user to rearrange the data base in ascending/descending order of value of any field.

Graphing Capability

The GRAPH command allows a user to draw various types of graphs on the basis of data stored in a worksheet. The user has the option of drawing a BAR CHART or a STACKED BAR for upto 6 variables at a time. For example the bar chart below plots sales, variable cost and profits for two years.





A pie chart can be drawn on any set of data which represents the various elements of a whole. For example, the pie chart above represents cost element of the total cost in a year. The GRAPH command also allows the user to draw a graph between two variables, each represented on one of the axis. In such graphs a point will represent a pair of values for the first and the second variables.

4.4 SUMMARY

This unit attempted to expose you to the capabilities of a spreadsheet software and in particular to LOTUS 1-2-3. The mechanics of using LOTUS were not discussed. These must be learned on a PC by actual use of the software.

The steps involved in using a spreadsheet package for solving problems typically encountered by managers were discussed. The ease with which models can be built was illustrated. Broad features of LOTUS 1-2-3 which combines spreadsheet, graphics and data base were described. Finally some advantages of using spreadsheet packages were documented.

The LOTUS is popular because of the usability of spreadsheet analysis in various managerial problems, and the fact that it can be learnt with little effort. There are some features of LOTUS that have not been mentioned so far. It has an extensive HELP facility which provides explanation of various commands and functions. HELP can be used by pressing the function key (keys to the left of the key board) F1. There are 9 other function keys. A one line description is given for all the function keys in the Appendix.

LOTUS also offers limited programming capability through MACROS. These might be classed as advanced concepts and allow a user to repeat several LOTUS commands through one key stroke. Macros also allow a user to repeat a sequence of instructions and create special menus which can be displayed on the control panel. Major use of MACROS is by professionals who design easy to use worksheet templates for particular types of analysis which can be used by simply making menu choices even by a manager unfamiliar with LOTUS.

LOTUS 1-2-3 was designed for performing financial analysis like budgeting, ratio analysis, cash flow analysis, investment analysis, and leasing; loans and amortisation computation.

However, managers soon discovered its use in a variety of other applications. Today, LOTUS 1-2-3 is extensively used by marketing managers for analysis of markets and sales, demand forecasting, costing and pricing, retail analysis, allocation of promotional expenditure. It has been reported that 20% of the LOTUS sales are to engineers. In fact all types of professionals are using it.

4.5 SELF-ASSESSMENT EXERCISES

1. State and explain what a command is List some LOTUS 1-2-3 Commands
2. Describe different functions and applications of LOTUS 1-2-3.
3. Describe various capabilities of a spreadsheet software.

4.6 FURTHER READINGS

Michael, Laric and Ronald, 1984 *LOTUS - 1-2-3 Manual,* Lotus Corporation.

LOTUS 1-2-3 for Marketing and Sales, Prentice-Hall: Engelwood-Cliffs.

Douglas, Ford Cobb and Leith, Anderson. 1984. *1-2-3 for Business,* One Corporation.

Geofferey Leblong and Douglas, *Ford Cabb. 1985 Using 1-2-3,* Que Corporation.

Jorgensen, 1986. *Master 1-2-3,* BPP Publications: New Delhi.

Gilbert and Williams, 1985. *ABC of 1-2-3,* BPP Publications: New Delhi.

Brooks, 1987, *101 Spread Sheet Exercises, Mc-Graw Hill: New York.*

APPENDIX

LOTUS 1-2-3 Command Trees with brief description

WORKSHEET

Global	:	Controls over-all settings for all the cells in a worksheet
Format	:	Set the display appearance of numbers (various formats: Fixed, Scientific, Per cent, Text, etc.)
Lebel-Prefix	:	Control label alignment within a cell Left : Right: Centre
Column-Width	:	Sets width of all columns. May be 1-72 characters wide
Recalculation	:	Controls recalculation options Natural, Column-wise, Row-wise, Automatic, Manual, Iteration
Protection	:	Contents of a protected cell cannot be modified. Controls access to protected cells through Enable, Disable
Insert	:	Creates a blank column/row
Column	:	Shifts the contents of all columns to the right of a specified column by one column. The formulas in shifted columns are readjusted automatically.
Row	:	Shifts the contents of all rows below a specified row down by one row. The formulas in shifted rows are adjusted automatically.
Column	:	Erases the contents of a specified column and shifts leftwards. The contents of all columns to the right by one column
Row	:	Erases the content of a specified row moves upwards the contents of all rows below by one row
Column-Width	:	Controls width of a specified column
Set	:	Set new width
Reset	:	Returns to global width
Erase	:	Erase entire worksheet
Window	:	Split display screen in two parts - Horizontal and Vertical. In each part the entire work sheet can be scrolled in a synchronised or unsynchronised mode
Status	:	Display global settings

RANGE

Format Label-Prefix Erase Protext Unprotect	:	Similar to options under worksheet but operate on a specific part of the worksheet.
Name	:	A range can be given a name (CREATE) through which it can be referenced subsequently. Names can be deleted (DELETE) and labels can be used to name adjacent cells (LABEL).
COPY	:	Copy range contents to new location. Previous contents of the new location are erased
MOVE	:	Move range contents to new location. Previous contents of the new locations are erased
FILE		
Retrieve	:	retrieve worksheet from disk file
Save	:	store worksheet on disk file
PRINT	:	For printing worksheet
GRAPH	:	Graphics commands
Type	:	select graph type
Line	:	graph upto a maximum of 6 data sets (A-F ranges) as lines and/or symbols
Bar	:	graph upto a maximum of 6 data sets (A-F ranges) as clustered bars
XY	:	plots pairs values of 2 data sets as points on a XY graph on which both axes are scaled.
Stacked-Bar	:	graph upto a maximum of 6 data sets (A-F ranges) as stacked bars
Pie	:	Graph A single range as pie chart
X A B C D E F	:	X range defines the Tables that would be used on the X axis of Line, bar, stacked bar and on the periphery of a pie chart. A-F ranges define the data sets that would be used to plot the graphs.
Reset	:	return settings to default values
Graph	:	clear all graph settings
X A B C D E F	:	remove range from current graph
Quit	:	return to main-Graph menu
View	:	display current graph on screen
Save	:	save current graph as a picture in disk file
DATA	:	Data base command
Fill	:	fill range with a series of numbers
Table	:	tabulate formula values

1	:	Automatically vary contents of one input cell and tabulate one or more formulas. Contents of input cell to be taken from a column of numbers stored in a specified area of the worksheet, prior to executing the command.
2	:	Automatically vary contents of two input cells and tabulate one formula. Contents to be picked up from a column and row of numbers already created in some area of the worksheet.
Sort	:	Rearrange records (rows) in a data base in a specified range on the value of one field (column).
Data-Range	:	set range to short
Primary-Key	:	set first column to sort on
Secondary-Key	:	set second column to sort on
Reset	:	cancel all sort settings
G	:	perform the sort
Quit	:	return to/Data menu
Query	:	perform operations on a database
Input	:	specify the database range
Criterion	:	specify the range where selection criterion will be specified
Output	:	specify the range where selected records will be output
Find	:	highlight records matching criteria
Extract	:	copy all records matching criteria to output range
Unique	:	extract, but delete duplicates
Delete	:	delete all records matching criteria
Reset	:	cancel input, criterion and output ranges
Quit	:	return to Ready Mode
Distribution	:	calculate frequency distribution
QUIT	:	leave 1-2-3

F1-F10 Operations

Function Key	Name	Operation
F1	Help	Displays the Help Screen
F2	Edit	Switches into or out of the Edit Mode
F3	Name	In the Point Mode, displays menu of range names
F4	Abs	In the Point Mode, designates cell addresses as absolute
F5	GoTo	Used to move the cell pointer to a designated cell

F6	Window	Used to move the cell pointer to the other side of the split screen
F7	Query	Repeats the Data Query operation last used
F8	Table	Repeats the Data Table Operation last used
F9	Calc	Recalculates the worksheet
F10	Graph	Recreates the last graph drawn

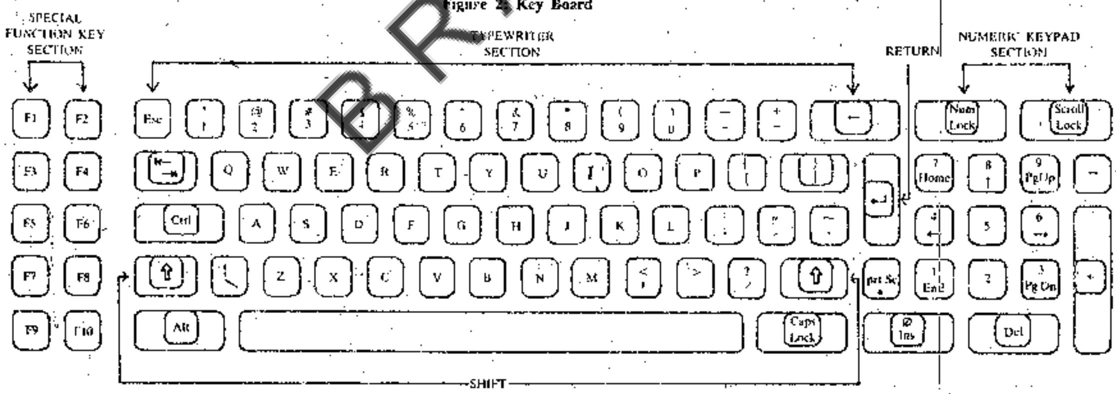
1-2-3 Function

Function	Configuration	Description
Mathematical		
@ABS	@ABS (a)	absolute value
@ACOS	@ACOS(a)	arc cosine
@ASIN	@ASIN (a)	arc sine
@ATAN	@ATAN (a)	2-quadrant arctangent
@ATAN 2	@ATAN (a,b)	4-quadrant arctangent
@COS	@COS (a)	cosine
@EXP	@EXP (a)	exponential
@INT	@INT (a)	integer
@LN	@LN (a)	natural log
@LOG	@LOG (a)	absolute value
@MOD	@MOD (a,b)	modulus of a/b
@PI	@PI	(pi)
@RAND	@RAND	random number (0 to 1)
@ROUND	@ROUND (a,b)	round a to b decimal places
@SIN	@SIN (a)	sine
@SORT	@SORT (a)	square root
@TAN	@TAN (A)	tangent
Statistical		
@AVG	@AVG (list)	average of values in a list
@COUNT	@COUNT (list)	counts items in a list
@MAX	@MAX (list)	maximum value in a list
@MIN	@MIN (list)	minimum value in a list
@STD	@STD (list)	standard deviation of a population
@SUM	@SUM (list)	sum of values in a list
@VAR	@VAR (list)	variance of a population

Logical		
@IF	@IF (expression true clause, false clause)	if expression = true, true clause if expression =/= true, false clause
@ISERR	@ISERR (a)	1, if a = ERR 0, if a = ERR
@ISNA	@ISNA (a)	1, if a NA 0, if a =/= NA
@FALSE	@FALSE	0
@TRUE	@TRUE	1
Financial		
@FV	@FV (payment, interest, periods)	future value
@IRR	@IRR (guess, range)	internal rate of return
@NPV	@NPV (interest rate, range)	present value of a series of unequal payments
@PV	@PV (periodic payment. periodic interest rate, no. of periods)	present value
@PMT	@PMT (principal, periodic interest rate, no. of periods)	periodic payment amount
Database	(Here 'a' is the number of columns to the right of left-most)	
@DAVG	@DAVG (input range, a, criterion range)	averages of values meeting criteria
@DCOUNT	@DCOUNT range, a, criterion range)	counts records that meet criteria
@DMIN	@DMIN (input range, a, criterion range)	minimum value that meets criteria
@DSTD	@DSTD (input range, a, criterion range)	standard deviation of values meeting criteria

@DSUM	@DSUM (input range, a, criterion range)	sum of values meeting criteria
@DVAR	@DVAR (input range, a criterion range)	variance of values meeting criteria
Date		
@DATE	@DATE (year, month, day)	returns a date serial no.
@DAY	@DAY (a)	day number of a date serial number
@MONTH	@MONTH (a)	month number of a date serial number
@TODAY	@TODAY	returns the last date input when 1-2-3 was loaded
@YEAR	@YEAR (a)	year number of a date serial number
Special		
@CHOOSE	@CHOOSE (test, response, response, response,...)	selects an argument
@ERR	@ERR	ERR
@HLOOKUP	@HLOOKUP (test, table, a)	horizontal look-up
@NA	@NA	NA
@VLOOKUP	@VLOOKUP (test, table, a)	vertical look up

Figure 2: Key Board



BLOCK 2 MANAGERIAL APPLICATIONS OF COMPUTERS

This block introduces you to the role of computers in efficient and effective performance of various management functions.

Unit 5 on Computer and Management Functions introduces the role of computers in management decision-making and explores their uses in management disciplines like Finance, Production, Inventories, Maintenance, Marketing, Personnel, etc.

Unit 6 then specifically discusses computer based financial systems. It discusses about the computer applications for financial systems, general and computerised ledger system, and computer aided financial planning. It also introduces you to the spreadsheet analysis.

Unit 7 deals with Computer based Inventory Management. It gives you an idea about the characteristics and purpose of computerised inventory system and their application. It also deals with inventory system design and software packages for inventory management.

Unit 8 on Computer in Human Resource Management specifically deals with the needs, design, uses and software packages of computer based information systems for an effective and efficient Human Resource Management System.

UNIT 5 COMPUTER AND MANAGEMENT FUNCTIONS

Objectives

After reading this unit, you should be able to:

- understand the role of computer in Management Decision-making
- appreciate the need for proper handling of the information and an efficient Management Information System at corporate level.

Structure

5.1 INTRODUCTION

The common experience in the Indian scene has been a plethora of transaction-processing systems which have been abiding for long. Left to themselves, computer professionals have usually been good at planning, designing developing and implementing the transaction-processing systems, due to their inherent advantages of recurring qualitative data and decisive logic base. On the other hand, MIS at a corporate level always needs involvement of the user-manager, in an intensive way and this has shown a tardy growth.

We enumerate below a few selective applications in some of the decision-making areas.

5.2 FINANCIAL DECISION-MAKING

At corporate level MIS would still need the data-base comprising the transaction-processing inputs, such as cash transactions, receipts, issues, returns, rejections, interests, depreciation, etc. The other inputs are in the nature of the various accounting rules and procedures, apart from the scales of payment.

Based on the above, the financial management decisions are made with monthly and annual final accounts, profit and loss accounts, balance-sheet etc. In addition, cash accounts are needed by management, apart from the cost accounting, pay-roll accounting, suppliers ledger and preparation of capital and operations budget.

To build up a corporate-level MIS, one can look at such selective data as bills payable and bills receivable, and segregate them commodity-wise, establishment-wise, region-wise, instrument-wise and age-wise. This immediately gives a management decision-making capability to examine and take remedial action on the differences and deviations from the set pattern. The analysed data for financial decision-making can have various management ratios and management trends leading on to managerial forecasts. Linked data can be of the form of inter-related items between finance and inventory, between finance and marketing, between finance and production costing, and between finance and personnel. Integrated data can take the form of obtaining a picture for total credit, total sales turn over, total profit and so on.

The corporate areas to be helped would be cash planning, credit planning, profit planning, and facility planning. For instance, cash inflow and outflow are of great interest for most organisations and much monumental bankruptcy had taken place in such organisations as Rolls Royce in the early 70s due to the faulty cash-flow adjustments. Similarly, the credit planning takes care of voluntary and involuntary credit, in establishment where voluntary credit is often deliberately introduced to push new products, to lift accumulated stock or to provide relief to bulk customers. In all such areas, decision-making is facilitated by an appropriate MIS, as has been experienced in many Indian production organisations.

In a service organisation, investment planning and budget planning are additional features. Interestingly, budget planning can be a common feature in both production and service organisations. In particular, the banking organisations have shown a tremendous scope for corporate MIS in terms of trends, analysis and forecasts arising out of various reconciliation processes such as inter-branch, inter-bank, or even for such instruments as bank drafts and travellers cheques.

In the domain of public administration, commercial tax data-base has been built-up in some of the Indian States for various categories of taxes on commodity-wise, establishment-wise, region-wise and year-wise bases. Such a data-base has given rise to a capability to concentrate upon tax evasion cases selectively, to strengthen the inspection machinery for tax non-realisation cases, to keep track of recovery of diverse instalments in court-injunction cases and generally to streamline pursuit of the outstandings. In the States of Karnataka and Maharashtra, most of these steps undertaken today are based on computerised MIS.

Activity A

Record your perceptions about the role of Managment Information System in Financial Decision-making in your organisation.

....

....

....

....

....

....

....

....

5.3 PERSONNEL DECISION-MAKING

Personnel pay-rolls have been an old hat in transaction-processing systems. In a number of organisations, pay-rolls have been extended to cover the entire gamut of employee service, namely, income-tax assessment and recovery, provident fund accumulation and withdrawal, compulsory deposits (when they existed) and statutory reports and returns. A comprehensive data-base for personnel decision-making would require not merely such financial data but also other categories of pre-recruitment and post-recruitment personnel data.

Examples of comprehensive personnel data would include prior qualifications, experience, categories and grades of posting, places of transfer, experience gained, new qualifications acquired, training undertaken, disciplinary actions, merit rewards and performance appraisals, etc. Such a data-base can be used for retrieval of selective data to assess grade-wise, year-wise and region-wise personnel inflow and out flow, for deciding upon recruitment planning as well as promotion planning. A similar approach can be undertaken to match availability of persons experience-wise, location-wise and duration-wise with the demand of such persons in new locations with appropriate grades and experience for purposes of personnel plan. Career development planning can also be done while looking at the grade-wise, experience-wise and qualification-wise profile and decide on fresh areas of training and education.

Finally, retirement planning can be done by looking at the selective aggregation of data on personnel wastage, so that recruitment can be undertaken and the bulk of retirement benefits can be released rightaway pending the settlement of accounts.

Personnel decision-making based on corporate MIS has become quite common in a number of large industrial organisations in the country. In addition, states like Andhra Pradesh and Madhya Pradesh have often done personnel planning for doctors and teachers respectively using comprehensive data-base approach. Bhilai Steel Plant has planned total human resource management with creation of complete data-base including history and skills of personnel; and manpower planning and administration, including establishment functions; training needs as identified in the appraisal reports and so on.

Activity B

Make a list of the pre-recruitment and post-recruitment personnel data required by your organisation for personnel decision-making.

5.4 MARKETING DECISION-MAKING

Sales invoicing and sales accounting have been the early transaction-processing systems in most organisations. The inputs have been the production, inspection and despatch of documents on goods produced and despatched apart from the price schedules and some occasional returns. Such data can be built into a comprehensive data-base which can help eventually the market planning and the publicity (advertisement) planning on one hand and linkages with order-processing on the other.

Both market planning and publicity planning depend on the creation of a number of profiles with full particulars such as customer profile, product profile and user industry profile. The seasonality of orders, the trends in market fluctuations, the philosophy of industrial indents and the ebb and flow of relevant markets can be assessed and taken care of in such a planning process. On the other hand, linkages with order-processing gives a vital control on the quantity to be produced or suspended to meet live and dead orders respectively. Such a dynamic interfacing can help to eliminate a lot of infructuous order execution and finished goods inventory creation, as has been experienced in the Tata Steel.

Activity C

Prepare a brief report on the market planning and publicity planning of your organisation, and the role computers and MIS can play in that.

5.5 PRODUCTION DECISION-MAKING

The basic inuts are the salient production data culled periodically from all the production shops as and when the main events occur. The corporate MIS would involve: performance review on a periodic basis (daily, monthly and annually), monitoring of in-process inventory, balancing of daily finished and semi-finished stocks, yields and other performance statistics, comparison of the current production with the past performance, receipt and consumption of services and energy resources and daily analysis of performance.

Corporate production decision-making involves production planning and control (PPC). The first element of PPC is the order processing involving maintenance of current status of orders and despatch programmes, loading schedules on the basis of despatch outstanding, stock availability and logistic facilities, and preparation of despatch advices. The second element of PPC is the corporate production MIS indicated earlier. The third element of PPC is the material tracking including maintenance of detailed production schedules through various stages of production operations (from initial loading up to the final finishing); preparation of inspection and tests certificates; scheduling of logistics facilities; preparation of production orders, based on production schedules; and in-process material and stacking report for all storage yards. The fourth element of PPC is the planning, including annual plans based on market demands; plant maintenance schedule and corporate plan requirements; quarterly and monthly plans based on annual plans and production targets; daily plans for fulfilment of monthly plans; comparison of plan versus performance at different time intervals; and preparation of subsidiary plans for requirement of raw-material, energy resources and finished inventory.

TISCO, TELCO, SAIL's Bhilai Plant and others have today incorporated PPC as part of their corporate MIS.

5.6 MATERIALS DECISION-MAKING

Materials accounting based on receipts, issues, returns, and rejections have been the part of transaction-processing system in all organisations so far. For the purpose of materials decision-making, comprehensive data-bases can have three components of purchasing, inventory, and materials review.

The purchasing data-base can help creating a corporate MIS for monitoring of purchase indents up to acceptance of tenders, monitoring of purchase orders against schedules of delivery, analysis of lead-time delays, vendor-rating (in terms of quality performance of delivery schedule), and monitoring payments against deliveries.

The inventory data-base can help in creating a corporate MIS for preparation of standard specifications for regular consumption items, prompt inspection and acceptance of delivered goods, preparation of receipt documents, monitoring of stock balance, purchase dues, and indents dues; monitoring of procurement of regular consumption items, decentralised control of consumption, analysis of consumption and movement (particularly for slow moving and non-moving) of items for control, forecasting and budgetary control of consumption and procurement and so on.

Materials review data-base can help in creating a corporate MIS for the on-going review of the various control parameters such as re-order level, re-order quantity, the phasing of deliveries for 'A' class items, the bulk purchasing of 'B' and 'C' class items, and so on. Basically, materials review should look critically at all the expected norms of purchasing and inventory so that a dynamic adjustment is possible before the on-set of any crisis.

5.7 MAINTENANCE DECISION-MAKING

Maintenance management distinguishes between preventive maintenance and break-down maintenance. While preventive maintenance can be planned, the break-down maintenance is invariably unplanned.

As regards plant preventive maintenance, the experience with oil refineries, industrial blast furnaces, dock facilities and thermal projects have all shown that a thorough planning and a scheduling of preventive maintenance can be immensely time-saving and cost-reducing.

Transaction-processing systems have looked at the records of components, spare-parts, fixtures and tools as inputs, and consumption statements and maintenance accounts as outputs. These inputs can still be valid for building up a corporate MIS with such additional data as equipment conditions, history of failures, direct cost of maintenance, inventory values and materials movement, man-hours spent, over-time paid, other resource usage, maintenance of workers' performance, reliability and maintainability of equipment, and maintenance contribution to unit manufacturing cost.

From such a comprehensive data-base, one can generate MIS reports for equipment control, such as equiptment register (to provide equipment specifications and project details), equipment history (to provide maintenance particulars), major failure report, forecast on maintenance (providing probable failure time and period), plant non-availability, plant reliability and maintainability, and maintenance schedules (providing maintenance activities to be done during a specific period). One can also generate MIS reports on work control, such as, craft performance report (providing planned and actual performance of each craft), craftman's performance report, maintenance planning efficientcy report (providing an efficiency ratio on planned activities), overtime report, resource levelling report (providing analysis of resource required job-wise), delay cause report, etc. One can finally generate materials control reports and cost control reports providing management details of costs and material transactions involved in maintenance management.

Activity D

Prepare a MIS report for equipment control in your organisation.

5.8 SUMMARY

This unit provides a brief account of the practice of computer -aided decision-making in such selected functional areas as finance, personnel, marketing, production, materials and maintenance. Some of the areas left out are project, medical and health care, township, training, fire service, energy and industrial engineering. These all have similar needs to build up from basic inputs a good quality data-base each and then to evaluate selective data, analysed data, interlinked data and aggregated data to derive suitable MIS support to decision-making at the corporate level. The practice is becoming more and more endemic in the Indian environment and the proliferation can only be beneficial to the management in the long run.

5.9 SELF-ASSESSMENT EXERCISES

1 Evaluate the role of computers in Management Decision-making.

2 Examine the advantages of Computerised Management Information System vis-a-vis a manual MIS.

3 Evaluate the existing Management Information System (MIS) in your organisation. Suggest measures to improve it.

UNIT 6 COMPUTER BASED FINANCIAL SYSTEMS

Objectives

After studying this unit you should be able to:

- define what are various aspects of financial systems
- identify problems associated with computerisation of financial systems
- design some financial applications for computer implementation
- design spreadsheet for financial applications.

Structure

6.1 INTRODUCTION

Historically, operations such as accounts payable, cost accounting and financial statement preparation were among the first to be automated, via punched cards systems in the 1930s and 1940s and on computers in the 1950s and 1960s as in pay-roll procedures. These functions are largely routine, relatively easy to automate. The benefits offered by such applications traditionally have been in the form of clerical cost reduction, and although they have not normally led to dramatic increase in profitability, they have produced sufficiently large savings to more than justify their computerisation.

In spite of the traditional nature of accounting and finance applications, several sophisticated, high benefit systems are being implemented in this administrative and business area. Cash management, financial modelling and advanced purchasing systems are three examples of the new interest in accounting and finance applications.

In large, diversified, or widely dispersed organisations, the management of cash resources is a difficult job, yet one that has a large impact on profitability. Failure to invest temporarily available cash, premature payment of obligations, or short-sighted investment programmes that force an organisation to borrow at high rates, all result in less than optimal use of financial resources. In an attempt to avoid these problems, many large organisations are using computers to help collect, analyse and report data about cash requirements and reserves. The benefits of such systems include less frequent and smaller short-term loans, lower rates of interest, and higher returns on short-term investments.

Financial analysis of proposals is the second area where computers are playing an increasingly important role. Relatively straightforward simulation models enable the financial manager or analyst to generate pro forma statements that show the financial impact of different proposals, such as adding a new product to a current line or opening a new warehouse. The advantage of such models is that many possible outcomes can be evaluated in the time that one or two could be calculated by hand.

Advanced purchasing systems are the third area of current interest. Computer-based systems monitor the data about price and quantity discounts, product quality and reliability, and speed of delivery. On the basis of these data, computer programme can calculate vendor rankings and economic purchase quantities for individual items and store product information that helps buyers evaluate vendor performance and negotiate favourable contracts with suppliers.

Table 1 lists many of the traditional accounting and finance applications as well as those of high current interest.

Table 1

Finance Systems - Computer Applications

Applications	Least Complex		.........Most Complex
General Accounting	●Cost record keeping	●Cost accounting comparison to standards or projected amounts ●Budgetary accounting ●Daily exception reporting	●Cost estimating
Accounts Payable (A/P)	●Preparation of A/P registers ●Check processing ●Check reconciliation ●A/P distribution		
Purchasing	●Vendor analysis-Volume purchase ●Purchase order preparation and follow-up	●Vendor analysis-Quality ●Knowing Economic purchase quantities ●Requirement	●Make or buy Analysis ●Analysis of financial proposals

● Financial statement preparation	Planning e.g. cash management system ● Maintenance of shareholders records

6.2 FINANCIAL SYSTEMS

These systems are primarily concerned with recording business transactions in respect of wages and salaries, purchases, sales and other aspects of income and expenditure, both capital and revenue. Records of such transactions provide basis for the preparation of periodic or annual profit and loss accounts and balance sheets.

As can be seen, the financial systems of a business are, effectively, accounting systems which are often structured as separate systems to the non-financial systems. When systems are computerised, the separately structured systems are often integrated for economy of data processing and administrative efficiency.

Functions of Financial Systems

One of the most important functions of financial systems is to ensure that all business transactions are recorded in proper books of account, on the basis of recognised accounting practice. Such accounting transactions are largely for purpose of custodianship, as a public limited company is responsible to the shareholders - the owners of the business - and accordingly it is essential that the business records potray a true and accurate record of profits and losses, assets and liabilities.

Other functions of financial systems are summarised below:

- Planning and controlling all expenditure, both capital and revenue
- Controlling the receipt and payment of cheques, etc. relating to business transactions and relevant banking transactions.
- Safeguarding the assets of the business in respect of plant and machinery, stocks, debtors and cash
- Maintaining statutory records as per Government's regulations
- Preparation of periodic reports for statistics, performance and results for internal control and audit.

Activity A

How are the above functions being performed in your organisation? Record your perceptions about them, function-wise, choosing one of the alternatives given below regarding their performance.

(i) Satisfactorily (ii) To some extent satisfactorily (iii) Not satisfactorily

....

....

....

Activity B

What would you suggest to make the system more efficient?

6.3 FINANCIAL MANAGEMENT SOFTWARE

The micro computer can provide a new and very efficient tool to improve the service. For example, it can transform efficiency by replacing a large amount of manual record-keeping and, perhaps more significantly, it can extend the quality of the service being offered. After introducing a computerised sales ledger, the regular balancing of the ledger (itself a considerably easier task using a controlled computer system) will be accompanied by the printing of an 'aged' list of balances. From this report the accountant can initiate and monitor a follow-up routine to recover any overdue amounts. In this and similar situations the firm will be offering a much more powerful financial control system that combines the interpretative skills of the accountant with the routine efficiency and analytical capabilities of computerised ledger systems.

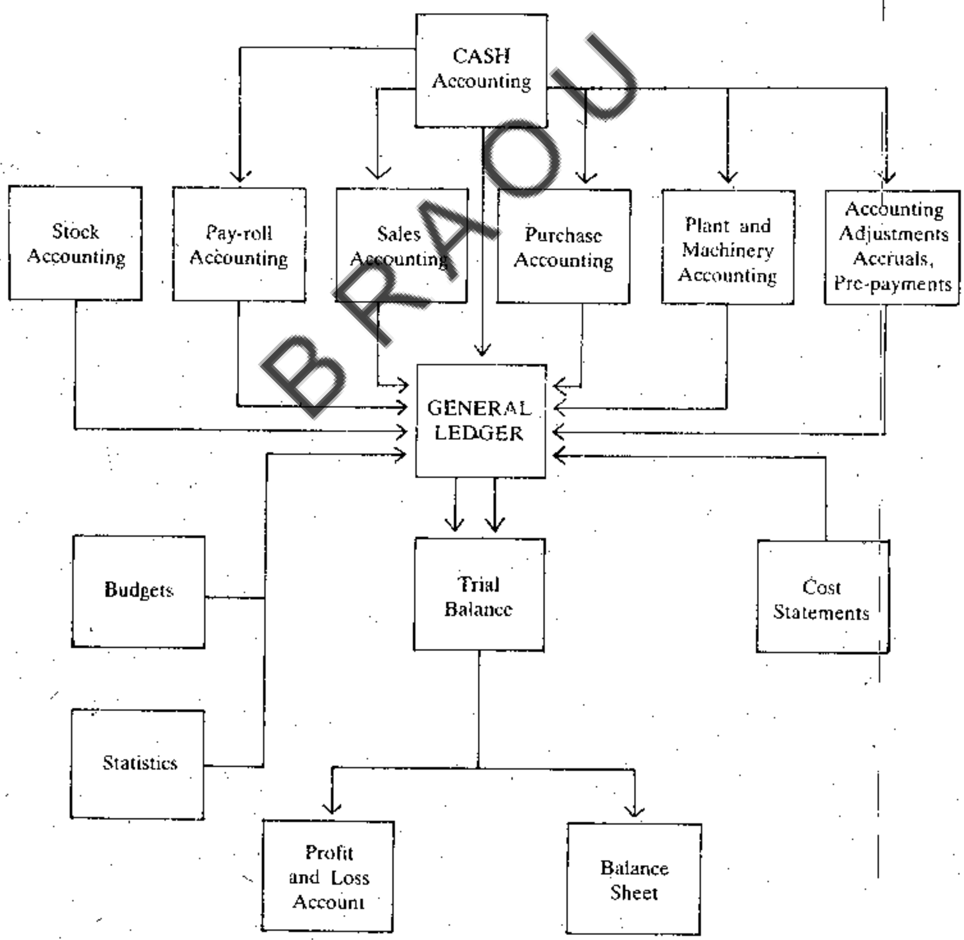
Much of the software available for the practising accountant will, of course, also be of use to his clients as well as to other professional firms. Although some features may be required specifically for the accountant, his management control needs will be very similar to those of other businesses or practices so most programmes offered are described in their publicity material in ways that illustrate their widest possible use, including uses for which they may be only marginally suited. Some of the most widely used financial management software can be listed as:

- General Ledger
- Sales ledger
- Purchase ledger
- Invoicing

(Stand alone or integrated, incorporating budgetary control and management accounting)

- Stock control
- CCA adjusted accounts
- Pay-roll
- Modelling techniques

Figure I: The General Ledger



General Ledger

The focal point of financial accounting systems in the general ledger is sometimes referred to as the nominal ledger. The nominal ledger consists of accounts in which transactions are recorded from the point of view of business and these are classified as 'impersonal accounts'. Impersonal accounts are sub-divided into 'real accounts' and 'nominal accounts'. Real accounts are concerned with the tangible assets such as plant, machinery and buildings, whereas nominal accounts are concerned with expenses, income, profits and losses. Financial accounting sub-systems are directly related to the general ledger by way of double entry convention for recording of business transactions. In respect of the purchase accounting system, the general ledger contains the purchase ledger control account and account from different classes of purchases. The Sales accounting system has accounts in the general ledger by way of Sales Ledger control account. Similarly, the wages and salaries accounting system has the wages and salaries control account in the general ledger.

Cash transactions, in respect of the sub-systems indicated above are effected in the general ledger: for cash receipts from customer, they are recorded in the Bank account and the Sales ledger control account (Figure I).

Computerised General Ledger Systems

The General ledger system on computer can be implemented as an integrated system to form a total system by combining a number of related sub-systems for the purpose of improving administrative efficiency. Figure II describes the system run chart for one of the general ledger systems.

We shall briefly mention the main features of a computerised general ledger system which are easily available in the market:

- Automatic double entry accounting
- Accruals and prepayments are automatically reversed in the next accounting period
- The general ledger is maintained as a data-base and the Analysis Printing Programme can report in any sort, sequence and in many reporting formats which can be stored on computer
- When linked with the purchase ledger, standard monthly postings can be made automatically
- A history file of all the transaction records can be maintained so that analysis and schedules can be produced at the year end
- Profit or Loss statements, Balance Sheet and operating statement for the management can be produced
- Budgetary control can be exercised producing variances from Budget
- Budget flexing enables budgets to be altered by percentage changes, and produce financial projections as a result of the change
- Nominal accounts can be reanalysed into cost or profit centres, producing reports in up up to seven different types of sort keys
- A year end report produces summary of each nominal account for each accounting period, compared to budget and or previous years.

Figure II illustrates computer runs for a general ledger system which is based on the outline Figure I. The input to run 1 is derived from data produced by the separate computer applications in respect of transactions relating to stocks, pay-roll, sales, purchases, plant and machinery including depreciation, accruals and pre-payments, accounting adjustments and cash. The transaction data in respect of each application is assumed to be stored on magnetic tape. Run 1 is concerned with consolidating all nominal ledger data and this is achieved by merging all the relevant transactions on the general ledger codes. The output from Run 1 is a consolidated file of transactions, which is input to Run 2 for updating the general ledger file. This file is stored on magnetic disc to facilitate direct access to relevent general ledger account codes. Run 2 also produces a list of detailed postings to each account.

At the end of the month, the general ledger file is input to Run 3 to produce a Profit and Loss Account and Balance Sheet. The general ledger file is also input to Run 4 together with a budget file which is updated with actual expenditure providing the cumulative expenditure to date for comparison with budgeted expenditure. A variance report is then printed for analysis by respective cost centres.

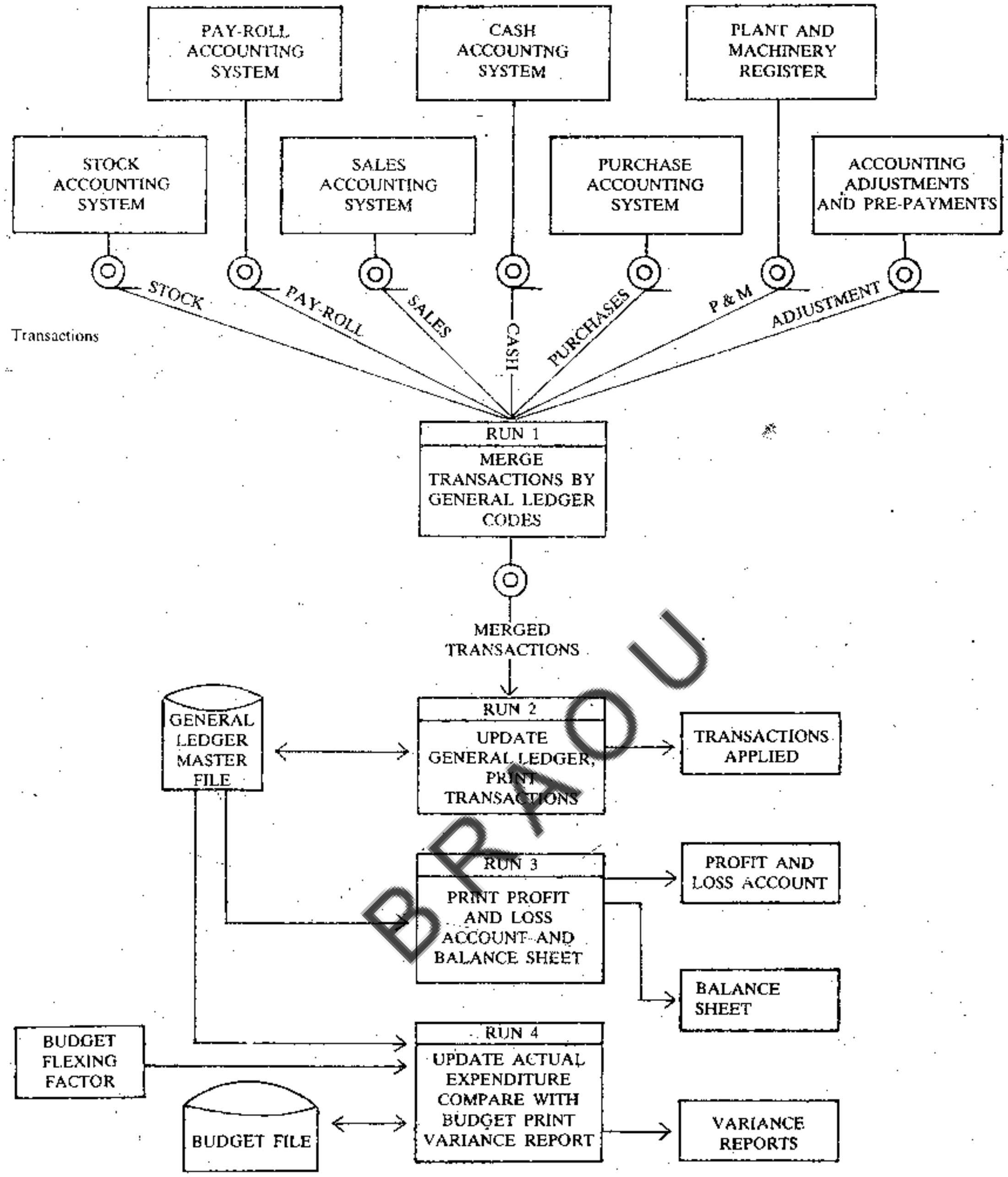
Incomplete records: This complex area has been a prime target for the development of computer programmes and these are available on most of the popular micro-computers. A typical package will hold between 250 to 900 nominal account headings and between 2,500 to 4,000 transactions. The routines for posting the transactions will be so structured that each prime entry source will be posted automatically to the respective nominal account.

Nominal account codings can be defined to suit each client. The format of the accounts can be used to provide an outline sequence within which to create the accounts code list. It is advisable to allocate the codings in blocks that will allow additional accounts to be opened without destroying this sequence. It is impossible to define this structure without knowing the coding facility of a particular package or the requirements of the client. One example using a four-digit structure could be as follows:

General Ledger	Accounts Codes
1000	Fixed assets
2000	Current assets
3000	Current liabilities
4000	Capital
5000	Reserves

Within each group up to 999 accounts might be available to allow for the detailed analysis. With the Reserves group (in the illustration the 5000 series), more numbers are available to be

FIGURE II: Computerised Integrated General Ledger System



allocated to revenue and expenses items to be identified in the profit and loss account. In this latter case, the series numbered 6000 onwards could be used to create a profit or cost centre analysis within the profit and loss account.

A common feature of many of these packages will be the automatic calculation of depreciation using preset depreciation parameters. Reports generated will usually include:

- Trial balance
- Bank and cash reconciliation
- Adjustments analysis
- Fixed asset schedule
- Nominal ledger accounts in detail
- Profit and loss account
- Balance sheet
- Source and application of funds.

Most programmes now offer perhaps the most useful facility of all, namely the preparation of a full set of final accounts, including notes to the accounts and the director's report. However, it must be said that some systems will not print to a particularly high standard and you may be unable to use the printout from the machine for presentation.

Integrated purchase/sales/general ledger: There are several packages available which offer the above reports either as individual modules or as an integrated system. One of these has been already discussed earlier. In addition, systems are now offered which link the sales and purchase ledgers to a stock control programme, and the sales ledger to an invoicing programme. Most systems offer either 'open item' or 'balance brought forward' on each account. Naturally, the open item system offers a more comprehensive service - balance brought forward being ideal for smaller businesses whose accounts are cleared monthly.

The system will create the double entry within the general ledger for all postings through the sales and purchase ledgers. Journal entries to the general ledger will only be accepted if the debits and credits equate to zero, as will input through the purchase and sales ledger, thereby ensuring that the trial balance must always balance. Additionally, pay-roll and job costing may be linked to the general ledger to produce final accounts and other management information.

All inputs to the system will be via the Visual Display Unit (VDU) with the screen not only displaying data keyed to the system but also instructing the operator on the next step to be taken or reporting any errors that may occur. Input of data is normally through a batch posting system, using a check-digit total for each batch to ensure accuracy.

Other financial Systems

A typical system will offer around 800 accounts and 4,000 to 6,000 postings per month. It will hold details of customer's accounts and register all sales transactions. Initially, the user will be presented with a 'menu' of the various routines available from which the required routine is chosen, as shown in Figure III.

Figure III: A typical menu presented on a VDU screen

XYZ Limtied	Sales Ledger Menu

SALES LEDGER

Press the required number

1	Master file update
2	Master file print
3	Batch posting
4	Account enquiry
5	Cash allocation
6	Month-end routine

The customer master file will hold the name, address and telephone number, credit limit, account balance 3 to 4 months old, turnover statistics, sales territory codes etc. The transaction file will hold details entered into the system together with all invoices, credits, adjustments and cash items posted during the current financial periods.

6.4 COMPUTER AIDED FINANCIAL PLANNING

Plans have a central role to play in a company's decision-making process. To do this effectively they often need to present a number of alternative evaluations based on different assumptions and different ideas for new projects, product introduction and the like. Here, then, is the fundamental defect of manual planning and budgeting methods, a defect which can often be overcome by the use of computerised methods. The plans, to be of any use, must be easy to produce answers in a number of 'what if?' formats:

'What happens if we give an extra discount of 5 per cent for orders over Rs. 1,000 resulting in increased sales volume of 2 per cent?'

'What happens if it only gives 1 per cent extra?'

'What if we keep the old version going for a further year and hold the price, at the same time deferring its replacement and putting in Rs. 1,00,000 more for development?'

'What if we open a new warehouse in Bhopal in two years' time which takes 10 per cent of the volume from Delhi and 25 per cent from Bombay while sales go up by 15 per cent in each region and how full will the three warehouses be?'

'What does the picture look like if we run all three ideas (above) together next year?'

'What will change if we defer the new warehouse for a further year?'

'What if we put on a second shift with a 30 per cent premium on wages but at the same time stop all production overtime?'

Given a calculator, large sheets of paper, the basic information and peace and quiet, any one of the above questions could be evaluated by most accountants or managers. The work is quite straightforward. It would not be so easy to find the time to work through all of them. But this is a scenario familiar enough in most companies, and often it results in decisions having to be taken without an adequate evaluation being completed. This is not so because there is, as is so often alleged, 'a lack of information on which to base the decision' but rather because there is insufficient time to work through all the options with pen and paper.

Other difficulties with manual systems, which admittedly can be partially alleviated by use of the more sophisticated types of calculator, include working out discounted cash flows and other types of evaluation of returns on investment. A computer, however, will drastically reduce the time taken.

Figure IV: COMPUTERISED SPREADSHEET—AN EXAMPLE

1 A	B	C	D	E	F	G	H	I	J	K	L	M	N
			CASH PROJECTION										
2	Jan	Feb	Mar	Apr	May	June	July	Aug	Sep	Oct	Nov	Dec	Total
3 CASH RECEIPTS													
4 Small Builders	0	0	0	0	0	0	0	0	0	0	0	0	0
5 Distributors	0	0	0	0	0	0	0	0	0	0	0	0	0
6 Private Customers	0	0	0	0	0	0	0	0	0	0	0	0	0
7 Others	0	0	0	0	0	0	0	0	0	0	0	0	0
8	–	–	–	–	–	–	–	–	–	–	–	–	–
9 TOTAL RECEIPTS	0	0	0	0	0	0	0	0	0	0	0	0	0
10	–	–	–	–	–	–	–	–	–	–	–	–	–
11 CASH PAYMENT													
12 Timber Suppliers	0	0	0	0	0	0	0	0	0	0	0	0	0
13 Misc. mat Suppliers	0	0	0	0	0	0	0	0	0	0	0	0	0
14 Wages	0	0	0	0	0	0	0	0	0	0	0	0	0
15 Rent and Rates	0	0	0	0	0	0	0	0	0	0	0	0	0
16 Other Overheads	0	0	0	0	0	0	0	0	0	0	0	0	0
17 Interest	0	0	0	0	0	0	0	0	0	0	0	0	0
18 Drawings	0	0	0	0	0	0	0	0	0	0	0	0	0
19 Tax Payments	0	0	0	0	0	0	0	0	0	0	0	0	0
20	–	–	–	–	–	–	–	–	–	–	–	–	–
21 TOTAL PAYMENTS	0	0	0	0	0	0	0	0	0	0	0	0	0
22	–	–	–	–	–	–	–	–	–	–	–	–	–
23 Opening Bank O' Dft	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400
24 Total Receipts	0	0	0	0	0	0	0	0	0	0	0	0	0
25 Total Payments	0	0	0	0	0	0	0	0	0	0	0	0	0
26 Closing Bank O' Dft	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400	1400

Lastly, once a range of options has been worked out, they need to be looked at from a variety of points of view, normally covering their impact on:

a) profitability

b) cash flow

c) capital requirements

d) balance sheet.

This in turn brings about a need for a great deal of further work if the full picture is to be collated each time the plans are in any way changed, and further underlines the fact that such an exercise is unlikely to be completed adequately, if at all, with manual methods alone.

Solutions offered by computers: Computers offer a variety of solutions to planning problems. They are often referred to as 'number crunchers', and this is very much the job they do when used in any planning application. But the use of computers in planning still leaves a number of other problems unchanged. The relevant information for input to the planning activity still has to be available in a suitable form. Chaotic accounting and control systems will not be of much use in providing it. The onus is still on management to think up new projects, ideas and products to bring the plans to life. Assumptions still have to be made about everything from the likely rate of inflation to the possible markets.

Without innovative thinking, there can be no plans, computerised or otherwise. Before they can form part of the plans, projects and ideas have to be investigated and costed. There is still a great deal of this and other preliminary work to be done manually. The computer's role is to take over the calculations, and allow a very large number of options to be evaluated in a very short time.

Spreadsheet Packages

In recent years, the single most successful package developed for micro-computers is the spreadsheet planner application. There is now quite a range of spreadsheet packages for IBM PC including the original Visicalc, and several packages which offer more. Both MBA from the Context Management system and Lotus's 1-2-3 offer graphing capabilities as well as spreadsheet calculations. Multi-plan is another spreadsheet package offered by Microsoft.

All developments of spreadsheet extend the simple concept of column analysis paper consisting of a few rows and columns to the limits of available micro-computer capacity with facilities to edit and calculate rapidly (a common size is 63 columns and 254 rows). Each column is usually identified by a letter combination A, AA etc. and the rows by numbers 1 to 254. So any individual position in the worksheet or cell can be identified by its column letter and row number e.g. A1, B24 etc. This facility enables relationships between cells to be defined e.g. cell A4 equals sum of cells A1 and A2 and A3 or cell D4 equals A4 nultiplied by B4. Once a data model has been built up from such relationships, data can be inserted in some cells and the dependent values will be automatically calculated and made available for display. Any change to an entry value will be reflected immediately in regenerated results. The range and complexity of spreadsheet planning models is enormous and its flexibility in business applications is only constrained by the imagination of the user. A sample spreadsheet is shown in Figure IV.

Interfacing accounting and planning software: There is one way in which computers can simplify the information-gathering process itself. This is where computerised accounting systems are being operated in the company where the planning is to be done. But obtaining information from such a source is not as simple as it might appear at first sight.

There are plenty of micro-computer packages (pre-written programmes or 'software') available for accounting work, covering such things as sales, general ledgers, invoicing, stock control and sales order entry. If the same micro-computer is to be used for both planning and accounting, a special link will normally have to be made between the two types of application (accounting and planning) in order to pass the data from one to the other. This 'interface' will normally be in the form of a programme, and the programmer will require detailed knowledge of how the accounting programmes are structured in order to know where the required information is held and what form it is in and to work out how to get at it. As the vendors of package software are usually reluctant to divulge such details for fear of piracy, the interface will often need to be programmed by the suppliers of the accounting software. They, in turn, may be unwilling to undertake such work or it may be too expensive. Either way, they will have to become familiar with the planning software as well. Such links, therefore, are not altogether straightforward, although some of the publicity material put out by the micro-computer manufactures can lead one to believe that the whole area is quite simple. If, however, the accounting software includes some form of 'report generator', this may provide the basis of a solution that is easier to attain in practice. This type utility programme is designed to extract data from files of information in a form specified by the user, and so it may be possible to construct a file of accounting data that can be passed across to the planning.

Activity C

Develop an operational flow-chart of computerised financial planning systems in your organisation.

....

....

....

....

....

....

....

....

....

....

6.5 SUMMARY

This unit has attempted to introduce various computer based financial packages.

We have identified various computer based application systems and discussed in detail about general ledger systems. We have also identified some problems encountered with their implementation.

Later we have examined computerised financial planning software and spreadsheet packages. We have also discussed problems of interfacing accounting and planning software.

6.6 SELF-ASSESSMENT EXERCISES

1 How the financial systems, facilitated by computer software, are important for economy of data processing and administrative efficiency?

2 Computers are often referred to as 'number crunchers'. Evaluate the statement in the context of financial planning.

6.7 FURTHER READINGS

Anderson, R.G. 1979. *Data Processing and Management Information Systems,* Macdonald and Evans: London

Best, P. and P. Barrett, 1984. *Auditing Computer Based Accounting System,* Prentice-Hall: Englewood-Cliffs.

Clifton, H.D., 1986. *Business Data Systems,* Prentice-Hall: Englewood-Cliffs.

Hanesien, W.D. and J.L. Camp, 1982. *Business System for Micro-computer,* Prentice-Hall: Englewood-Cliffs.

Lambourn, S. 1974. *Computer Application in Business,* Longman: Harlow (Essex).

Pocock, M.A. and A.H. Taylor, 1984. *Handbook of Financial Planning and Control,* Gower Publishing Co.: London

UNIT 7 COMPUTER BASED INVENTORY SYSTEMS

Objectives

After studying this unit you should be able to:

- define what is meant by inventory and its characteristics
- identify the purpose of computerised inventory systems
- design inventory system files and their relationships
- to know what type of software packages are readily available in the market for inventory management.

Structure

7.1 INTRODUCTION

When we go to a super market, we take it for granted that the shelves will be well stocked with the goods we want. We expect the produce to be fresh, in good condition and reasonably priced. Most of the time, this is what we find.

However, the problems of keeping a super market supplied with the right goods, in the right quantities, in the right condition and at the right prices are immense. Goods are bought from a large number of suppliers. Some are delivered in bulk to warehouses by the dealers, while the perishable goods must be sold the same day they are delivered. All goods have limited shelf life, and most are subject to health law regulations. As problems are caused by overstocking a super market or a warehouse as by understocking, any goods which have not been sold by their expiry date must be destroyed resulting in wastage.

The costs of transporting goods from one place to another are very high. It is important that delivery vehicles are fully loaded on each journey, and that the routes they take are as short as possible. The costs must be kept to a minimum in order to keep prices down. Small price increase can lose large numbers of customers. For these reasons, large super markets have to use computers for their inventory control.

This is a simplified example of inventory control as applied to the case of a super market. However, this becomes very complicated in case of a different situation, say, production environment.

7.2 WHAT IS INVENTORY?

The terms 'inventory' and 'stock' are usually seen as being synonymous and are used to describe materials which can be identified at various stages of the transformation process in organisations. It is customary to divide inventory into three categories:

- Raw materials including components and energy
- Work-in-progress
- Finished goods.

It will be evident that one organisation's output of finished goods may be another organisation's raw material input. Statistics indicate that all over the world, in manufacturing and distribution trade, raw materials, work-in-progress, and finished goods represent approximately 25, 25 and 50 per cent respectively of total stock value. Though the effect on an economy of aggregate stock levels is significant, the issue of inventory management is primarily an organisational one.

Usually, the consequences of the physical presence or absence of stock are expressed in money terms, and it is in this respect that the effect of successful inventory management is measured.

The above apportionment of stock across raw materials, work-in-progress, and finished goods conceals the fact that there are marked differences according to industry, and organisations within industries. A case can be mentioned of capital goods industry where 60% of the total stock is raw materials and compare this with the consumer goods industry where only four per cent of stock falls in this category. Indeed, in this latter case a fourth area, the distribution system, was estimated to hold 56 per cent of total stocks. Evidently, the particular circumstances of the market will influence both the size and disposition of stocks.

As is the case with much of the operations area, accurate meausurement of inventory is possible. The extent to which these are translated into accurate records depends on the systems used to capture and store data. The final step needed to provide a sound basis for effective inventory management is the manner in which the data are processed and presented for interpretation.

7.3 THE CHARACTERISTICS OF INVENTORY

A number of features of inventory, most of which are measurable, can be identified. An understanding of the following characteristics is an essential basis for inventory management.

Type

As stated above, three types may be recognised:

i) Raw materials: these include components, sub-assemblies, and energy.

ii) Work-in-progress: which is to be found at various stages in the production process, and which implies a measure of added value.

iii) Finished goods: these represent the output of the production process which remain until they are disposed of (ideally by sale to a customer).

Function

When inventory moves through the manufacturing process it serves different functions. Most importantly, it provides the Working Stocks so that there will be enough items for manufacture and sales. This is also frequently referred to as Lot Size Inventory because of the usual need to order or provide in lots or batches. When the most economical lot size is used, it is referred to as the Economic Order Quantity (EOQ) or Economic Batch Quantity (EBQ). Management must also take account of the fluctuations resulting from demand and lead time by establishing Safety Stock. This is also referred to as reserve, buffer, fluctuation, or stabilisation stock. These functions will be examined at greater length below. If the firm sees prospects of increased sales or supply shortages or if the supply is subject to marked seasonal effects, it may increase its inventories with what is known as Anticipation Stock. The final function is Transportation or Pipeline Stock, which reflects a product's movement during the manufacturing process or shipment to a customer.

Demand

Demand is a measurement of the total requirement for a given item over a specific time period. However, inventory demand should be differentiated from sales or usages. For example, sales only measure those orders satisfied, but do not include demand lost because of stock-outs. The importance of this difference is readily observed in the preparation of a demand forecast. The accuracy of this forecast requires data based on the total historical demand which includes both the amount supplied and the amount back ordered or lost. While the actual differences between demand and sales or usages for most firms are small, an examination of the historical data must be made to determine if the difference is significant.

Management must not only study the rate of demand, but the nature of demand as well. Autonomous demand which originates from customer orders receives much more attention than the demand which arises when higher-level assemblies are required. In this case components are used only when actual assembly takes place, generally at intervals, as opposed to the more continuous rate of depletion of autonomous (independent) demand. Examination of the movement of stock items indicates that the pattern of independent demand varies from being almost constant to highly irregular.

Lead Time

The length of lead time directly affects the size of the capital investment in inventories. Inventory lead time can be broken into four periods: review, order, manufacturing or purchasing, and receiving.

The Review Period is the time period between reviews in a fixed interval system. This period is not found in an order point system. The Order Period represents the actual time required to review inventory status and prepare the appropriate order. The Manufacturing Period includes order processing and schedule time, any queue time, machine set-up time, plus the actual manufacturing time. The Purchasing Period includes the time involved in processing a purchased order plus the elapsed time between a vendor's receipt of an order and its actual arrival at the manufacturer's facilities. The last period is the Receiving Period, which represents the time needed to process incoming order. This includes inspection and storage time, plus the time required to log the order into the inventory control system, so that it may be used by manufacturing.

Cost

There are three types of inventory cost, which may be involved in balancing the conflicting objectives for inventory.

The first cost is the Ordering or Set-up Cost and includes the following bought-in-items:

i) The cost of the actual purchase requisition by inventory control.

ii) The cost of issuing a purchase order and any follow-up time (by the Purchasing Department).

iii) The cost of the Receiving Department's order processing (clerical etc.)

Set-up costs for manufactured items include the following:

i) The cost of generating the manufacturing requisition by inventory control.

ii) The cost of planning, scheduling, loading and expediting the item by production control.

iii) Machine Set-up time.

iv) Any one-time cost (start-up, scrap, etc.).

The second cost is the inventory carrying cost, and includes all the expenses required to maintain a volume of inventory. This can be expressed as a proportion of the value of the inventory per unit of time.

The third cost is the Stock-out Cost, which includes all expenses from running out of stock. If expediting is not used, the stock-out would represent any backorder cost plus loss of profits due to lost sales. The backorder costs include any extra paper work that might have been required, loss of return on capital not received during the stock-out, and loss of customer goodwill resulting from failure to fill the order.

Although stock-out costs and customer service are of great importance the former are difficult to calculate and as a result such estimates are made infrequently. The normal attempt to control stock-outs is through the adoption of service levels, although, as will be seen below, this approach involves an implicit estimation of the cost of being out of stock.

7.4 PURPOSE OF COMPUTERISED INVENTORY CONTROL SYSTEM

The main purpose of the inventory control is the monitoring of the stock levels and this is achieved by recording stock movements on stock records. Through these means the current inventory situation, including shortages, is available.

It must be appreciated that inventory forms a buffer to facilitate variations in supply and demand. As it often consumes a high proportion of working capital, it must be controlled in the most effective manner to achieve the following:

Avoidance of excessive stocks

Avoidance of excessive stock shortages

Optimising cost of ordering supplies

optimising cost of carrying inventory.

Inventory Control parameters

The control of inventory is achieved not only by recording inventory movements, but also by means of control parameters for each item of inventory. Examples of such parameters are:

Re-order level

Economic order quantity

Safety stock

Maximum stock
Average stock.

Inventory Applications

Because the manufacturing process in many organisations is highly cor..plex involving large inventory stocks, many departments and users also account for the largest portion of product cost, and even small improvements in its resourre utilisaticn can generate large profits. Reduction of work-in-progress inventory, eliminauon of duplicate holdings and order status information that permits the identification and expediting are some of the benefits gained from the computerised inventory systems.

Table 1 given below describes the kinds of computerised inventory applications used in order of increasing complexities. A prerequisite for computerised inventory system application is an item-coding that can be used for all sub-systems of inventory, bills of materials and purchasing records. At the simplest stage inventory accounting systems are implemented, followed by requirements generation (bills of materials) and physical inventory systems.

Table 1
Computer Based Inventory Applications

Least Complex Application	....	....	**Most Complex Application**
Weekly stock reporting with daily activity listings	Use Reporting	Requirements forecasting	Continuous updating of inventory
Item catalogue preparation	Requirements reporting	EOQ calculations	Automatic item status reporting
ABC Analysis (distribution by Value)	Time series requirements reporting		Automatic replenishment initiation (Purchase Orders)
Net return analysis	Listing of procurement requirements		
	Cycle count physical inventory accounting		

7.5 INVENTORY SYSTEM DESIGN-AN EXMPLE

Most modern businesses require some method of inventory control, whether they are involved in outside sales or need to monitor equipment and supplies for internal use. Beginning with this, we will design files for such an inventory system for a retail sales enterprise. This will give us an opportunity to know about inventory system files and data relationships.

General Design of Inventory Systems

Inventory systems generally require the use of several data files. One file, usually referred to as the master file, keeps track of the quality of each item presently in stock. In addition, it may also keep track of the location of each item in the warehouse, the cost of the item, the reorder point, the quantity currently on order, and the name and address of the vendor the item is purchased from. This allows the user of the system to locate an item quickly, get a listing of goods that need to be ordered, create and print new orders automatically and perform other similar stock-control tasks.

Most businesses also like to keep track of each individual transaction that occurs within the business. Information on these individual transactions is usually maintained in separate data files called **transaction files.** One such file might keep track of individual sales transactions: to whom items were sold, when, for how much, and the invoice or receipt number. A second transaction file might maintain an on-going record of all new stock received. We can see the relationship between the master file and two such transaction files in Figure I.

The transaction files maintain a history of all individual transactions involving the sale and receipt of goods, while the master file maintains the current status of goods in stock at any given moment by using information supplied directly from the two transaction files. In a sense, the master file is an up-to-date summary of all activities in the two transaction files.

Key Fields

Whenever we design a system involving multiple data files such as these, it is very important to define a key field that can be used to relate the various files to one another. A key field has the following characteristics:

- It is unique for every record (item) in the master file.
- It exists in the master file and in the transaction files with the identical field name, type, and width.

Figure I: Relationship of typical inventory system data files

MASTER FILE
Records:
Part number
Item name
Amount in stock
Cost
Amount in order
Location in warehouse
Vendor Information
Order Date

SALES/ISSUES TRANSACTION FILE
Records:
Part Number
Quantity sold/issued
Selling price
Date sold/issued
Customer name
Clerk name

NEW STOCK TRANSACTION FILE
Records:
Part number
Quantity received
Purchase price
Data received
Vendor

The key field in most software systems is an identifying code. In an inventory system such as the one we are designing here, this code is usually a part number: each item the store carries has a unique part number. Without the use of key fields, data can be very difficult to manage.

Inventory System Problem Definition

Our Inventory System will be designed to allow a small business to keep track of all goods in stock, goods on order, and the location of the goods in the store. Whenever the store decides to carry a new item, the item will be assigned a unique part number and added to the master file. The system will provide the user with reports of all goods in stock, items that need to be reordered, and items on order, and will automatically produce purchase orders.

The system will also allow the user to keep track of all individual sales transactions: items sold, to whom sold, salesperson, date of sale, and invoice number. Similarly, the user will be able to keep track of all incoming goods as stock is replenished. The individual transaction records from sales and goods received will be used to automatically update the stock status records in the master file.

Inventory System I/O Specifications: The primary output from the Inventory System master file will be reports concerning the status of the inventory, and purchase orders for restocking. A current-inventory report will display the part number, part description, quantity on hand, purchase price, location in warehouse, and quantity on order for each item in stock. A record report will list all those items that need to be reordered, and an on-order report will list all items currently on open order. Finally, the system will actually print the purchase orders when the user is ready to restock.

We will need at least the following input in the master file to develop all these reports.

Part number
Item description
Quantity in stock
Purchase price
Reorder point
Quantity on order
Location in warehouse
Vendor s name
Vendor's address
Date of last update
Date of last order
Quantity to order.

The system should produce a receipt for each sales transaction. The receipt should contain the part number, quantity, invoice (receipt) number, saleperson's name, customer's name, selling price, and date. So the sales transaction file will need to include the following:

Part number

Invoice number
Salesperson's name
Customer's name
Quantity sold
Selling price
Date sold
Whether posted.

And finally, the system must keep track of goods received, and produce a report showing the part number, quantity received, purchase price, date received, and name of vendor. So the new-stock transaction file will need to include:

Part number
Quantity
Purchase price
Date received
Vendor's name
Whether posted

Inventory System Files Design

We will use three data files to manage the inventory, the master file and two transaction files.

The Master File: The structure of master file, with a brief description of each field's contents, is shown in Figure II. You can use this information to build the Master File.

Figure II: Structure for the Master data file

Field name		Width	Contents
1	Part-No	5	Part number (key field)
2	Title	20	item description
3	Qty	4	Quantity in stock
4	Cost	9	Purchase price
5	Reorder	4	Reorder point
6	On-Order	4	Quantity in stock
7	Location	5	Location in warehouse
8	Vendor	25	Vendor's name
9	Vendor-add	25	Vendor's address
10	Vendor-CSP	25	Vendor's city, state, PIN
11	Date	8	Date of last update
12	Order-date	8	Date of last order
13	New-order	4	Quantity to order

The Sales File: Next we will create a data file to keep track of individual sales transactions. Notice that in the **Master** file just created, the key field named Part-No. has one character of data type, and is five characters wide. Since this will be the field used for updating the Master file, the Sales transaction file must have an identical key field. The structure of Sales File is shown in Figure III. Field 8, posted, is a special field that will be used to determine whether or not a given transaction has already been recorded in (or posted to) the master inventory file, to keep transactions from accidentally being posted twice.

Figure III: Structure for the Sales data file

Field name		Width	Contents
1	Part-Nunber	5	Part number (key field)
2	Invoice-No.	6	Invoice number
3	Clerk	12	Sales person's name (or code)
4	Customer	12	Customer's name
5	Qty	4	Quantity sold
6	Price	9	Selling Price
7	Date	8	Date sold
8	Posted	1	Posted to master file yet?

The New Stock File: The third file in the system will be used to keep track of goods received to replenish stock. Again, since the master file will be receiving information from this file, we will need to include the key field Part-No. The structure of **New Stock** file is shown in Figure IV.

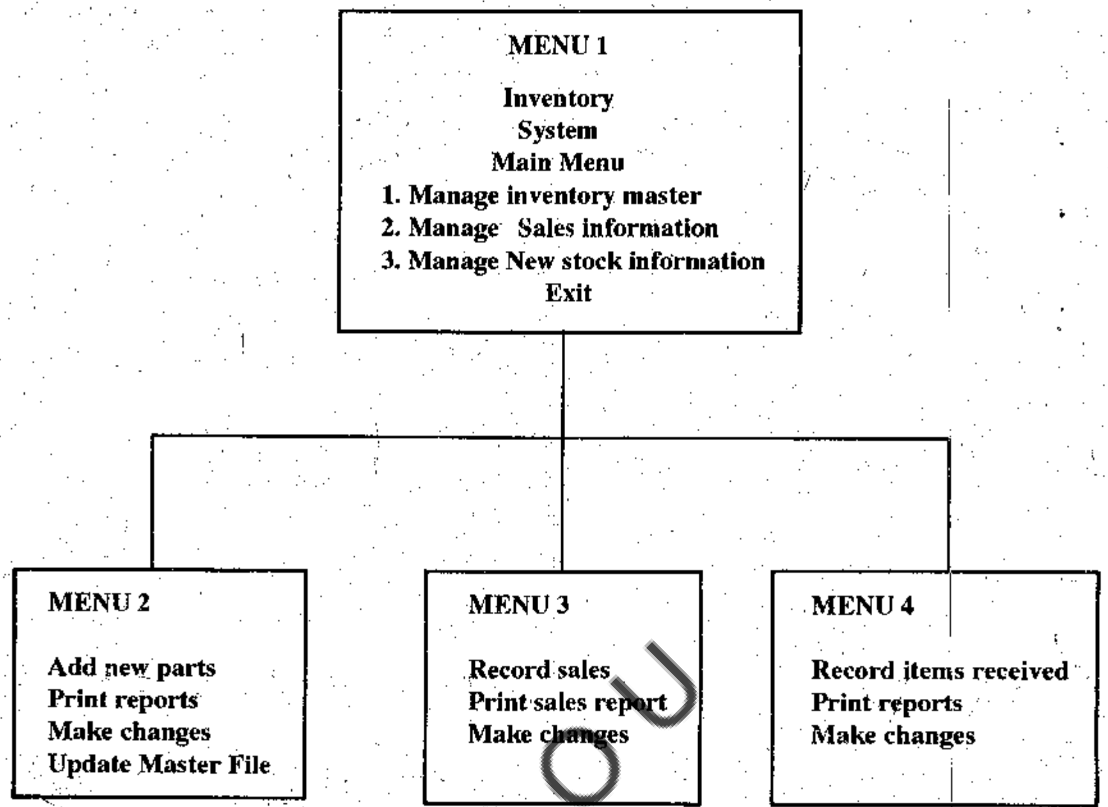
Figure IV: Structure for the New stock Data file

Field name		Width	Contents
1	Part-Number	5	Part number (key field)
2	Qty.	4	Quantity received
3	Cost	9	Purchase price
4	Date	8	Date received
5	Vendor	25	Vendor's name
6	Posted	1	Posted to master file yet?

Now that we have designed the Files for our Inventory System, we need to design the input menu (Screens).

We want to design the system to allow different individuals to perform the various major tasks. That is, the store manager will be responsible for managing the master inventory, placing orders, checking the status of goods, and so forth. Individual sales transactions will be entered by sales clerks at the point of sale, and incoming stock will be recorded by a stockroom clerk. In a sense, we will be using three separate but related systems, with the overall structure shown in Figure V.

Figure V: MENU sub-systems



The inventory system can be simplified and its functions can be represented by four menus shown in figure V. the Menu 1 shall be overall inventory system menu. Depending on user's choice it would activate programme/sub-programme which would result in display of secondary level of menus e.g. menu 2, menu 3 or menu 4.

The main menu 1 would give the user access to Inventory, Sales or Stock master files. Use of secondary level menu shall provide user option to record new parts, print reports, amend the existing records. Suitable programmes can then be developed to perform the above desired functions.

Activity A

Prepare a flow diagram of the Inventory system in your organisation. Examine it in context of the I/O specifications and file design system described above.

7.6 SOFTWARE PACKAGES IN INVENTORY MANAGEMENT

Table 2 lists some of the computer software packages which have been published in whole or in part, to deal with inventory management. The rates of introduction of new packages and modification of existing packages are high, so the details in the table are only indicative and only outline of the types of packages available is given.

The packages listed in the table are mainly derivatives of earlier software and are designed for implementation on mainframe computers. In general these packages offer a wide range of facilities, in addition to inventory management. Such commercially available mainframe packages often provide planning and control routines, purchase order processing etc., as well as inventory management.

Table 2

Name	Supplier	References
OMAC	ICL	
PCS	Burroughs	*Jour. of Industrial Engg,* March 1977
MAPICS	IBM	
IMS/66	Honeywell	*Jour. of Industrial Engg.* Jan., 1977, 79
CYBER	Control data	*Jour. of Industrial Engg.* Dec., 1977, 79
Industrial Management System 201	Mod Comp Business Systems	*Jour. of Industrial Engg.* Feb., 1977, 79
GIPSH Inventory Control and Replenishment system	Aston University	*International Jour.* of Prod. Reasearch 1980-81

There is, however, a move towards the development of micro-computer based yet comprehensive inventory management packages which provide not only for inventory record-keeping but also for decision making in uncertain conditions against required service levels. An indicative list of inventory management software available locally is also given below:

Name of the Package	Supplier	Operating system/Language
Inventory stock Accounting and Control System	Uptron Ltd.	UDOS, B-BASIC
Inventory Control	Par Computers Ltd.	MSDOS, CP/M, DBOS
Inventory control Computerised System	Datamatics Consultants Ltd.	Wang OS 6.3
Inventory Management System	Softscan Corpn.	APPLE-PASCAL
Inventory Module	Maegabyte Consultancy Pvt. Ltd.	CP/M, COBOL
Inventory Control	Eiko Sales Pvt. Ltd.	CP/M, BASIC
Inventory Monitoring System	NELCO	CP/M, BASIC
Inventory Management	IDM Ltd.	CP/M, MP/M, COBOL

There are a number of other inventory software available currently in India. Details of these can be obtained from software houses, computer dealers and other computer journals published. Most of the micro-based software are interactive and menu driven. They provide facilities for on-line entry of transaction and masters, validation of input data, stores ledger and stock status, consumption analysis and summary, 'exception reporting' for minimum level and reorder level.

Activity B

Briefly review the computerised Inventory Control System, evaluate their advantages and limitations in the context of your organisation.

....

....

....

....

....

....

7.7 SUMMARY

This unit attempts to help you to understand computer based inventory application systems better.

We have defined the term inventory and its characteristics. Various computer based inventory applications have been discussed in terms of their complexities.

An example has been provided to design computerised inventory system files to establish data relationships. The example can be further developed by you to incorporate various reporting for outputs to the users.

A small indicative list of inventory packages has also been provided to give you an idea of what is readily available in the market.

7.8 SELF-ASSESSMENT EXERCISES

1 State and explain 'Inventory' and discuss its main features. Describe Lead Time and Demand forecast.

2 Read the following statements. Write True (T)/False (F) in front of them.

   i) Demand is a measurement of the total requirement for a given item over a specific time period;

   ii) Review, order, manufacturing and receiving periods are the part of demand forecast;

   iii) Order period represents the actual time required to review the inventory status and prepare the appropriate order;

   iv) Review period is the time period between reviews in a fixed interval system;

   v) Set up costs for manufactured items does not include the cost of planning, scheduling, loading and expediting the item by production control.

3. Discuss various applications of Inventory and its file design system.

Answer Key

Question No. 2

(i) T (ii) F (iii) T (iv) T (v) F

7.9 FURTHER READINGS

Clifton, H.D., 1986. *Business Data Systems,* Prentice-Hall: Englewood-Cliffs. (Chapter 10).

Lewis, C.D., *Scientific Inventory Controls,* Butterworth: London. (Chapter 8).

Moskowity, H. and G.P. Wright, *Operation Research Techniques for Management,* Prentice-Hall: Englewood-Cliffs. (Chapter 15)

Simmons, D.M., Optimal Inventory Policies Under a Hierarchy of Set-up Costs, *Management Science Applications,* 18, 10 June, B591-B599.

UNIT 8 COMPUTERS IN HUMAN RESOURCE MANAGEMENT

Objectives

The objectives of this unit are:

- to explain the various computer based information systems for human resource management
- to understand human resource management system needs
- to appreciate the main requirements for these systems.

Structure

8.1 INTRODUCTION

Every company has to develop a personnel information system to meet its own requirements. A system cannot simply be inserted into a company and expected to produce desired results. It is necessary first, to define what the system is intended to do, and then to examine what records and systems are needed to achieve this.

The purpose of a system will vary enormously according to the objectives of the company and may well have to be decided by senior management, leaving the application to the individual operating managers who will need to define their own information requirements within the broad framework of overall objectives. These could be any of the following:

- To support wage and salary administration
- To facilitate selection and appointments
- To help identify high potential people
- To determine in-company promotion
- To help in the administration of management development and training
- To help and develop full potential of employees and department.

The objectives of a system will change in the process according to circumstances. In the past, for instance, determining wages has been the main purpose of employee information. Today, such things as records of skills for manpower planning have begun to assume equal importance.

8.2 THE SYSTEM NEEDS

Human resources management information systems are required for three main purposes:

1. To store the personal details of individual employees for reference.
2. To provide a basis for decision-making in every area of personnel work, especially in
   - manpower forecasting and planning
   - recruitment and selection
   - employment, including promotion, transfers, disciplinary procedures, termination and redundancy
   - education and training
   - pay administration
   - health and safety.
3. Supplying data for returns to government departments.

Personnel records and information procedures can be based on an entirely manual system but, increasingly, they are being computerised to a greater or lesser degree. The advent of micro-computers is accelerating this process.

A comprehensive system of records covers all the information required about individual employees or needed for personnel decision-making. But the information must be relevant. Every piece of information must be challenged with the questions 'What purpose will this serve?', 'To what use will it be put?'. The first point to clear when setting up a record system is the objective of each item in terms of the decisions it will help to make, its contribution to the assembly of essential statistical information, or its importance as a reference point in dealing with matters affecting individual employees.

It is necessary to avoid any gaps in information which is essential for decision-making. It is equally necessary to avoid gathering useless data or maintaining elaborate statistics to which no one ever refers. Too often, a 'one-off' request for information leads to the setting up of a permanent record or data collection system, although the information may never be requested again. Regular reviews should be made of all records and returns to ensure that they are serving useful purpose and that they are generally cost-effective. It may be cheaper in some circumstances to maintain manual records raterh than to computerise. It may be less time-consuming and costly to carry out a special exercise rather than to maintain a permanent record.

Identifying information requirements

The starting point should be an analysis of the decisions that the company makes or may need to make about individuals or groups of employees or the work force as a whole. This should be followed up by an analysis of the information required by government departments and agencies and by employers.

Human Resource Management decisions requiring statistical data

The main decisions for which statistical information or individual data may be required are:

- Forcasting the future supply of manpower by analysing, for each category of staff, labour turnover, age distribution, absenteeism and promotions

- Forecasting the future demand for manpower by ratio-trend analysis (calculating current ratios of manpower to activity levels and forecasting future ratios by reference to projected activity levels) and other statistical means
- The introduction of productivity improvement or cost reduction campaigns based upon analysis of present manpower productivity levels and costs (e.g. manpower cost per unit of output, or the ratio of manpower costs to sales turnover or profit)
- Planning recruitment campaigns on the basis of analysis of the results of previous campaigns, especially sources of recruits, media costs and success rates, and the relative pulling power of different inducements and recruitment methods.
- Introducing new or improved interviewing and testing techniques on the basis of comparisons between interview and test assessments and subsequent performance
- Identifying people with particular skills or potential for new appointments or promotion
- Improving disciplinary procedures or amending work rules by analysing disciplinary cases
- Introducing new or improved time-keeping methods or considering the introduction of flexi-time by reference to time-keeping records
- Planning redundancies-consulting unions, transferring or retraining employees, selecting employees for redundancy, helping to place redundant employees
- Planning training programmes-subjects to be covered, types of courses and number of courses-by reference to analysis of future changes in manpower (numbers and skills), performance review records and job and training specifications
- Taking steps to improve job satisfaction and morale by reference to statistics on labour turnover, absenteeism, sickness, accidents, discipline cases and grievances
- Changing pay systems on the basis of statistics of price index fluctuations in earnings, the proportion of employees on average earnings rather than payment-by results, cost per unit of output, fluctuations in earnings, the number and consequences of arguments over job rates
- Reviewing pay structures and levels of pay by reference to statistics of earnings in the company, rates of pay elsewhere, and the distribution of rates in each pay grade
- Controlling performance reviews by analysing the distribution of merit awards in relation to budgets and guidelines
- Taking steps to improve employee relations by analysing the causes of disputes
- Determining the information that should be communicated to unions and employees about the company or to assist in negotiations and joint consultation
- Improving health, safety and fire precautions by analysing reports on industrial disease, accidents and dangerous occurrences, monitoring returns on exposure to health hazards in relation to predetermined threshold limits, and studying reports on health, safety and fire inspection, spot-checks and audits.

Individual data

Individual information should include:

- The application form giving personal particulars
- Interview and test record
- Job history after joining the firm including details of transfers, promotions and changes in occupation
- Current pay details and changes in salary or pay
- Education and training record with details of courses attended and results obtained
- Details or performance assessments and reports from appraisal or counselling sessions
- Absence, lateness, accident, medical and disciplinary with details of formal warnings and suspensions
- Holiday entitlement
- Pension data
- Termination record, with details of exit interview and suitability for re-engagement.

Collective data

Collective information may include:

- Numbers, grades and occupations of employees
- Absenteeism, labour turnover and lateness statistics
- Accident rates
- Age and length of service distributions
- Total wage and salary bill
- Wage rates and salary levels
- Employee costs
- Overtime statistics
- Records of grievances and disputes
- Training records.

8.3 DESIGNING THE SYSTEM

The type and complexity of the personnel records of information system must obviously depend upon the company and its needs. Small companies may only need a basic card index system for individual employees and a simple set of forms for recording information on numbers employed, labour turnover and absenteeism. But a larger company will almost certainly need a more complex system because more information has to be handled, many more decisions have to be made, and the data changes more often. Card indexes are not enough, because supplementary records may be needed to give more detailed information about individual employees.

The key decisions to be made when designing the system concern:

- The design of the basic records, forms and input material
- The use of computers
- The extent to which records should be centralised or decentralised
- The procedures and programme for collecting, recording, updating and disseminating information.

The use of Computers: Advances in computer technology mean that relatively small firms can use computers, not only to store data but also to generate information for decision-making purposes. When thinking about converting manual records to a computer system or updating an existing system, the points to consider are:

1. Why computerise?
2. To what uses can a computer be put?
3. What systems should be selected and how?
4. How should the system be operated?

Computers will hold records in a more compact and accessible way, and they can be justified for this reason alone. They can generate information for decision-making more flexibly, more quickly and more comprehensively than any manual system.

8.4 USES OF COMPUTERS IN HUMAN RESOURCE MANAGEMENT

The main uses to which computers can be put in human resource departments are:

Keeping records: Replacing card indexes and filing cabinets by magnetic discs.

Listings: Quickly providing listings of employees by department, occupation, grade, pay level, length of service, age, sex, qualifications, skills, etc.

Automatic letter writing: Producing standard letters and forms for recruitment, promotion, transfer, upgrading, appraisal, pay review and new contracts of employment.

Manpower planning: Using manpower data to forecast the future demand and supply of people. Manpower models can be used, for the following: **forecasting model for** examining a hierarchy of grades by age. Useful for gaining a general understanding of future movement of staff and changes in age/grade structure.

Labour turnover analysis: Providing labour turnover statistics.

Career development: As a development of manpower planning models, computerised personnel information can be used to improve succession planning.

Recruitment: The computer can, in effect, be used as a filing cabinet to store details of each applicant, date of receipt of application, when called for interview and the outcome. If an applicant contacts the company, he can be informed of the progress of his application. Managers can be given details of the number of applicants and how many have been interviewed. Lists and automatic letters can be produced when calling for interview rejecting applicants or making offers.

Training: Records can be kept to check on who has received training or on progress through other training schemes. Listings of skills and qualifications by department or occupation can be produced to identify gaps and training needs.

Pay: Information can be drawn from both personnel and pay-roll systems to analyse pay-roll costs and ratios and to assess the impact of various pay increase options on the pay stucture and on total pay-roll costs. Budgetary control systems can be computerised to show actual pay-roll costs against budget and project future costs.

Salary administration: Salary analysis reports can be produced which give information by employee on occupation, salary position in salary range, total percentage increases over previous years and annual appraisal. Individual forms and departmental schedules can be generated for salary reviews and analysis can be made of the salary structure (e.g., comparisons).

Job evaluation: Data bases can be created to hold and process the information on job evaluations, such as grades and point scores. Weightings of job evaluation factors can be computerised. In a job evaluation exercise, the information system can be used to point out the names of those whose jobs are to be evaluated. Details of job, grade, function, location, sample size and current point ratings can be programmed in. The data base can link together similar posts in different parts of the organisation. Listings of all gradings, re-gradings and points/scored can be produced.

Absence and sickness: Absence and sickness can be recorded by employee, with reasons and analysis can be produced of absenteeism and sickness.

Health and safety: Records can be maintained on accidents and absence due to health hazards. Trends can be analysed and information produced on who has worked in certain areas, or who has used certain processes and for how long.

Activity A

Critically evaluate the use of computers by your organisation for Human Resource Management decision-making, in the context of the diverse uses described in this unit.

Activity B

What other activities would you like your organisation to take up for computerisation?

....

....

....

....

....

....

....

....

....

....

....

....

....

8.5 SORTWARE PACKAGES IN HUMAN RESOURCE MANAGEMENT

Table 1 given below describes the kind of applications used for human resource management information system in order of increasing complexities. Historically, the payroll function was often the first administrative operation to be computerised. The highly repetitive nature of the job with well defined rules of computation, made it a logical target for automation, particularly since large number of clerical personnel were required to perform the function.

Table

Computerised applications for human resource management system

Least complex Application .... **Most Complex Application**

• Pay-roll preparation	• Automation of basic Personnel data	• Salary Analysis	• Skills Inventory
• Cheques preparation	• Preparation of scheduled reports e.g.	• Performance appraisal	• Manpower planning analysis
• Production of required reports	- Seniority - Vacations	• High-Flier tracking	

Table 2 gives an indicative list of sofware packages available in the market. There are a number of other software packages available in India for wide range of computer systems. Details on these can be obtained from software houses, computer dealers and other computer journals. Most of the above micro-computer based software packages are interactive and menu-driven which can also be customised to a limited extent at the customer's site.

Table 2

Indicative List of Software Packages

Name of the package	Supplier	Operating system/Language
Personnel Inventory	Mantec Consultant	CP/M, DBASE-II
Pay-roll Accounting	....	....
Skills Inventory	MECON	COBOL
Manpower Planning	MECON	COBOL, FORTRAN

8.6 SUMMARY

This unit should have given you an understanding of a Human Resource Management Information System in an organisation. The unit identifies the specific needs and describes various types of data and software packages which might be usefully applied in a Human Resource Management Information System.

It also explains the main uses and design considerations of the computers relevant for an effective and efficient Human Resource Management Information System.

8.7 SELF-ASSESSMENT EXERCISES

1 How does computerised personnel Information System help in managing Human Resource more efficiently?

2 Evaluate your organisation's Human Resource Management Information System and suggest how to make it more efficient.

8.8 FURTHER READINGS

Edwards, C., 1982. *Developing Micro-computer Based Business Systems,* Prentice Hall: Englewood-Cliffs.

Moris, R., 1984. *Computer Basics for Managers,* Business Books Ltd.: London.

Perspective in Manpower Planning, Institute of Personnel Management (I.P.M.): London.

Springall, J., *Personal Records and the Computer,* Industrial Society/IPM: London.

Wille, E., 1981. *The Computer in Personnel Work,* IPM: London.

BLOCK 3 COMPUTERS AND DECISIONAL TECHNIQUES

Having introduced the computer applications in various functional areas in Block 2, we will now describe specific quantitative techniques of decision-making in which computers are used rather extensively. One of the most important disciplines for quantitative decision-making is popularly called Operations Research (OR). The first unit of this block will introduce the diverse applications of OR in managerial decision-making. It also enlists various OR techniques.

The second and the third unit describe in detail one of the most popular OR techniques, namely, Linear Programming. With the help of illustrative problems and exercises, variants of linear programming technique are explained.

In this block, we have added an additional section i.e. "do-it-yourself exercises." These problems can be solved manually as well as on computers. Large size problems requiring use of computers can also be made available to participants who have access to computer facilities.

All these factors, you will observe, have reduced the scope of decision-maker's ability of making decisions based upon his past experience and intuition only. Further, it is also important to know what constitutes an 'effective' decision. An effective decision depends on so many different economic, social and political factors and view points. For example, a government decision on the location of a new fertiliser factory would be based on economic factors such as construction costs, labour costs, taxes, energy and pollution control costs, transportation costs, etc. On the other hand, location of a new medical college may be influenced by central, state and local politics. Hence, understanding of the possible use of scientific approach (also known as the scientific method) to decision-making is of great importance to the business students.

Operations Research (also known as quantitative methods, management science, decision science and a few other names) can be viewed as a scientific method that has evolved as an aid to the decision-maker in all types of organisations.

Operations Research is assuming an increasing degree of importance in theory and practice of management. Some of the factors which are responsible for this development are:

i) decision problems of modern management are so complex that only a systematic and scientifically based analysis can yield realistic solutions.

ii) availability of different types of quantitative models for solving these comple managerial problems.

iii) availability of high-speed computers has made it possible both in terms of time and cost to apply quantitative models to all real-life problems in all types of organisations such as business, industry, military, government, health and so on.

Operations Research is not a fixed formula which can be applied to all types of problems. This requires that the problems be defined, analysed and solved in a rational, logical, systematic and scientific manner based on data, facts, information and logic and not on intuition and subjective judgment. However, Operations Research does not totally eliminate the scope of qualitative or judgment ability of the decision-maker. In actual practice, Operations Research is useful only if quantitative models can be built upon, and modified by the experience and creative insights of the decision-maker.

Activity A

Consider any major decision you made recently. Do you know how you made that decision? List the steps in reaching that decision.

....

....

....

9.2 HISTORY OF OPERATIONS RESEARCH

Operations Research came into existence during World War II, when the British and American military management called upon a group of scientists with diverse educational background namely, Physics, Biology, Statistics, Mathematics, Psychology, etc., to develop and apply a scientific approach to deal with strategic and tactical problems of various military operations. The objective was to allocate scarce resources in an effective manner to various military operations and to the activities within each operation. The name Operations Research (OR) came directly from the context in which it was used and developed, viz., research on military operations.

In the wake of World War II, the success stories of military teams attracted the attention of industrial managers in UK and USA who were seeking solutions to their complex executive problems. By 1948, it had taken hold in UK and was in the process of achieving the same in USA. By the early 1950s industries in USA had realised the importance of Operations Research in solving their management problems.

During the 50s, Operations Research achieved recognition as a subject for study in the universities. Since then the subject has gained increasing importance for the students of Management, Public Administration, Behavioural Sciences, Engineering, Mathematics, Economics and Commerce, etc. Today, Operations Research is also widely used in regional planning, transportation, public health, communication, etc., besides military and industrial operations.

In India, Operations Research came into existence in 1949 with the opening of an Operations Research Unit at the Regional Research Laboratory at Hyderabad and also in the Defence Science Laboratory at Delhi which devoted itself to the problems of stores, purchase and planning. For national planning and survey, an Operations Research Unit was established in 1953 at the Indian Statistical Institute, Calcutta. In 1957, Operations Research Society of India was formed. You may be aware that almost all the universities and institutions in India are offering the inputs of Operations Research in their curriculum in some form or the other. Industries, government and other agencies are gradually becoming conscious of the role of Operations Research in decision-making.

9.3 NATURE OF OPERATIONS RESEARCH

As its name implies, Operations Research involves research on (military) operations. This indicates the approach as well as the area of applications of the field. Thus it is an approach to problems of how to coordinate and control the operations or activities within an organisation. Following is an example which will elaborate the nature of Operations Research.

In order to run an organisation effectively as a whole, the problem that arises frequently is of coordination among the conflicting goals of its various functional departments. For example, consider the problem of stocks of finished goods. The various departments of the organisation may like to handle this problem differently. To the marketing departments, stock of a large variety of products is a means of supplying the company's customers with what they want, and when they want it. Clearly, a fully stocked warehouse is of prime importance to the company. The production department argues for long production runs preferably on a smaller product range, particularly if there is a significant time loss when production is switched from one variety to another. The result would again be a tendency to increase the amount of stock carried but it is also vital that the plant should be kept running. On the other hand, the finance department sees stocks kept as capital tied up unproductively and argues strongly for their

roads, etc., whereas the actual objects are complex and might not allow direct handling or manipulation. The objectives of constructing these models is to understand by analogy.

2 Mathematical Models

The complexity of relationship in some systems cannot be represented physically, or the physical representation may be cumbersome and take time to construct or manipulate. Therefore, a more abstract model is used with the aid of symbols. The Input-Output model of national economy involving several objectives, constraints, inputs and inter-linkages between them is a classic example of representing a complex system with the help of a set of equations. Mathematical models are extensively used in oil exploration. Using a set of data related to seismological characteristics of earth, it is possible to predict the presence of oil with reasonable probability.

All mathematical models generally comprise three types of variables: Dependent variables, Decision variables and Uncontrollable variables. The **dependent variables** reflect the level of effectiveness of the system. A dependent variable means that for an event described by this variable to occur, another event must occur first. The **decision variables** are those factors where a choice must be made. These decision variables can be manipulated and controlled by the decision-maker. These are classified mathematically as independent variables or unknown variables. The **uncontrollable variables** are those factors which are not under control of the decision-maker. These variables are also the independent variables since they affect the dependent variables.

Activity C

What is the type of the following 'models'?

i) Micro-filmed documents,

ii) Motion film,

iii) Frequency curves in statistics,

iv) Family of equations describing the structure of an atom, and

v) Flow charts in production control.

....

....

....

....

....

9.6 METHODOLOGY OF OPERATIONS RESEARCH

By using different Operations Research techniques, we generally attempt to arrive at an optimal solution of the problem based on some criteria or criterion for optimality. The term optimality is discussed in a significant way in this unit. The major emphasis in this study material is on quantitative measures of optimality (some times called measures of effectiveness) involving costs and/or profits. The general approach to solve a problem in Operations Research is as follows:

Step 1: Identifying the problem

Step 2: Constructing a mathematical model

Step 3: Solution of the model

Step 4: Testing the model and the solution

Step 5: Establishing controls

Step 6: Implementing the solution.

We will elaborate them in the following paragraphs:

Step 1: Identifying the problem: This is the first critical step because it sets the boundaries for all that follows. The identification of all significant interactions between the problem area and other operations of an organisation as a whole leads to the formulation of the problem. Each problem has its unique characteristics and may require different approaches to formulate it.

Step 2: Constructing a mathematical model: After establishing the criteria of optimality, determine the specific mathematical relationships (or models) which exist among all variables (controllable and uncontrollable). These are usually stated in the form of an equation (or a set of equations) or inequalities.

Step 3: Solution of the model: Substitute the values of uncontrollable variables in the mathematical model to determine the value of controllable variables (decision variables) that optimise the given objective function. The actual solution procedure will vary according to the type of model. When the value of any uncontrollable variable is likely to vary or is not known with certainty, the solution can be re-evaluated over a range of feasible changes in this variable

Step 4: Testing the model and the solution: A model is a particular representation of a problem under study, which in turn represents spesified aspects of reality. A model is said to be valid if it can provide a reliable prediction of the system's performance.

If the model cannot be tested prior to implementation, then perhaps it can be implemented in phases for validation. For example, a new model for inventory control may be implemented for a certain selected group of items while the older system is retained for the majority of items. As the model proves itself, more items can be placed within its jurisdiction. A model must be applicable for a longer time and can be updated from time to time taking into consideration the past, present and future aspects of the problem.

Step 5: Establishing controls: Control processes are necessary to ensure that the solution suggested by the model results in the predicted changes in performance. Beyond initial implementation, controls are necessary for maintenance of the solution.

Step 6: Implementing the solution: The decision-maker has not only to identify good decision alternatives but also to select alternatives that are capable of being implemented. This implies an assessment of the organisational climate for change and the decision-maker's abilities to move the organisation. The behavioural aspects of change are exceedingly important to the successful implementation of results. In any case, the decision-makers who are in the best position to implement results must be aware of the objective, assumptions, commissions and limitations of the model.

Activity D

List the major phases in the methodology of Operations Research.

....

....

....

....

9.7 VARIOUS OPERATIONS RESEARCH TECHNIQUES

A brief comment on certain standard techniques or prototype models of operations research which can be helpful to a decision-maker in solving the following classes of problems is given below. However, each one of these models of Operations Research involves detailed studies and in our context we are merely listing these to arouse your interest.

Allocation models: This class of models is concerned with the allocation of scarce resources so as to optimise the given objective function, subject to certain constraints. Procedures for solving such type of problems are collectively called mathematical programming problems. Mathematical programming problems are either linear or non-linear in nature. Unit 10 and Unit 11 have described the various methods of solving linear programming problems.

The distinction between linear and non-linear programming problems is based on the nature of the objective function and/or constraints. For example, linear programming is used in the formulation and solution of those problems where the objective function is linear and constraints are in the form of linear inequalities. 'Transportation and Assignment Problems' can be viewed as special cases of linear programming. These can be solved by the use of specially devised procedure as described in Unit 12 and Unit 13, respectively. An airline company can improve the fleet utilisation by using transportation model which will determine the deployment pattern of various aircrafts on different routes. Similarly, the assignment model can be used for assigning an optimal number of salesmen to various sales territories in terms of sales potential and number of customers in each territory.

Inventory models: The need for inventory arises out of differences in the timing or location of demand and supply. This applies whether you are thinking about raw meterials for a production process or finished goods stocked by a retailer, wholesaler or manufacturer. Inventory models are concerned with the determination of the optimal (called economic) order quantity and frequency considering such factors as demand per unit time, cost of placing orders, costs associated with goods held in inventory and cost of running short. For instance, the manager of a warehouse has to constantly balance the cost of carrying inventory, the loss of sales on account of shortages and administrative cost of ordering the stocks.

Waiting line (Queuing) models: Having to wait in line (or queue) is a common experience which all of us consider as unpleasant. Waiting to board a bus, to be served in a restaurant or in line at a bank, we are confronted with dead time. These models attempt to predict the operating characteristics (e.g., average length of waiting time of queue; average time spent in the queue by the customer, etc.) of queuing systems. The objective of these models is to minimise the sum of costs of providing service and costs of obtaining service, primarily in terms of time spent in queue. These models have been successfully applied in the context of general hospitals where by determining an optimal number of doctors, it is possible to reduce the average waiting time for patients.

Markovian models: These models are applicable in the situation where the status of the system, called its 'states', can be defined by some descriptive measure of numerical value and where the system moves from one state to another on a probability basis. In these models, we are given a sequence of events in which the probability of occurrence for an event depends upon the immediately preceding event. Markovian models have been successfully applied to analyse consumer buying patterns, to forecast bad debts, for planning personnel needs, to analyse equipment replacement, etc.

Network models: Whether you realise it or not, you already have experience at planning and managing projects. A project is any human undertaking with a clear beginning and a clear ending. These models enable us to cope with complexities and interdependencies involved in large projects. Programme Evaluation and Review Technique (PERT) and Critical Path Method (CPM) are such type of models which are used for planning, scheduling and controlling complex projects which can be characterised as networks of multiple activities.

Sequencing models: These models are concerned with the selection of an appropriate sequence of performing a series of jobs to be done on service facilities (machines) so as to optimise some efficiency measure of performance of the system. Thus, sequencing is concerned with such a problem in which efficiency measure depends upon the order or sequence of performing a series of jobs.

Replacement models: These models deal with the study of the technique of formulating the appropriate replacement policy in situations when some items or machinery need replacement for one reason or the other. Replacement problems are also to be used in respect of those equipments which fail completely and instantaneously.

Simulation models: A special class of mathematical models in management decision-making comprises what are termed simulation models. Simulation is an experimental method used to study behaviour over time. The process is similar to trial and error decision-making. Instead of implementing the decision and seeing the results in the real world, a model is constructed and used to test the decision. This is the way decisions can be evaluated without the risks inherent in actually implementing them.

The question then is, why use a simulation model, if it does not yield answers, while other mathematical approaches do. These models can be built for problems that are not amenable to other methods.

Decision theory: Decision theory assists the decision-maker in analysing complex problems with numerous alternatives and possible consequences. The basic objective of decision theory is to provide more concrete information concerning these consequences, so that the decision-maker may be able to identify the best course of action. Decision theory relies heavily not only upon the nature of the problem on hand, but also upon the decision environment. Basically there are four different states of decision environment as given below:

States of Decisions	Consequences
Certainty	Deterministic
Risk	Probabilistic
Uncertainty	Unknown
Conflict	Influenced by an opponent

Decision-making under certainty: The certainty state exists when all the information required to make a decision is known and available. For example, in a linear programming problem, we know the resources required to produce a product and its unit profit. In these types of decision problems, the decision-maker knows with certainty the consequences of every alternative course of action.

Decision-making under risk: The risk condition refers to the situation where the probabilities of certain outcomes (consequences) are known or can be estimated. There are many business situations which can be considered in this type of decision environment. Some of these are:

i) Should a particular product be introduced?

ii) Should we build a new plant or expand the present one?

iii) How many packets of bread should a bakery make for sale each day?

iv) Should a new advertising programme be initiated?

In each of the above decision situations we have the elements of a problem in decision-making under risk.

Decision-making under uncertainty: The uncertainty state refers to the condition where the probabilities of certain outcomes are not known. It is similar to decision-making under risk with one major difference, i.e., in this type of situation, we have no knowledge of the probabilities of future events and no idea at all of the likelihood of the various consequences. In many of our decisions under uncertainty, we can express our degree of optimism or convert the problem to risk with reasonable accuracy. Consider the following example: In a controlled economy, large business comes under a heavy influence of the thinking of the ruling party administered in the form of policy packages. Now whenever the ruling party changes, the top managements of large businesses face the situation of decision-making under uncertainty because no one can be sure of the changes a new government might administer.

Decision-making under conflict: A condition of conflict exists when the interests of two or more decision-makers are in a competitive situation. For example, if decision maker X benefits from a course of action he takes, it is only because decision-maker Y has also taken a certain course of action. Therefore, in this type of situation, the decision-makers are interested not only in what they do individually but also in what they do together. For example, when in a market which is largely controlled by two firms, one of them decides to reduce the prices of its products the other firm now faces the situation of decision-making under conflict. Because the decision of the first firm will affect the sales of the second firm and if the second firm also decides to reduce the price, both the firms will be once again affected.

Activity E

Each of the following is an example of a decision. Place each decision in its appropriate category: certainty, uncertainty, risk, conflict.

a) Deciding what to wear in the morning.
b) Deciding whether or not to raise the price of a product.
c) How many cakes should a bakery make for sale each day?
d) Should a costly new advertising programme be initiated?
e) Which mode of transportation to use for your product?
f) Settlement of labour management negotiations.

....
....
....
....
....

9.8 OPERATIONS RESEARCH AND MANAGERIAL DECISION-MAKING

Executives at all levels in business and industry come across with the problem of making decisions at every stage in their day-to-day activities. Operations Research provides the executive with quantitative basis for decision-making, and enhances his ability to make long-range plans and to solve everyday problems of running a business and industry with greater efficiency, competence and confidence.

The advantages of the study of Operations Research approach in decision-making may be classified as follows:

1 **Better control:** Sometimes it is a difficult and costly affair to provide continuous executive supervision to every routine work. An Operations Research approach may provide the executive an analytical and quantitative basis to identify the problem areas. This certainly enhances the effectiveness of the executive in achieving the specific objective.

2 **Better system:** Operations Research approach is also initiated to analyse a particular problem of decision-making such as best location for factories, warehouses as well as the economical means of transportation, production scheduling, replacement of an old machine, etc.

3 **Better decisions:** Since an incorrect decision can have serious consequences, it is important that the best information and decision-making tools are available to the executive. An Operations Research approach may provide the executive with techniques that will enable him to make decisions in a better way and with as much precision as possible. In other words, Operations Research approach gives the executive an improved insight into how he makes his decisions so that he can improve his decision-making process.

9.9 OPERATIONS RESEARCH IN MANAGEMENT

Some of the industrial business problems which can be analysed by Operations Research approach are classified as follows:

1 **Finance, Budgeting and Investment**

- Cashflow analysis, long-range capital requirements, dividend policies, investment portfolios
- Credit policies, credit risks and delinquent account procedures
- Claim and complaint procedures.

2 **Marketing**

- Product selection, timing competitive actions
- Number of salesmen, routing of salesmen, frequency of calling on accounts, percentage of time to be spent on prospects vs. current customers
- Advertising media with respect to cost and time
- Physical distribution of goods
- Warehousing decisions
- To predict and examine the behaviour of consumer.

3 **Purchasing, Procurement and Exploration**

- Rules for buying supplies under varying prices
- Determination of quantities and timing of purchases
- Bidding policies
- Strategies for exploration and exploitation of new material sources
- Replacement policies.

4 **Production Management**

- Assembly line balancing
- Aggregate production planning and smoothing
- Production scheduling and sequencing
- Stabilisation of production and employment, training, lay-offs and optimum product-mix
- Maintenance policies and preventive maintenance
- Maintenance of crew sizes
- Project scheduling and allocation of resources.

5 **Personnel Management**

- Determination of equitable salaries
- Mixes of the age and skills
- Recruiting policies and assignment of jobs
- Manpower planning, scheduling and deployment
- Management of absenteeism
- Labour management negotiations.

6 Research and Development

- Determination of areas of concentration of research and development
- Project selection
- Reliability and alternative design
- Determination of time-cost trade-off and control of development projects.

Activity F

List at least two applications of Operations Research in each functional area of management.

....

....

....

....

....

9.10 USE OF COMPUTERS

The use of computers has become closely associated with Operations Research. Many managerial problems are rather complex, involving numerous inter-related variables. The search for and evaluation of alternative solutions may become a gigantic computational project. Thus, many problems are solvable only with the aid of high speed computers. Computers have the advantage of being a relatively inexpensive means of rapid calculation and possess the accuracy and flexibility invaluable in experimenting with and solving managerial decision models.

The computer has provided a means for solving those problems which have long been quantifiable but computationally too complex or time-consuming for human calculation. Problems which would take months to solve manually can be solved in seconds using computers. It is, therefore, safe to say that the development of Operations Research goes hand in hand with the use of computers.

9.11 SUMMARY

The scientific method and Operations Research concepts are explained to help you improve your skill in solving management problems. This is done with a view to guide your thinking, to provide information, to help you solve problems, and in some cases to automate decision-making. The importance and complexity of the decision-making process has resulted in the wide application of Operations Research into diversified field of business and management. This unit ends with a brief description on the use of computers.

9.12 KEY WORDS

Analog model: A model in which one physical property is used to represent another physical property.

Decision theory models: A class of Operations Research models designed to select an optimal course of action from a set of alternative courses of action.

Decision variables: The unknown that are to be determined by solving the model.

Deterministic models: A model in which the functional relationships and parameters are known with certainty.

Iconic model: A scaled physical representation of a real system.

Operations Research: Operations Research is a scientific method of providing executive departments with a quantitative basis for decisions regarding the operations under control.

9.13 FURTHER READINGS

Gupta, M.P. and J.K. Sharma, 1987. *Operations Research for Management,* National Publishing House: New Delhi.

Herbert, M. and G.P. Wright, 1979. *Operations Research Techniques for Management,* Prentice Hall: Englewood-Cliffs.

Hsiao, J.C. and D.S. Cleaver, 1982. *Management Science,* Houghton Mifflin Company: Boston.

Loomba, N.P. 1978. *Management-A Quantitative Perspective,* Macmillan Publishing Company: New York.

Markland, R.E. 1983. *Topics in Management Science,* John Wiley & Sons: New York.

UNIT 10 LINEAR PROGRAMMING-PROBLEM FORMULATION AND GRAPHICAL METHOD

Objectives

After studying this unit, you should be able to appreciate:

- the nature of linear programming (LP) problems
- the requirements of LP problems, and specifically the concepts of 'linear', 'programming' and 'optimal'
- advantages and applications of LP problems
- limitations of LP problems
- description of the general structure of a LP problem and the concepts of 'objective function', 'constraints', and 'non-negativity'
- the use of graphical method for solving LP problems.

Structure

10.1 INTRODUCTION

The resource allocation problem affects all of us. For instance, each day you must determine how to allocate your time among activities such as: study, work and recreation. Similarly, most of us work with a limited financial budget and must make difficult decisions as to how these funds can best be used for family living. If you are a manager, then the problem may arise in allocating limited resources (such as men, machines, capital equipment, materials, etc.) among competing activities. Linear Programming (LP) has been successfully applied to solve such type of problems. In this unit we will undertake the study of Linear Programming.

Linear Programming is one of the most widely used and best understood Operations Research techniques. The LP is concerned with the problem of allocating limited resources among competing activities in an optimal manner. This type of problem arises in a number of situations such as manufacturing an item at a minimum cost, blending of chemicals, allocating sales men to sales territories, selection of various media for advertising campaign, scheduling production, etc. In each of these situations, you may observe that the common ingredient is the requirement that some type of scarce/limited resource must be allocated to some specific activity. Since the resources employed generally involve costs and produce profits, LP problems generally concern either minimising the cost or maximising the profit.

10.2 TERMINOLOGY AND REQUIREMENTS OF LINEAR PROGRAMMING

Terminology

The word 'linear' is used to describe the relationship among two or more variables which are directly proportional. For example, if doubling (or tripling) the production of a product will exactly double (or triple) the profit and required resources, then it is a linear relationship. The word 'programming' means planning of activities in a manner that achieves some 'optimal' result with restricted resources.

A programme is 'optimal' if it maximises or minimises some measure or criterion of effectiveness, such as profit, cost or sales.

Requirements

Regardless of the way one defines Linear Programming, certain basic requirements which are given below are necessary before it can be used for optimisation problems.

Decision variable and their relationship: The decision (activity) variables refer to candidates (products, services, projects etc.) that are competing with one another for sharing the given limited resources. These variables are usually inter-related in terms of utilisation of resources and need simultaneous solutions. The relationship among these variables should be linear.

Objective function: The Linear Programming problem must have a well-defined (explicit) objective function to optimise. For example, maximisation of profits, minimisation of costs or elapsed time of the system being studied. It should be expressed as linear function of decision variables.

Constraints: There must be limitations on resources which are to be allocated among various competing activities. These resources may be production capacity, manpower, time, space or machinery. These must be capable of being expressed as linear equalities or inequalities in terms of decision variables.

Alternative courses of action: There must be alternative courses of action. For example, there may be many processes open to a firm for producing a commodity and one process can be substituted for another.

Non-negativity restriction: All variables must assume non-negative values, that is, all variables must take on values equal to or greater than zero. Therefore, the problem should not result in negative values for the variables.

Linearity and divisibility: All relationships (objective function and constraints) must exhibit linearity, that is, relationships among decision variables must be directly proportional. For example, if our resource increases by some percentage, then it should increase the outcome by the same percentage. Divisibility means that the variables are not limited to integers. It is assumed that decision variables are continuous, i.e., fractional values of these variables must be permissible in obtaining an optimal solution.

Deterministic: In LP model (objective function and constraints), it is assumed that all the model coefficients are completely known (deterministic), e.g., profit per unit of each product, and amount of resource available are assumed to be fixed during the planning period.

Activity A

What are the characteristics of an LP problem?

....

....

....

....

....

10.3 ADVANTAGES OF LINEAR PROGRAMMING

1 Linear Programming can be used to solve allocation type problems. Such problems are very common and extremely important in organisations. Their solution is difficult due to the fact that an infinite number of possible solutions may exist. An LP not only provides the optimal solution but does so in a very efficient manner. Further, it provides additional information concerning the value of the resources to be allocated.

2 Linear Programming gives possible and practical solutions as there might be other constrainsts operating outside the problem which must be taken into account. Just because we can produce so many units does not mean that they can be sold. It allows modification of its mathematical solution for the sake of convenience to the decisionmaker.

3 Linear Programming improves the quality of decisions. It makes decisions more objective rather than subjective.

4 Linear Programming helps in highlighting the bottlenecks in the production processes.

5 Linear Programming helps in attaining optimum use of productive factors. It also indicates the significance and utility of these factors more effectively.

10.4 APPLICATION AREAS OF LINEAR PROGRAMMING

Linear Programming is the most widely used technique of decision-making in business, industry and in various other fields. In this section, we will discuss a few of the broad application areas of LP.

1 Production management: Linear Programming can be applied for determining the optimal product-mix to make best use of machine and manhours available. It is also used for product smoothing and assembly line-balancing.

2 Marketing management: Linear Programming helps in analysing the audience coverage of an advertising campaign based on the available advertising media and budget. It also helps travelling salesman in finding the shortest route for his tour. It is also used to determine the optimal distribution schedule for transporting the product from several warehouses to various market locations in such a way that the total transportation cost is minimum.

3 Personnel management: Linear Programming enables the Personnel Manager to analyse problems relating to recruitment, selection, training, development of manpower. It is also used to determine the minimum number of employees needed to work in various shifts to meet a time varying workload.

4 Financial management: Linear Programming helps in selection of specific investments from among a wide variety of alternatives. Using LP model, decision can be made with regard to how much production is to be supported by internally and externally generated funds.

5 Agriculture: Agricultural applications fall into two categories, farm economics and farm management. The former deals with agricultural economy of a nation or region, while the latter is concerned with the problems of the individual farm. The study of farm economics deals with inter-regional competition and optimum spatial allocation of crop production. Efficient production patterns can be specified by LP model under regional land resources and national demands constraints. An LP can be applied in agricultural planning, e.g., allocation of limited resources such as acreage, labour, water supply and working capital, etc., in such a way as to maximise net revenue.

6 Military: Military applications include the problem of selecting an air weapons system against the enemy so as to keep them pinned down and at the same time minimise the amount of aviation gasoline used. A variation of transportation issue is the problem of community defence against disaster, the solution of which yields the number of defence units that should be used in a given attack in order to provide the required level of protection at the lowest possible cost.

Other applications of LP include the areas of administration, education, inventory control, fleet utilisation, hospitals, awarding contracts, capital budgeting, etc.

Activity B

List any five applications of LP in the context of your job.

....

....

....

....

....

....

10.5 LIMITATIONS OF LINEAR PROGRAMMING

In spite of having wide field of applications, there are some limitations associated with Linear Programming which are given below:

1 The Linear Programming problem assumes the linearity of objective function and constraints. But, generally, the objective function and the constraints in real-life situations concerning business and industrial problems are not linearly related to the variables. Such problems cannot be solved by LP technique.

2 There is no guarantee that the solution by LP technique will give integer valued solution. For example, in finding out how many men and machines would be required to perform a particular job, the solution may be in fractions, and rounding off the solution to the nearest integer may not yield an optimal solution. In such cases, LP cannot be used.

3 Linear Programming model does not take into consideration the effect of time and uncertainty.

4 Parameters appearing in the model are assumed to be constant. But in real-life situation, they are frequently neither known nor are they constants.

5 Linear Programming deals with only a single objective, whereas in real-life situations we may come across more than one objective. When above situations (limitations) arise, other techniques such as integer programming can be utilised.

Activity C

Differentiate between requirements and limitations of LP model.

....

....

....

....

....

....

....

10.6 FORMULATION OF LINEAR PROGRAMMING PROBLEMS

Here the formulation means writing or expressing a problem in a convenient mathematical form. Let us consider the following examples:

Example 1

A furniture firm manufactures tables and chairs. Data given below shows the resources consumed and unit profit in manufacturing a table and a chair. Here it is assumed that wood and labour are the only two resources which are consumed in manufacturing furniture. The manager of the firm wishes to determine how many tables and chairs should be made to maximise the total profit. Let us formulate the problem as a linear programming problem.

Resource	Unit requirements		Amount available
	Table	Chair	
Wood (sq. ft.)	30	20	300
Labour (hours)	5	10	110
Unit profit (Rs.)	6	8	

Fomulation: We will begin by treating the number of tables and chairs to be manufactured as unknown quantities or decision variables. Let X_1 and X_2 be the number of tables and chairs to be manufactured, respectively.

Objecitive function: Since total profit consists of the profit derived from selling tables at Rs. 6 each plus the profit derived from selling chairs at Rs. 8 each. Thus, $6X_1$ is the profit earned by selling tables and $8X_2$ is the corresponding profit by selling chairs. As the manager wants to achieve the greatest possible profit (say Z) it can be stated algebraically by writing profit equation as:

$$\text{Maximise } Z = 6X_1 + 8X_2$$

In this form, the profit expression provides the objective function.

Constraints: Constraints are limitations or restrictions placed on availability of resources:

i) Wood used for tables + wood used for chairs ≤ Available Wood (sq.ft.)

$$\text{i.e., } 30X_1 + 20X_2 \leq 300$$

the sign ≤ is read "less than or equal to".

ii) Labour hours for table + hours for chair ≤ Avialable Labour (hours)

$$\text{i.e., } 5X_1 + 10X_2 \leq 110$$

Further we cannot have negative production, i.e., either we manufacture or do not manufacture.

$$\text{i.e., } X_1 \geq 0 \text{ and } X_2 \geq 0$$

These are called non-negativity restrictions.

We can now state the problem also referred to as a Linear Programming problem in full.

$$\text{Max } Z = 6X_1 + 8X_2$$

subject to $30X_1 + 20X_2 \leq 300$

and, $5X_1 + 10X_2 \leq 110$

$$X_1, X_2 \geq 0$$

Activity D

Given the data below for products A and B:

Resource	Product		Resource available
	A	B	
R_1	60	20	1200
R_2	40	50	2000
Contribution Unit	Rs. 3	Rs.2	

Fomulate the given problem as a linear programming problem.

Example 2

The vitamins A and B are found in two different foods F_1 and F_2. One unit of food F_1 contains two units of vitamins A and five units of vitamin B. One unit of F_2 contains four units of vitamin A and two units of vitamin B. One unit of food F_1 and F_2 cost Rs. 3 and Rs. 2.50, respectively. The minimum daily requirement for a person of vitamin A and B is 40 and 50 units, respectively. Find the optimal mix of food F_1 and F_2 at the minimum cost which meets the daily minimum requirement of vitamin A and B. Assume that anything in excess of daily minimum requirement of vitamin A and B is not harmful. Let us, formulate the given problem as a linear programming problem.

Formulation: The data of the given problem can conveniently be summarised in the following tabular form:

Vitamin	Food F_1	Food F_2	Daily requirement
A	2	4	40
B	5	2	50
Cost (Rs.)/Unit	3	2.5	

Let X_1 and X_2 be the number of units needed of food F_1 and F_2, respectively to meet the daily requirement of vitamin A and B.

Since the daily minimum requirement of vitamins A and B is 40 and 50 units, respectively, the constraints are given by

$$2X_1 + 4X_2 \geq 40$$
$$5X_1 + 2X_2 \geq 50$$

As cost per unit of food F_1 and F_2 are Rs. 3 and Rs. 2.50, respectively, then the objective function will become

$$\text{Min} \quad Z = 3X_1 + 2.5X_2$$

Also either we need some units of food F_1 and F_2 or at worst we may not need any unit of the food. Therefore, the non-negativity restriction is given as:

$$X_1 \geq 0 \text{ and } X_2 \geq 0$$

The given problem then can be expressed in LP model as:

$$\text{Min } Z = 3X_1 + 2.5X_2$$

subject to $\quad 2X_1 + 4X_2 \geq 40$

$\quad 5X_1 + 2X_2 \geq 50$

and, $\quad X_1, X_2 \geq 0$

Activity E

Suppose you are trying to select the cheapest combination of two foods X and Y to meet daily vitamin needs. The vitamin needs call for at least 40 units of vitamin A, 50 units of vitamin B, and 49 units of vitamin C. Each gram of food X provides 4 units of vitamin A, 10 units of vitamin B, and 7 units of vitamin C. Each gram of food Y provides 10 units of vitamin A, 5 units of vitamin B, and 7 units of vitamin C. Food X costs Rs. 5 per gram, and food Y costs Rs. 8 per gram. The objective is to find the least expensive way to satisfy your vitamin needs. Formulate this problem as a linear programming problem.

....

....

....

....

....

Example 3

A company is making two products A and B. The cost of producing one unit of product A and B is Rs. 60 and Rs. 80, respectively. As per the agreement, the company has to supply at least 200 units of product B to its regular customers. One unit of product A requires one machine hour whereas product B has machine hours available abundantly within the company. Total machine hours available for product A are 400 hours. One unit of each product A and B requires one labour hour each and total of 500 labour hours are available.

The company wants to minimise the cost of production by satisfying the given requirements. Let us formulate this problem as a linear programming problem.

Formulation: Let X_1 and X_2 be the number of units of product A and B to be produced, respectively, then LP model is given by:

Min $Z = 60X_1 + 80X_2$

subject to

$X_2 \geq 200$ (Agreement constraint)

$X_1 \leq 400$ (machine hours constraint for product A)

$X_1 + X_2 \leq 500$ (Labour hours constraint)

and $X_1, X_2 \geq 0$

10.7 MATHEMATICAL FORMULATION OF LINEAR PROGRAMMING PROBLEMS

Instead of two decision variables and few constraints, if we have n decision variables and m constraints in the problem, then we would have the following type of mathematical formulation of LP problem.

Optimise (maximise or minimise)

$$Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$$

subject to

$$a_{11} X_1 + a_{12} X_2 + \ldots + a_{1n} X_n \lesseqgtr b_1$$
$$a_{21} X_1 + a_{22} X_2 + \ldots + a_{2n} X_n \lesseqgtr b_2$$
$$\vdots$$
$$a_{m1} X_1 + a_{m2} X_2 + \ldots + a_{mn} X_n \lesseqgtr b_m$$

and $X_1, X_2, \ldots, X_n \geq 0$

The above formulation may be put in the following compact form by using the (Σ) summation notation.

Optimise (max. or min.) $Z = \sum_{j=1}^{n} C_j X_j$ (objective function)

subject to $\sum_{j=1}^{n} a_{ij} X_j \lesseqgtr b_i$; $i = 1,2, \ldots, m$ (constraints)

and $X_j \geq 0$; $j = 1,2 \ldots, n$ (non-negativity restrictions)

where,

X_j = decision variables, i.e., quantity of the jth variable of interest to the decision-maker.

C_j = constant representing per unit contribution to the objective function (profit or cost) of the jth decision variable.

a_{ij} = constant, representing exchange coefficients of the jth decision variable in the ith constraint.

b_i = constant, representing ith constraint requirement or availability.

and the expression ($\lesseqgtr$) means that each constraint may take only one of the three possible forms:

i) less than or equal to ($\leq$)

ii) equal to (=)

iii) greater than or equal to ($\geq$)

The expression $X_j \geq 0$ simply means that the $X_{j\text{s}}$ must be non-negative.

The basic difference between a maximisation and minimisation problem in linear programming is found in the signs of the inequalities of constraints. The constraints are generally expressed by 'less than or equal to' ($\leq$) sign in the maximisation problem, whereas those of the minimisation problem are expressed by 'greater than or equal to' ($\geq$) sign.

In particular any minimisation problem can be converted into an equivalent maximisation problem by changing the sign of $C_{j's}$ in the objective function. For example, the linear objective function

Minimise $$Z = \sum_{j=1}^{n} C_j X_j$$

is equivalent to Maximise $Z^* = \sum_{j=1}^{n} (-C_j) X_j$ whereas, $Z^* = -Z$

Thus, for example, maximising the objective function

$$Z = 3X_1 + 5X_2$$

is equivalent to minimising its negative counterpart, i.e.,

$$Z^* = -Z = -3X1 - 5X2$$

Equality Sign

So far, we have discussed only those constains which are either expressed by ($\leq$) sign or by ($\geq$) sign. If there exists a situation, when some of the constraints are expressed by equations, each of the equations may be replaced by two inequalities. For example,

$$a_{11} x_1 + a_{12} x_2 = b_1$$

is equivalent to the two simultaneous constraints

$$a_{11} x_1 + a_{12} x_2 \leq b_1$$

and $$a_{11} x_1 + a_{12} x_2 \geq b_1$$

or $$-a_{11} x_1 - a_{12} x_2 \leq -b_1$$

Unrestricted Variables

In certain problems, there may exist situations, when some decision variables are unrestricted in sign (positive, negative or zero). In all such cases, the decision variables can always be expressed as the difference between two non-negative variables. For example, if X_j is unrestricted in sign, then we define two non-negative variables X_j and X_i^- such that

$$X_i = X_i^+ - X_i^-$$
$$X_i^+,\ X_i^- \geq 0$$

10.8 SOLUTION TO LP PROBLEM

Several methods or algorithms have been developed to get the optimal solution to LP problem. In this section, we shall be discussing only graphical method. Simplex method, a more general method for solving LP problems, will be discussed in the following unit.

Graphical Method

Graphical method is used for solving those LP problems which involve only two variables. Consider example 1 for illustrating the graphical method.

On our graph, let the horizontal axis represent the number of tables manufactured and the vertical axis the number of chairs manufactured. We will plot a line for each of the two constraints and the two non-negativity restrictions. We shall begin by plotting the lines corresponding to the non-negativity restrictions. This plotting suggests that the graphical method always starts with the first quadrant of the graph.

Now in order to plot the constraints on the graph, temporarily we will consider inequalities as equations, i.e.,

$$30X_1 + 20X_2 = 300$$
$$5X_1 + 10X_2 = 110$$

When plotted on the graph, these will represent straight lines. All points on these straight lines represent combinations of tables and chairs quantities that consume the exact amount of resource that is available. In other words, points on these lines are satisfying the equality and any point below the line will include the inequality part of the constraint.

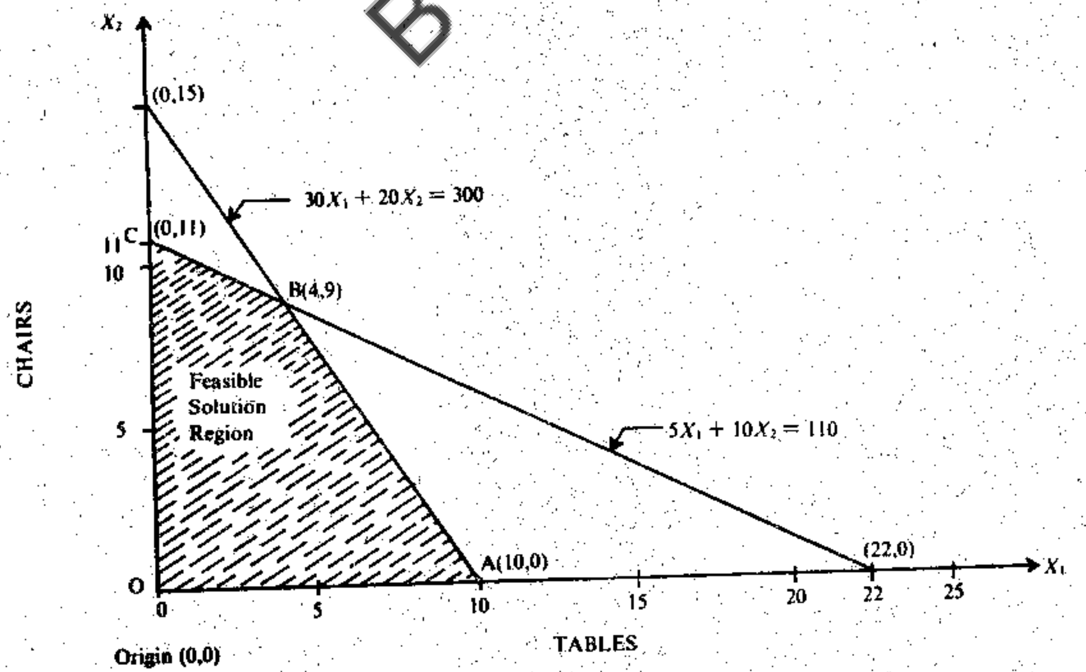
A straight line is completely specified by knowing any two points that fall on that line. Therefore, to plot any constraint, we need only to specify two points on that line and then draw the line by connecting these two points. To illustrate let $X_1 = 0$ in the availability wood constraint, then we get:

$$30\,(X_1 = 0) + 20X_2 = 300$$

or

$$X_2 = 15$$

Figure I: Graphical Solution for Maximisation Problem



This represents the vertical interceptor on X_2 axis and is written as (0,15), i.e., 0 tables and 15 chairs. Similarly, horizontal interceptor on X_1 axis is obtained by putting $X_2 = 0$, we get,

$$30 X_1 + 20(0) = 300$$

or

$$X_1 = 10$$

It is also written as (10,0) and represents 10 tables and 0 chairs. Join these two interceptors to draw the constraint line. Thus, the points on or below the line represent the region satisfied by this constraint. Any point which does not lie on or below the line gives infeasible solution.

Similarly, the availability (labour) constraint can also be plotted on the graph as shown in the Figure.

Any combination of values of X_1 and X_2 which satisfies the given constraints is called a "feasible solution". The area OABC in the figure satisfied by the constraints is shown by shaded area and is also called feasible solution region or space. The coordinates of the points on the corners of the region OABC can be obtained by solving equations intersecting on these points.

The shaded region may contain an infinite number of points which would satisfy the constraints of the given LP problem. However, it can be proved that the optimal solution of any LP problem corresponds to one of the corner points (also called extreme points) of the feasible solution region. Thus we confine only to those points which correspond to corners of feasible solution region to get an optimal value of the objective function of given LP problem.

In the present example, value of objective function on the corner points can be evaluated as follows:

Coordinates of corner point		Objective function $Z = 6X_1 + 8X_2$	Value
O	(0,0)	6(0) + 8(0)	0
A	(10,0)	6(10) + 8(0)	60
B	(4,9)	6(4) + 8(9)	96
C	(0,11)	6(0) + 8(11)	88

The maximum profit of Rs. 96 is obtained at the corner point B(4,9), i.e. $X_1 = 4$ and $X_2 = 9$ which also satisfy the given constraints. Hence to maximise profit, the company must produce 4 tables and 9 chairs.

The steps of graphical method can be summarised as follows:

1 Formulate the linear programming problem.

2 Plot the constraint lines considering them as equations.

3 Identify the feasible solution region.

4 Locate the corner points of the feasible region.

5 Calculate the value of the objective function on the corner points.

6 Choose the point where the objective function has optimal value.

Activity F

Use graphical method to obtain the solutio.. for the exercise (given earlier in this unit) on formulation for the maximisation case.

....

....

....

10.9 USING THE GRAPHICAL METHOD FOR MINIMISATION

In this section we will consider an LP problem in which the objective is to minimise costs. The solution of minimisation problem follows the same procedure as that of maximisation problem. The basic difference is that we now want the smallest possible value for the objective function. To illustrate, let us consider Example 2 discussed earlier and is reproduced below:

$$\text{Min } Z = 3X_1 + 2.25X_2$$

subject to

$$2X_1 + 4X_2 \geq 40$$

$$5X_1 + 2X_2 \geq 50$$

and

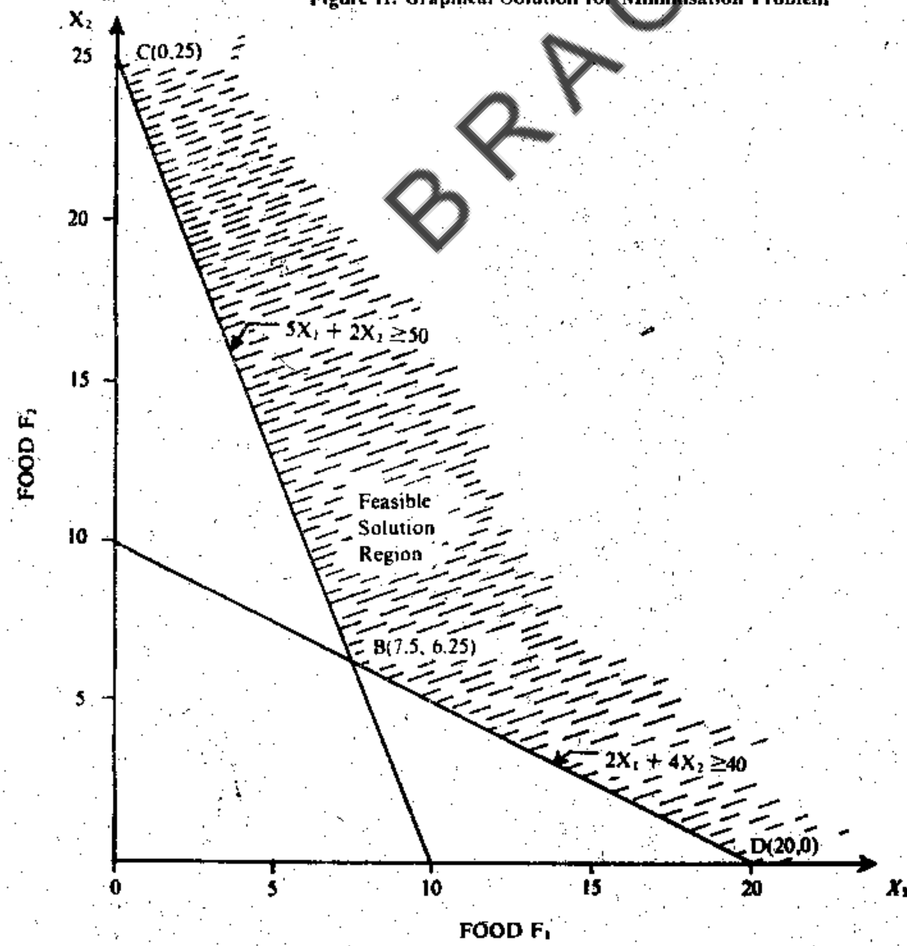
$$X_1, X_2 \geq 0$$

Because there are only two variables X_1 and X_2 we can construct a graph of the set of feasible solutions. Recall that the non-negativity restrictions reduce this feasible region to the first quadrant of the graph. To this quadrant we first plot the constraint for vitamin A as a straight line given by

$$2X_1 + 4X_2 = 40$$

We now obtain the two points necessary to plot this line. We first set $X_1 = 0$, which yields $X_2 = 10$ and set $X_2 = 0$ getting $X_1 = 20$. We then plot this line as shown in the graph. Similarly, we can plot the constraint for vitamin B. Since the inequalities are of the greater than or equal to type, the feasible region is formed by considering the area to the upper right side of each equation (away from the origin).

Figure II: Graphical Solution for Minimisation Problem



The area above CBD is satisfied by the two constraints and is shown by shaded area which is termed as feasible solution region. There is a significant difference between the feasible region for this minimisation problem and the one for the maximisation problem. The feasible region for this minimisation problem is unbounded and unlimited because any combination of vitamins A and B will satisfy the constraints.

Since the optimal solution corresponds to one of the corner (extreme) points, we will calculate the values of the objective function for each corner point, viz., D (20,0); B(7.5, 6.25); and C(0.25). The calculations are shown below in the table.

Coordinates of corner point	Objective function $Z = 3X_1 + 2.25X_2$	Value of Z
D (20,0)	3 (20) + 2.25 (0)	60
B (7.5, 6.25)	3 (7.5) + 2.25 (6.25)	36.5625
C (0,25)	3 (0) + 2.25 (25)	56.25

The minimum cost is obtained at the corner point B (7.5, 6.25), i.e., $X_1 = 7.5$ and $X_2 = 6.25$. Hence to minimise cost, and to meet the daily vitamins requirement, one should have 7.5 units of food F_1 and 6.25 units of food F_2.

Although the two examples so far discussed have contained constraints of one type either all 'less than or equal to' or all 'greater than or equal to', it is possible to have mixed constraints.

Activity G

Use graphical method to obtain the solution for the exercise (given earlier in this unit) on formulation for the minimisation case.

....

....

....

....

Activity H

Use graphical method to solve example 3 given earlier in this unit.

....

....

....

....

10.10 SPECIAL CASES IN LINEAR PROGRAMMING

Each of the LP problems discussed in this unit had a unique optimal solution. However, it is possible for LP problems to have some special cases. Some of these special cases include:

1 Alternative optimal solution
2 Unbounded solution
3 Infeasible solution
4 Redundant constraint

Each of these special cases will now be described briefly.

Alternative Optimal Solution
All our examples so far have resulted in unique optimal solution. This may not always be true. Any particular LP problem may have multiple solutions, all of which are optimal.

Unbounded Solution
When a LP solution is permitted to be infinitely large, it is known to be unbounded. For example, in a maximisation problem at least one of the constraints should be an equality or less than or equal to ($\leq$) type. If all of the constraints are greater than or equal to ($\geq$) type, then there will be no upper limit on the feasible region. Similarly, for minimisation, there must be an equality or a greater than or equal to ($\geq$) type constraint if a solution is to be found.

Infeasible Solution
If there is no solution that satisfies all the constraints, it is said to be infeasible. Such a condition generally indicates that the LP problem has been wrongly formulated. The only way to arrive at a feasible solution is to reformulate one or more of the constraints.

Redundant Constraint
In a properly formulated LP problem, each of the constraints will define a portion of the boundary of the feasible solution region. Whenever, a constraint does not define a portion of the boundary of the feasible solution region, it is called a redundant constraint. A redundant constraint is unnecessary for the problem and may be omitted from the problem formulation.

10.11 LIMITATIONS OF THE GRAPHICAL METHOD

The graphical method is widely used as an academic exercise because it enables the student to visualise the LP solution process. The method is severely limited in application by the fact that the number of variables should be restricted to two dimensions. For three variables problem, it becomes very difficult as one must be familiar with three dimensional space. Of course, the method breaks down completely for four or more variables.

Other methods of solution which avoid the limitations of the graphical method have been developed. The most widely used method is known as Simplex Method which is described in the next unit.

Use of Computer
Real-life LP problems are almost always too large and complex for graphical or manual simplex methods. Therefore, they are solved with the help of computers. Numerous LP packages exist for solving linear programming problems, ranging from simplified procedures designed for classroom use to sophisticated and complex programmes. Virtually all computer manufacturers and software consultants offer these LP packages.

10.12 SUMMARY

Linear Programming is a powerful technique of Operations Research designed to solve allocation problems. It helps in choosing the best combinations of various activities to meet various criteria in such a way that the solution obtained is optimal. LP may be applicable if you have a problem which (a) is deterministic in nature, (b) involves multiple variables which are inter-related, (c) has a set of criteria which must be met, and (d) has a single objective. All LP problems can be formulated in a common format. The graphical method provides a good conceptual introduction to linear programming problems. However, its usefulness in solving practical problems is quite limited. For real-life problems, simplex method (discussed in next unit) with the help of computer packages can be used.

10.13 KEY WORDS

Constraints: An upper limit on the availability of a resource or a lower limit on necessary levels to achieve.

Corner (or Extra) points: The points of feasible solution region formed by the constraints and the non-negativity restrictions of a LP problem.

Feasible solution region: The region defined by the constraints and the non-negativity restrictions of the LP problem.

Linear programming: A linear deterministic model used to solve the problem of allocating limited resources among competing activities.

Non-negativity restrictions: The conditions that require the values of the decision variables to be either zero or more than zero.

Objective function: An equation that specifies the dependent relationship between the decision objective and the decision variables.

10.14 FURTHER READINGS

Dannenbring, D.G. and M.K. Starr, 1981. *Management Science,* McGraw-Hill International Book Company: New York.

Gupta, M.P. and J.K. Sharma, 1983. *Linear Programming for Management,* National Publishing House: New Delhi.

Loomba, N.P. 1978. *Manangement-A Quantitative Perspective,* MacMillan Publishing Company: New York.

Turban, E. and J.R. Meredith, 1985. *Fundamentals of Management Science,* Business Publications Inc: New York.

UNIT 11 LINEAR PROGRAMMING - THE SIMPLEX METHOD

Objectives

After studying this unit, you should be able to:

- describe the general logic of the simplex method
- prepare a LP problem prior to use of the simplex method
- build a simplex table and describe its components
- solve problems by the simplex method
- describe special situations in LP problem such as degeneracy, unbounded solution and multiple solutions.

Structure

11.1 INTRODUCTION

In the previous unit, we used graphical method for solving linear programming (LP) problems when only two variables were considered. We can solve LP problems with two or more than two variables by using a systematic procedure called the **Simplex Method.** The simplex method is the name given to the **solution algorithm** for solving LP problems developed by George B. Dantzig in 1947. By solution algorithm we refer to an iterative procedure having fixed computational rules that leads to a solution to the problem in a finite number of steps. By using the simplex method, one is capable of solving large LP problems efficiently. The simplex method allows us to evaluate the corner points in such a way that each successive corner point gives the same or better solution than the previous one. This search for examining successive corner points will continue until no better solution can be found.

The objective of this unit is to describe the simplex method. After you have gone through with this unit, you will have the necessary understanding to solve at least simple LP problems.

11.2 THE SIMPLEX METHOD (MAXIMISATION CASE)

It was pointed out during the graphical solution procedure that the search for an optimal solution can be limited to only the corner points of the feasible solution region. This was very easy to do manually for a problem with two variables and less number of constraints, but for larger problems a more efficient procedure is necessary for identifying and evaluating corner points. This is one of the main objectives of the simplex method. The simplex method is a well defined systematic procedure called an **algorithm.** The simplex algorithm provides a more structured method for moving from one corner point to another, always maintaining or improving the objective function, until an optimal solution is obtained. To explain and develop the simplex method, let us consider Example 1 as given in the previous unit and its formulation is reproduced below:

$$\text{Max} \quad Z = 6X_1 + 8X_2$$

subject to

$$30X_1 + 20X_2 \leq 300$$
$$5X_1 + 10X_2 \leq 110$$

and

$$X_1, X_2 \geq 0$$

The problem must first be stated in equation form so that the simplex method can be used. Thus, every inequality constraint in the LP problem must first be converted into an equality constraint so that the problem can be written in a standard form. We do this by adding a **slack variable** to each constraint. Each slack variable corresponds to the amount of unused capacity (or resources) for the constraint to which it is added. We will use the symbol 'S' to represent slack variables. Slack variables are always added to the 'less than' type of constraint to make it an equality. These slack variables must be non-negative; otherwise, the capacity utilised will exceed the total available capacity. The constraints for the given problem can now be rewritten with slack variables to form the equalities as given below:

$$30X_1 + 20X_2 + S_1 = 300$$
$$5X_1 + 10X_2 + S_2 = 110$$

Since slack variables represent unused resources, their contribution in the objective function is zero. Including these slack variables in the objective function, we get

$$\text{Max} \quad Z = 6X_1 + 8X_2 + 0S_1 + 0S_2$$

The logic of the simplex method is based on the fact that only the corner points of the feasible solution region can give unique optimal solution. The search starts with a solution at the origin indicating nothing can be produced and therefore the values of decision variables are zero, i.e., $X_1 = 0$ and $X_2 = 0$. When we are not producing anything, obviously we are left with unused capacity of $S_1 = 300$ and $S_2 = 110$. We note that the current solution has two variables (slack variables S_1 and S_2) with non-zero solution values and two variables (decision variables X_1 and X_2) with zero values. Variables with non-zero solution values are called basic variables. Variables with zero values are called non-basic variables. We can observe that there will always be the same number of basic variables as there are the number of constraints, provided the constraints are not redundant and that a basic feasible solution exists. Solutions with basic variables are called basic solutions. The basic solutions can further be divided into two categories: feasible and infeasible. The basic feasible solutions are those that satisfy all the constraints. The simplex procedure searches for basic feasible solutions only at the corner points of the feasible solution region. Introducing the slack variables, our problem can be stated as given below:

$$\text{Max } Z = 6X_1 + 8X_2 + 0S_1 + 0S_2$$

subject to

$$30X_1 + 20X_2 + S_1 = 300$$

$$5X_1 + 10X_2 + S_2 = 110$$

and

$$X_1, X_2, S_1, S_2 \geq 0$$

To use simplex method, it is convenient to put the problem in a tabular form as shown below:

Table I

$C_j \rightarrow$	CONTRIBUTION PER UNIT		6	8	0	0	MINIMUM RATIO
↓	BASIC VARIABLES	SOLUTION VALUES	X_1	X_2	S_1	S_2	
0	S_1	300	30	20	1	0	300/20 = 15
0	S_2	110	5	[10]	0	1	110/10 = 11 ← KEY ROW
CONTRIBUTION LOSS PER UNIT Z_j			0	0	0	0	$Z = 0$
NET CONTRIBUTION PER UNIT $C_j - Z_j$			6	8	0	0	

KEY ELEMENT (10) ↑ KEY COLUMN (X_2)

The first row of the table indicates the value of C_j, the coefficients of the objective function, and indicates the contribution per unit to the objective function of each of the variables. The second row of the table provides the column headings for the table. The first column heading lists the coefficient of the objective function of the current basic variables. The second column represents the basic variables in the current solution. The next column heading 'solution values' is the current solution. In our example, when the solution is at the origin, the basic variables are the slack variables S_1 and S_2, and these are listed in the second column of Table 1. Referring back to the first column, the coefficients for these two basic variables in the objective functions are $S_1 = 0$ and $S_2 = 0$, respectively. If the current solution at the origin is X_1=0 and X_2=0, then the solution values correspond to $S_1 = 300$ and $S_2 = 110$ as shown in the third column of the Table. The next four columns headed by X_1, X_2, S_1, and S_2 are the coefficients of the constraint set.

The Z_j row represents the decrease in the value of the objective function that will result if one unit of the jth variable is brought into the solution. Therefore, the Z_j values can be thought of as the objective function contribution loss per unit and is determined by summing the products of the coefficients of C_j column and the physical rates of substitution (coefficients in the constraint set) associated with corresponding basic variables. In our example, contribution loss per unit (Z_j) row values are determined as given below:

Contribution loss per unit (Z_j) =Summation of (Coefficients of C_j column X corresponding Coefficients of the constraint set)

For X_1 Column: $Z_j = 0 \times 30 + 0 \times 5 = 0$

Similarly we can calculate Z_j values for other columns as shown in Table 1.

The last $(C_j - Z_j)$ row represents the net contribution per unit and its value is determined by subtracting the appropriate Z_j value from the corresponding coefficient C_j value in the objective function for that column. This value of $(C_j - Z_j)$ is the difference between the contribution C_j and the loss Z_j that result from one unit of X_j being produced. For determining the value of the last row $(C_j - Z_j)$, the calculations are shown below:

Net contribution per unit $(C_j - Z_j)$ = contribution per unit - contribution loss per unit

In our example, the values of net contribution per unit $(C_j - Z_j)$ for X_1 column is determined as given below:

Net contribution per unit $(C_j - Z_j) = 6 - 0 = 6$ and similarly we can calculate the other values of $(C_j - Z_j)$ for other columns as shown in Table 1.

The positive values in the $(C_j - Z_j)$ row indicate improvement possibility in the existing solution. The objective is to maximise profit, therefore, consider that column, where per unit contribution is the largest. In our case, we notice that in X_2 column, contribution per unit (i.e., Rs. 8) is maximum. This criterion helps us to know the variable to be entered into the solution basis. Thus X_2 is the entering variable. The column corrsponding to entering variable is termed as 'key column'. Since we have decided to enter one variable as the basic variable into the solution basis, therefore, we must replace one of the existing basic variable to depart from the solution basis. The variable to depart is identified by forming the ratios of solution values to physical rates of substitution of entering variable. Thus, in our example, we have

For S_1; ratio = 300/20 = 15

For S_2; ratio = 110/10 = 11

The one with the minimum ratio (S_2) represents the variable to depart from the solution basis. Thus S_2 is the departing variable. This procedure of selecting the departing variable guarantees that no basic variable will ever be negative. The row corresponding to the departing variable is called 'key row'. The element on the intersection of key row and key column is called the 'key element' and is denoted by making a box (☐) in the simplex table.

At this point, we are ready to develop a new improved solution by revising Table I. Now you will need two tables to carry out the calculations: the old table and a new table.

To revise the key row, divide all values in the key row (S_2) by the value of the key element (i.e. 10) and replace departing variable (S_2) by the entering variable (X_2).

Also replace the new values in C_j column accordingly. Put all other values so obtained at the appropriate places. In our example, this new row becomes:

8 X_2 11 5/10 1 0 1/10

For other non-key rows, new values can be obtained as given below:

New row value = old row value - (corresponding to old **value in the key column X corresponding new** value in the revised key row).

In our example, the values of other row (S_1) can be obtained as follows:

For solution value column: = 300 - 20 X 11 = 80

For X_1 column: = 30 - 20 X 5/10 = 20

For X_2 column: = 20 - 20 X 1 = 0

For S_1 column: = 1 - 20 X 0 = 1

for S_2 column: = 0 - 20 X 1/10 = - 2

Therefore, the values for the new row becomes

$$0\ S_1\ 80\ 20\ 0\ 1\ -2$$

The values of C_j and $(C_j - Z_j)$ rows are calculated in the same way as discussed earlier.

The value of the objective function is calculated by substituting the values of the variables in the objective function. For example, in this improved solution

$$Z = 0 \times 80 + 8 \times 11 = 88$$

The new revised improved solution is shown in Table II.

Table II

$C_j \rightarrow \downarrow$	CONTRIBUTION PER UNIT		6	8	0	0	MINIMUM RATIO
	BASIC VARIABLES	SOLUTION VALUES	X_1	X_2	S_1	S_2	
0	S_1	80	[20]	0	1	−2	80/20 = 4 ← KEY ROW
8	X_2	11	5/10	1	0	1/10	11 ÷ 5/10 = 22
CONTRIBUTION LOSS PER UNIT Zj			4	8	0	8/10	$Z = 88$
NET CONTRIBUTION PER UNIT $Cj - Zj$			2	0	0	−8/10	

KEY ELEMENT (20) ↑ KEY COLUMN (X_1)

Proceeding in the same manner, we notice that variable X_1 will enter into the solution and variable S_1 will depart. The key element is 20. The next improved solution is given below in Table III.

Table III

$C_j \rightarrow \downarrow$	CONTRIBUTION PER UNIT		6	8	0	0
	BASIC VARIABLES	SOLUTION VALUES	X_1	X_2	S_1	S_2
6	X_1	4	1	0	1/20	−1/10
8	X_2	9	0	1	−1/40	3/20
$Z = 96$	CONTRIBUTION LOSS PER UNIT Zj		6	8	1/10	6/10
	NET CONTRIBUTION PER UNIT $Cj - Zj$		0	0	−1/10	−6/10

Since all entries in the net contribution per unit ($C_j - Z_j$) row are negative, therefore this indicates no sign of further improvement in the contribution, therefore, we have to terminate the procedure. Hence Table III provides the optimal solution i.e.,

$$X_1 = 4; X_2 = 9; S_1 = 0; S_2 = 0 \text{ and } Z = 96$$

These optimum values suggest that the further manufacturer should manufacture four tables and nine chairs and the maximum profit is Rs. 96.

You are requested to compare the results so obtained by simplex method with that of graphical method. What do you observe?

Activity A

Use simplex method to solve the following LP problem:

$$\text{Max } Z = 4X + 5Y + 8Z$$

subject to

$$X + Y + Z \leq 100$$

$$3X + 2Y + 4Z \leq 500$$

and

$$X, Y \geq 0$$

....

....

....

....

11.3 THE SIMPLEX METHOD (MINIMISATION CASE)

Until now we have limited the application of the simplex method only to maximisation problem. Now that you have developed some familiarity and understanding of the simplex method, we will apply it to a minimisation problem.

Another way to minimise the problem is to convert the problem by multiplying the objective function by - 1. This yields negative solution values whose sign must be reversed for application. We do not recommend this approach, since a direct solution is more convenient to use.

For convenience we will use example-2 that we solved graphically in the previous unit. We previously formulated the problem as:

$$\text{Min} \quad Z = 3X_1 + 2.5X_2$$

subject to

$$2X_1 + 4X_2 \geq 40$$

$$5X_1 + 2X_2 \geq 50$$

and

$$X_1, X_2 \geq 0$$

The first step in the simplex method is to convert the inequality constraint into equality constraint. In minimisation problems the slack variables are actually referred to as **surplus variables** (negative slack). Rather than representing unused capacity, they represent the excess amount by which a particular requirement is met. Converting inequalities into equalities and denoting surplus variables by S_1 and S_2, we get

$$2X_1 + 4X_2 - S_1 = 40$$

$$5X_1 + 2X_2 - S_2 = 50$$

By doing so, we quickly find that the surplus variables take on negative values, which violates the non-negativity restrictions. To overcome this problem, we need to add an additional variable to each constraint that has a positive value with a surplus variable. These new variables are called **artificial variables** because they are used to convert the origin artificially from infeasible to feasible. It may be emphasised that the use of an artificial variable is not limited to minimisation problems. An artificial variable may be used in maximisation problems as well. In general, as long as at least one greater-than or equal-to constraint is involved, regardless of minimisation or maximisation problem, artificial variable is used. In our example, introducing artificial variables, the constraints become:

$$2X_1 + 4X_2 - S_1 + A_1 = 40$$
$$5X_1 + 2X_2 - S_2 + A_2 = 50$$

The simplex method then selects the artificial variables as the initial basic variables. Therefore, the decision and surplus variables are non-basic variables and can be set equal to zero. Thus, we notice that the addition of the artificial variables has permitted us to convert the origin from infeasible point to feasible one. To correct this problem, we must add each artificial variable to the objective function. To ensure that the artificial variables are not basic variables in the optimal solution, we assign them very high costs. One convenient way of doing this is to assign each artificial variable a cost of M, where M defined to be a very large number. It should be emphasised that the artificial variables are used only as a mathematical convenience to obtain an initial basic feasible solution. This is the reason why the name artificial has been given to these variables since they are fictitious and have no physical meaning for the original problem. Introducing artificial variables in the objective function, we get

$$\text{Min } Z = 3X_1 + 2.25X_2 + 0S_1 + 0S_2 + MA_1 + MA_2$$

The modified problem is now to

$$\text{Min } Z = 3X_1 + 2.25X_2 + 0S_1 + 0S_2 + MA_1 + MA_2$$

subject to $2X_1 + 4X_2 - S_1 + A_1 = 40$

$5X_1 + 2X_2 - S_2 + A_2 = 50$

and $X_1, X_2, S_1, S_2, A_1, A_2 \geq 0$

We can now set up the initial simplex table exactly the same way as we did for maximisation problem. This is shown below:

Table IV

$C_j \rightarrow$ ↓			3	2.25	0	0	M	M	MINIMUM RATIO
	BASIC VARIABLES	SOLUTION VALUES	X_1	X_2	S_1	S_2	A_1	A_2	
M	A_1	40	2	4	−1	0	1	0	40/2 = 20
M	A_2	50	[5]	2	0	−1	0	1	50/5 = 10 ← KEY ROW
		Z_j	$7M$	$6M$	$-M$	$-M$	M	M	$Z = 90M$
		$C_j - Z_j$	$3-7M$	$2.25-6M$	M	M	0	0	

KEY ELEMENT ↑ KEY COLUMN

Table V

$C_j \rightarrow$ ↓	BASIC VARIABLES	SOLUTION VALUES	3 X_1	2.25 X_2	0 S_1	0 S_2	M A_1	M A_2	MINIMUM RATIO
M	A_1	20	0	[16/5]	−1	2/5	1	−2/5	20 × 5/16 = 6.25 ← KEY ROW
3	X_1	10	1	2/5	0	−1/5	0	1/5	10×5/2=25
		Z_j	3	$16M/5$ +6/5	$-M$	$2M/5$ −3/5	M	$-2M/5$ + 1/5	$Z=30+20M$
		$C_j - Z_j$	0	$-16M/5$ +21/20	M	$-2M/5$ +3/5	0	$7M/5$ −3/5	

KEY ELEMENT ↑ KEY COLUMN

Since we are trying to minimise costs, the entering variable will be selected for that variable for which the value of $(C_j - Z_j)$ is the largest negative. In our case, variable X_1 has the largest negative value. We therefore select X_2 as the entering variable. The selection of the departing variable is made exactly the same way as in the case of maximisation. Thus artificial variable A_2 is chosen as the departing variable. Therefore, the element which corresponds to the intersection of key row and key column, i.e., 5 becomes the key element. Having selected the key element, we can proceed in the same way as we did for the maximisation case. The revised improved solution is shown in Table V.

Proceeding in the same manner, we get Table VI as shown below:

Table VI

$C_j \rightarrow$ ↓	BASIC VARIABLES	SOLUTION VALUES	3 X_1	2.25 X_2	0 S_1	0 S_2	M A_1	M A_2
2.25	X_2	6.25	0	1	−5/16	1/8	5/16	−1/8
3	X_1	7.50	1	0	1/8	−1/4	−1/8	1/4
$Z = 36.5625$		Z_j	3	2.25	−21/64	−15/32	21/64	15/32
		$C_j - Z_j$	0	0	21/64	15/32	M −21/64	M −15/32

Examining the $(C_j - Z_j)$ row for Table VI, we find that there are no negative values. Thus the solution is optimal. Hence to meet the daily vitamins requirement, one should have 7.5 units of food F_1 and 6.25 units of food F_2 and the minimum cost is Rs. 36.5625.

Activity B

Use simplex method to solve the following LP problem:

$$\text{Min. } Z = 80X_1 + 100X_2$$

subject to
$$80X_1 + 60X_2 \geq 1500$$
$$20X_1 + 90X_2 \geq 1200$$

and
$$X_1, X_2, \geq 0.$$

11.4 THE SIMPLEX METHOD (MIXED TYPE CONSTRAINTS)

The application of simplex method was described earlier when the constraints were either $\geq$ type or $\leq$ type. A situation may arise when the constraints are of mixed type. Let us consider example 3 described earlier in the previous unit. To make the discussion more meaningful, let us change the last constraint of less than or equal to ($\leq$) type into a equality (=) type in example 3. Its modified formulation of LP problem can now be rewritten as given below:

$$\text{Min } \quad Z = 60X_1 + 80X_2$$

subject to
$$X_2 \geq 200$$
$$X_1 \leq 400$$
$$X_1 + X_2 = 500$$

and
$$X_1, X_2 \geq 0.$$

The problem can be converted into the standard form by adding slack, surplus and artificial variables in the set of constraints and assigning appropriate costs to these variables in the objective function. Thus, the problem can be stated as follows:

$$\text{Min } Z = 60X_1 + 80X_2 + 0S_1 + 0S_2 + MA_1 + MA_2$$

subject to
$$X_2 - S_1 + A_1 = 200$$
$$X_1 + S_2 = 400$$
$$X_1 + X_2 + A_2 = 500$$

and
$$X_1, X_2, S_1, S_2, A_1, A_2 \geq 0.$$

The initial simplex table for this problem is shown below in Table VII.

Table VII

$C_j \rightarrow$ ↓	BASIC VARIABLES	SOLUTION VALUES	60 X_1	80 X_2	0 S_1	0 S_2	M A_1	M A_2	MINIMUM RATIO
M	A_1	200	0	1	−1	0	1	0	200/1 ← KEY ROW
0	S_2	400	1	0	0	1	0	0	−
M	A_2	500	1	1	0	0	0	1	500/1 = 500
		Zj	M	$2M$	$-M$	0	M	M	$Z = 700M$
		$Cj - Zj$	$60-M$	$80-2M$	M	0	0	0	

KEY ELEMENT ↑ KEY COLUMN

Table VIII

$Cj \rightarrow$ ↓	BASIC VARIABLES	SOLUTION VALUES	60 X_1	80 X_2	0 S_1	0 S_2	M A_1	M A_2	MINIMUM RATIO
80	X_2	200	0	1	−1	0		0	
0	S_2	400	1	0	0	1	0	0	400/1 = 400
M	A_2	300	1	0	1	0	−1	1	300/1=300 ← KEY ROW
		Zj	M	80	$M-80$	0	$-M+80$	M	$Z = 16000 + 300M$
		$Cj - Zj$	$60-M$	0	$80-M$	0	$2M-80$	0	

KEY ELEMENT ↑ KEY COLUMN

Since our problem is of minimisation, therefore, instead of selecting the largest positive value in $(C_j - Z_j)$ row, we shall consider the largest negative value. Since both X_1 and X_2 have negative values in $(C_j - Z_j)$ row, therefore we select the largest negative value corresponding to X_2 column as key column. The key row is obtained by choosing the minimum ratio which corresponds to A_4. The intersection of these two elements gives us the key element, i.e. 1. Continuing as before, we get the improved solution as shown in Table VIII.

11.5 SPECIAL CASES IN APPLYING THE SIMPLEX METHOD

There are several special situations which you may come across while applying the simplex method manually. Let us consider some of these cases that may arise in applying the simplex method.

Tie for the Key Column

In simplex method, we proceed from one basic feasible solution to another improved solution until an optimum solution (if any) is arrived at. To go from one basic feasible solution to the next we select a key column. The key column can be selected by considering the most positive (or negative) value in $(C_j - Z_j)$ row in the case of maximisation (or minimisation) problem. But a situation may arise when at any interaction, two or more columns have exactly the same positive (or negative) value in $(C_j - Z_j)$ row. In that situation, selection for key column can be made arbitrarily.

Tie for the Key Row (Degeneracy)

The situations presented thus far had the property where the number of variables in the solution was the same as the number of constraints. If the number of positive variables in the solution is less than the number of constraints, the solution is said to be **degenerate.** In the simplex method, degeneracy occurs when there is a tie for the minimum ratio for choosing the departing variable. Again the choice for selecting the departing variable may be made arbitrarily.

Activity D

By using simplex method show that the following LP problem is degenerate.

$$\text{Max } Z = 2X_1 + 3X_2 + 4X_3$$

subject to

$$2X_1 + X_2 + 2X_3 \leq 50$$
$$X_1 + 3X_2 + X_3 \leq 25$$
$$X_1 + 2X_2 + X_3 \leq 26$$

and

$$X_1, X_2, X_3 \geq 0.$$

. . . .

. . . .

. . . .

. . . .

. . . .

Multiple Optimal Solution.

This situation occurs when the slope of any constraint and the objective function line is same. In such a case, there will be an infinite number of solutions and such solutions are called multiple optimal solutions. This situation can be recognised in the simplex method when one of the basic variables in the $(C_j - Z_j)$ row will have a value of zero. Hence a non-basic variable may be brought into the solution basis, thereby creating a new solution with the same value of the objective function.

Activity E

By using simplex method show that the following LP problem has an alternative optimal solution.

$$\text{Max} \quad Z = 4X_1 + 14X_2$$

subject to $2X_1 + 7X_2 \geq 21$

$7X_1 + 2X_2 \geq 21$

and $X_1, X_2, \geq 0.$

....

....

....

....

....

....

Unbounded Solution

In graphical method for unbounded solution, the feasible solution region extends indefinitely. In the simplex method, this situation occurs when there is no departing variable, i.e., the minimum ratio values are either infinite or negative.

No Feasible Solution

One of the abnormal situations occurs when the LP problem has an infeasible solution. The simplex method provides a clear indication that no feasible solution is possible. This is given by the fact that at least one of the artificial variables, which should be driven to zero by the simplex method, will be present as a positive basic variable in the final solution. When an infeasible solution is indicated, care should be taken to reconsider the fomulation of the LP problem.

Activity F

By using simplex method show that the following LP problem has infeasible solution.

$$\text{Min.} \quad Z = 3X_1 + 2X_2$$

subject to $2X_1 + X_2 \geq 2$

$3X_1 + 4X_2 \geq 12$

and $X_1, X_2, \geq 0.$

....

....

....

....

....

....

....

11.6 SUMMARY

The simplex method is a general method for solving any LP problem. To apply it, all relationships must be put in standard form. The simplex method starts searching from one basic feasible solution to another till an optimal solution is achieved. Applications of the simplex method to both maximisation and minimisation problems, having 'less than' 'equal to' and 'greater than' constraints have been discussed. We have also reviewed the special situations that commonly arise in applying the simplex method. Although the simplex method is efficient, large problems are not easy to solve manually but rather require computers. Numerous computer programmes have been developed for this purpose and are available on most computers.

11.7 KEY WORDS

Algorithm: A formalised systematic procedure for solving problems.

Artificial Variable: A variable that has no physical meaning and is used to obtain an initial basic feasible solution to a LP problem.

Basis : The set of basic variables that comprises a basic solution to a LP problem.

Basic Variables: The variables that normally take non-zero values to obtain a solution.

Degeneracy : A situation in the simplex method where a basic variable takes on the value zero.

Redundant Constraint: A constraint that does not affect the feasible solution region.

Simplex Method: An algorithm for solving LP problems investigates feasible corner points only, always maintaining or improving the objective function, until an optimal solution is obtained.

Slack Variable : A variable used to convert a less than or equal to constraint into an equality constraint by adding it to the left hand side of the constraint.

Surplus Variable: A variable used to convert a greater than or equal to constraint into an equality constraint by subtracting it from the left hand side of the constraint.

Unbounded Solution: A solution involving the infinite use of some resource.

11.8 FURTHER READINGS

Dannenbring, D.G. and M.K. Starr, 1981. *Management Science,* McGraw-Hill International Book Company: New York.

Gupta, M.P. and J.K. Sharma, 1983. *Linear Programming for Management,* National Publishing House: New Delhi.

Loomba, N.P. 1978. *Management-A Quantitative Perspective,* MacMillan Publishing Company: New York.

Turban, E. and J.R. Meredith, 1985. *Fundamentals of Management Science,* Business Publications Inc: New York.

DO IT YOURSELF EXERCISES

Section A: OR and Management Decision-making

1 Define Operations Research and discuss its importance in management decision-making.

2 What areas of Operations Research have made a significant impact on decision-making process? Why is it important to keep an open mind in utilising Operations Research techniques?

3 "Operations Research advocates a system approach and is concerned with optimisation. It provides a quantitative analysis for decision-making". Comment.

4 Discuss the methodology of Operations Research and describe the main stages involved in carrying out an Operations Research project.

5 'Model building' is a central element in Operations Research method. Give a description of the following basic types of model:

(i) Iconic, (ii) Analog, (iii) Mathematical (Symbolic) with particular reference to the special insights provided by each.

Section B: Linear Programming: Graphical Method

1 What is Linear Programming? What are its major limitations? Describe its applications in any functional area of management.

2 A publisher of text books is in the process of presenting a new book to the market. The book may be bound by either cloth or hard paper. Each clothbound book sold contributes Rs. 24, and each paper-bound book contributes Rs. 23. It takes 10 minutes to bind a cloth cover, and 9 minutes to bind a paperback. The total available time for binding is 800 hours. After considerable market survey, it is predicted that the cloth-cover sales will exceed at least 10,000 copies, but the paperback sales will be no more than 6,000 copies. Formulate the problem as a LP problem and use graphical method to solve the problem.

3 PQR Feed Company markets two feed mixes for cattle. The first mix, Fertilex, requires at least twice as much wheat as barley. The second mix, Multiplex, requires at least twice as much barley as wheat. Wheat costs Rs. 1.50 per kg., and only 1,000 kg. are available this month. Barley costs Rs. 1.25 per kg. and 1,200 kg. are available. Fertilex sells for Rs. 1.80 per kg. up to 99 kg. and each additional kg. over 99 sells for Rs. 1.65. Multiplex sells at Rs. 1.70 per kg. up to 99 kg. and each additional kg. over 99 sells for Rs. 1.55. Bharat Farms will buy any and all amounts of both mixes of PQR Feed Company. Set up the linear programming problem to determine the product mix that results in maximum profits.

4 A company manufacturing television sets and radios has four major departments: chasis, cabinet, assembly and final testing. Monthly capacities are as follows:

Department	Television capacity		Radio capacity
Chasis	1,500	or	4,500
Cabinet	1,000	or	8,000
Assembly	2,000	or	4,000
Testing	3,000	or	9,000

The contribution of television is Rs. 150 each and the contribution of radio is Rs. 250 each. Assuming that the company can sell any quantity of either product, determine the optimal combination of output.

5 A timber company cuts raw timber-oak and pine logs into wooden boards. Two steps are required to produce boards from logs. The first step involves removing the bark from the logs. Two hours are required to remove the bark from 1,000 feet of oak logs and three hours per 1,000 feet of pine logs. After the logs have been debarked, they must be cut into boards. It takes 2.4 hours for cutting 1,000 feet of oak logs into boards and 1.2 hours for 1,000 feet of pine logs. The bark-removing machines can operate up to 60 hours per week, while the cutting machines are limited to 48 hours per week. The company can buy a maximum of 18,000 feet of raw oak logs and 12,000 feet of raw pine logs each week. The profit per 1,000 feet of processed logs is Rs. 1,800 and Rs. 1,200 for oak and pine logs, respectively. Solve the problem to determine how many feet of each type of log should be processed each week in order to maximise profit.

6 Upon completing the construction of his house, Mr. Sharma discovers that 100 square feet of plywood scrap and 80 square feet of white pine scrap are in usable form for the construction of tables and book cases. It takes 16 square feet of plywood and 16 square feet of white pine to construct a book case. By selling the finished products to a local furniture store, Mr. Sharma can realise a profit of Rs. 25 on each table and Rs. 20 on each book-case. How can he most profitably use the left-over wood? Use Graphical Method to solve the problem.

7 A rubber company is engaged in producing three different kinds of tyres A, B and C. These three different tyres are produced at the company's two different plants with different production capacities. In a normal 8 hours working day, Plant 1 produces 50, 100 and 100 tyres of type A, B and C, respectively. Plant 2, produces 60, 60 and 200 tyres of type A, B and C, respectively. The monthly demand for type A, B and C is 2,500, 3,000 and 7,000 units, respectively. The daily cost of operation of Plant 1 and Plant 2 is Rs. 2,500 and Rs. 3,500, respectively. Use graphical method to determine the minimum number of days of operation per month at two different plants to minimise the total cost while meeting the demand.

8 Two products A and B are to be manufactured. A single unit of product A requires 2.4 minutes of punch press time and 5 minutes of assembly time. The profit for product A is Rs. 0.60 per unit. A single unit of product B requires 3 minutes of punch press time and 2.5 minutes of welding time. The profit for product B is Rs. 0.70 per unit. The

capacity of the punch press department available for these products is 1,200 minutes/ week. The welding department has an idle capacity of 600 minutes/week and assembly department has 1,500 minutes/week.

a) formulate the problem as linear programming problem.

b) Determine the quantities of products A and B so that total profit is maximised.

9 The XYZ company during the festival season combines two factors A and B to form a gift pack which must weigh 5 kg. At least 2 kg. of A and not more than 4 kg. of B should be used. The net profit contribution to the company is Rs. 5 per kg. for A and Rs. 6 per kg. for B. Use graphical method to find the optimal factor mix.

10 An advertising agency wishes to reach two types of audiences: customers with annual income of more than Rs. 15,000 (target audience A) and customers with annual income of less than Rs. 15,000 (target audience B). The total advertising budget is Rs. 2,00,000. One programme of TV advertising costs Rs. 50,000; one programme of radio advertising costs Rs. 20,000. For contract reasons, at least 3 programmes have to be on TV and the number of radio programme must be limited to 5. Surveys indicate that a single TV programme reaches 4,50,000 customers in target audience A and 50,000 in the target audience B. One radio programme reaches 20,000 in target audience A and 80,000 in the target audience B. Determine the media-mix to maximise the total reach

Section C: Linear Programming : Simplex Method

1 Explain various steps of the simplex method involved in the computation of an optimum solution to a linear programming problem.

2 Explain the meaning of 'basic feasible solution' and 'degenerate solution' in a linear programming problem.

3 Explain the use of 'artificial variable' in a linear programming problem.

4 A firm makes two types of furniture: chairs and tables. The contribution for each product as calculated by the accounting department is Rs. 20 per chair and Rs. 30 per table. Both products are processed on three machines M_1, M_2 and M3. The time required by each product and total time available per week on each machine are as follows:

Machine	Chair	Table	Available hours
M_1	3	3	36
M_2	5	2	50
M_3	2	6	60

How should the manufacturer schedule his production in order to maximise contribution?

5 The ABC manufacturing company can make two products P_1 and P_2. Each of the products requires time on a cutting machine and a finishing machine. Relevant data are:

	Product	
	P_1	P_2
Cutting hrs. (per unit)	2	1
Finishing hrs. (per unit)	3	3
Profit (Rs. per unit)	6	4
Maximum sales (unit per week)	N	200

The number of cutting hours available per week is 390 and the number of finishing hours available per week is 810. How much should be produced of each product in order to achieve maximum profit for the company?

6 A company makes two kinds of leather belts. Belt A is a high quality belt, and belt B is of lower quality. The respective profits are Re. 0.40 and Re. 0.30 per belt. Each belt of type A requires twice as much time as a belt of type B, and if all belts were of type B, the company could make 1,000 per day. The supply of leather is sufficient for only 800 belts per day (both A and B combined). Belt A requires a fancy buckle, and only 400 per day are available. There are only 700 buckles a day available for belt B.

What should be the daily production of each type of belt? Formulate the linear programming problem and solve it by simplex method.

7 Mr. Jain, the marketing manager of ABC Typewriter Company is trying to decide on how to allocate his salesmen to the Company's three primary markets. Market-1 is an urban area and the salesmen can sell, on an average 40 typewriters a week. Salesmen in the other two markets can sell, on an average, 36 and 25 typewriters per week, respectively. For the coming week, 3 of the salesmen will be on vacation, leaving only 12 men available for duty. Also because of the lack of company cars, maximum of 5 salesmen can be allocated to market area 1. The selling expenses per week for salesmen in each area are Rs. 800 per week for area 1, Rs. 700 per week for area 2, and Rs. 500 per week for area 3. The budget for the next week is Rs. 7500. The profit margin per typewriter is Rs. 150.

a) Formulate a linear programming model to determine how many salesmen should be assigned to each area in order to maximise profits.

b) Solve the above problem using the Simplex Method.

8 An animal feed company must produce 200 kg. of a mixture consisting of ingredients X_1 and X_2 daily. X_1 cost Rs. 3 per kg. and X_2 Rs. 8 per kg. No morethan 80 kg. of X_1 can be used, and at least 60 kg. of X_2 must be used. Find how much of each ingredient should be used if the company wants to minimise cost.

9 The ABC Printing Company is facing a tight financial squeeze and is attempting to cut costs wherever possible. At present it has only one printing contract and, luckily, the book is selling well in both the hardcover and paperback editions. It has just received a request to print more copies of this book in either the hardcover or paperback form. Printing cost

for hardcover books is Rs. 600 per 100 while printing cost for paperback is only Rs. 500 per 100. Although the company is attempting to economise, it does not wish to lay off any employees. Therefore, it feels obliged to run its two printing presses at least 80 and 60 hours per week, respectively. Press I can produce 100 hardcover books in 2 hours or 100 paperback books in 1 hour. Press II can produce 100 hardcover books in 1 hour or 100 paperback books in 2 hours. Determine how many books of each type should be printed in order to minimise cost.

10 A medical scientist claims to have found a cure for the common cold that consists of three drugs called K, S and H. His results indicate that the minimum daily adult dosage for effective treatment is 10 mg. of drug K, 6 mg. of drug S, and 8 mg. of drug H. Two substances are readily available for preparing pills or drugs. Each unit of substance A contains 6 mg., 1 mg. and 2 mg. of drugs K, S and H respectively, and each unit of substance B contains 2 mg, 3 mg, and 2 mg., of the same drugs. Substance A costs Rs. 3 per unit and substance B costs Rs. 5 per unit.

a) Find the least-cost combination of the two substances that will yield a pill designed to contain the minimum daily recommended adult dosage.

b) Suppose that the costs of the two substances are interchanged so that substance A costs Rs. 5 per unit and substance B costs Rs. 3 per unit. Find the new optimal solution.

BLOCK 4 ADVANCED DECISIONAL TECHNIQUES

This block comprising three units deals with some advanced decisional techniques. Block 3 dealing with linear programming is a prerequisite for this block. The first unit in this block deals with Transportation Models which can be converted into LP problems. The use of Transportation Models, the solution techniques and certain special situations are discussed. Next unit in this block is on Assignment Models, which are specially structured linear programs. Assignment problem deals with allocation of certain resources to certain activities with the goal of optimisation. Methods of solving assignment problems using Transportation Model, and a more efficient model namely Hungarian method are discussed. The last unit in this block discusses PERT and CPM. These are two well known techniques for project management. Solution procedures with deterministic and probabilistic times for completion of activities are discussed.

UNIT 12 TRANSPORTATION MODELS

Objectives

After studying this unit, you should be able to:

- recognise a transportation problem
- convert a transportation problem into a LP problem
- develop a trasportation table and apply the transportation method to get the solution.
- describe special situations in trasportation problem such as: multiple optimal solutions, unbalanced trasportation problems, degeneracy, maximisation in transportation problem, and prohibited routes.

Structure

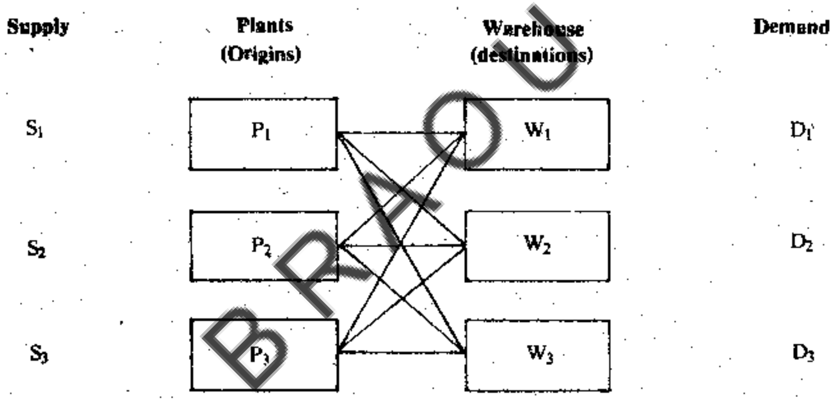
12.1 INTRODUCTION

Transportation plays an important role in our economy and in managerial decision making. In this unit, we will not cover the entire field of transportation, rather we will focus on one particular type of transportation problem. This type of transportation problem usually involves the physical movement of goods and services from various supply origins to multiple demand destinations within the given constraints of supply and demand in such a way that the total transportation cost is minimised. A trasportation problem is a special type of Linear Programming (LP) problem that may be solved using the simplex method. But even small transportation problems will involve lots of variables and constraints, so if you want to use simplex method, it will consume a lot of computational time even if you use computer. However, a transportation problem has a special mathematical structure which permits it to be solved by a fairly efficient method known as transportation method.

12.2 TRANSPORTATION MODELS

Transportation models deal with the trasportation of a product manufactured at different plants (supply origins) to a number of different warehouses (demand destinations). The objective is to satisfy the demand at destinations given the supply constraints at the minimum transporation cost. Transportation models typically arise in situations involving physical movement of goods from plants to warehouses, warehouses to wholesalers, wholesalers to retailers and retailers to customers. Solution of the transportation models requires the determination of how many units should be transported from each supply origin to each demand destination in order to satisfy all the destination demands while minimising the total associated cost of transportation. The transportation problem can be formulated as a standard LP problem. We will see how this is done later. However, the characteristics of a transportation model are such that it is usually solved by a specialised procedure (algorithm or method) rather than by simplex method.

The easiest way to recognise a transportation problem is to consider a typical situation as shown in the figure. Assume that a manufacturer has three plants P_1, P_2 and P_3 producing the same product. From these plants, the product is transported to three warehouses W_1, W_2 and W_3. Each plant has a limited supply (capacity), and each warehouse has specific demand. Each plant can transport to each warehouse, but the transportation costs vary for differing combinations. The problem is to determine the quantity each plant should transport to each warehouse in order to minimise total transportation costs.



It was stated earlier that a transportation problem is a special type of LP problem. To illustrate this, let us see how this problem can be formulated as a LP problem. Let X_{ij} represent the quantity transported from plant P_i to warehouse W_j. Similarly, C_{ij} is the per unit transportation cost from plant P_i to warehouse W_j. The objective is to minimise total transportation costs. The LP objective function is,

Minimise

$$Z = C_{11} X_{11} + C_{12} X_{12} + C_{13} X_{13} + C_{21} X_{21} + C_{22} X_{22} + C_{23} X_{23} + C_{31} X_{31} + C_{32} X_{32} + C_{33} X_{33}$$

The supply constraints are

$$X_{11} + X_{12} + X_{13} = S_1$$
$$X_{21} + X_{22} + X_{23} = S_2$$
$$X_{31} + X_{32} + Z_{33} = S_3$$

The demand constraints are

$$X_{11} + X_{21} + X_{31} = D_1$$
$$X_{12} + X_{22} + X_{32} = D_2$$
$$X_{13} + X_{23} + X_{33} = D_3$$

and $X_{ij} \geq O$ for $i = 1, 2, 3; j = 1, 2, 3$

Additionally, it is also assumed that the total supply available at the plants will exactly satisfy the demand required at the destinations i.e.,

$$S_1 + S_2 + S_3 = D_1 + D_2 + D_3$$

The problem where total supply is equal to total demand is called the balanced transportation problem. If the total supply is not equal to total demand, then such types of problems are referred to as unbalanced transportation problems and will be discussed later in this unit.

A total of six constraints is given, one each for demand and supply. It can be proved that for the above transportation problem only five (rather than six) constraints are needed to get the feasible solution. Since total supply is equal to total demand, any solution satisfying five of the six constraints will also satisfy the remaining constraint. Therefore, in general, if we have m rows (supply) and n columns (demand) in a given transportation problem, then the problem can be solved completely if we have exactly (m+n-1) basic varibles. Thus, a basic feasible solution to a balanced transportation problem would be represented in the transportation table as having exactly (m+n-1) positive $X_{ij's}$ (allocations). These allocations are referred to as occupied cells and others as unoccupied (empty) cells. It the number of occupied cells is less than (m+n-1) allocations then it becomes a case of degeneracy. 'Degeneracy' in transportation problems will be discussed in a later section of this unit.

Activity A

What are the characteristics of a transportation problem?

....

....

....

....

....

Activity B

For a transportation problem with seven plants and twelve warehouses, how many decision variables, constraints and occupied cells (for a basic feasible solution) will there be?

....

....

....

....

12.3 SOLUTION PROCEDURE FOR TRANSPORTATION PROBLEMS

We now turn our attention to the solution procedure for a transportation problem known as the transportation method. Conceptually, the transportation method is similar to the simplex method. We begin with an initial feasible solution. This initial feasible solution may or may not be optimal. The only way you can find it out is to test it. If the solution is not optimal, it is revised and the test is repeated. Each iteration should bring you closer to the optimal solution.

Let us consider the example discussed earlier in this unit. The supply of each plant, the demand of each warehouse, and per unit transportation costs are shown in Table 1.

Table 1

		Warehouse			Supply
		W_1	W_2	W_3	S_i
	P_1	7	6	9	20
Plant	P_2	5	7	3	28
	P_3	4	5	8	17
Demand	D_j	21	25	19	65

The format followed in Table 1 will be used throughout the unit. Each row corresponds to a specific plant and each column corresponds to a specific ware house. Plant supplies are shown to the right of the table and warehouse requirements are shown below the table. The larger box (also known as cells) at the intersection of a specific row and column will contain both quantity to be transported and per unit transportation cost. The quantity to be transported will be shown in the centre of the box and will be encircled and the per unit transportation cost is shown in the smaller rectangular box at the left hand side corner.

12.4 FINDING AN INITIAL FEASIBLE SOLUTION

There are a number of methods available for generating an initial feasible solution for a transportation problem. We will consider three of the following:

i) North West Corner Method

ii) Least Cost Method

iii) Vogel's Approximation Method

North West Corner Method (NWCM): the simplest of the procedures used to generate an initial feasible solution is NWCM. It is so called because we begin with the North West or upper left corner cell of our transportation table. Various steps of this method can be summarised as under:

Step 1: Select the North West (upper left-hand) corner cell of the transportation table and allocate as many units as possible equal to the minimum between available supply and demand requirement i.e. min (S_1, D_1).

Step 2: Adjust the supply and demand numbers in the respective rows and columns allocation.

Step 3:

a) If the supply for the first row is exhausted, then move down to the first cell in the second row and first column and go to step 2.

b) If the demand for the first column is satisfied, then move horizontally to the next cell in the second column and first row and go to step 2.

Step 4: If for any cell, supply equals demand, then the next allocation can be made in cell either in the next row or column.

Step 5: Continue the procedure until the total available quantity is fully allocated to the cells as required.

Remark 1: The quantities so allocated are circled to indicate the value of the corresponding variable.

Remark 2: Empty cells indicate the value of the corresponding variable as zero, i.e. no unit is shipped to this cell.

To illustrate the NWCM, let us consider the trasportation Table 1 as given in the previous example.

As stated in this method, we start with the cell (P_1, W_1) and allocate the min (S_1, D_1) = min(20, 21) = 20. Therefore we allocate 20 units to this cell which completely exhausts the supply of Plant P_1 and leaves a balance of (21 - 20) = 1 unit of demand at warehouse W_1. Now, we move vertically downward to the cell (P_2, W_1). At this stage, the largest allocation possible is the min $(S_2, D_1 - 20)$ = min (28, 1) = 1. This allocation of 1 unit to the cell (P_2, W_1) completely satisfies the demand of warehouse W_1. However, this leaves a balance of (28-1) = 27 units of supply at plant P_2. Now, we move again horizontally to the cell (P_2, W_2). Since the demand of warehouse W_2 is 25 units while supply available at plant P_2 is 27 units, therefore, the min (27, 25) = 25 units are allocated to the cell (P_2, W_2). The demand of warehouse W_2 is now satisfied and a balance of (27 - 25) = 2 units of supply remain at plant P_2. Moving again horizontally, we allocate two units to the cell (P_2, W_3) which completely exhaust the supply at plant P_2 and leaves a balance of 17 units demand at warehouse W_3. Now we move vertically downward to the cell (P_3, W_3). At this cell, 17 units are available at plant P_3 and 17 units are required at warehouse W_3. So we allocate 17 units to this cell (P_3, W_3). Hence we have made all the allocations. It may be noted here that there are 5 (3 + 3 - 1) allocation which are necessary to proceed further. The initial feasible solution is shown below in Table 2.

The total transportation cost for this initial solution is:

Total cost = 20 x 7 + 1 x 5 + 25 x 7 + 2 x 3 + 17 x 8 = Rs. 462

Table 2

	Warehouse W_1	W_2	W_3	Supply S_i
P_1	7 (20)	6	9	~~20~~ 0
P_2	5 (1)	7 (25)	3 (2)	~~28~~ ~~27~~ ~~2~~ 0
P_3	4	5	8 (17)	17
D_j	~~21~~ ~~1~~ 0	~~25~~ 0	~~19~~ ~~17~~ 0	65

Activity C

Set up a trasportation table (matrix) by using the data given below:

Plant	Supply	Warehouse	Demand
P_1	500	W_1	400
P_2	800	W_2	700
P_3	1000	W_3	1200

Transportation Costs (Rs./Unit)

From	To W_1	W_2	W_3
P_1	2	3	6
P_2	2	1	5
P_3	4	6	12

Also find an initial solution by using NWCM. What is the total transportation cost for this initial solution?

....

....

....

Least Cost Method (LCM)

The allocation according to this method is very useful as it takes into consideration the lowest cost and therefore, reduces the computation as well as the amount of time necessary to arrive at the optimal solution. Various steps of this method can be summarised as under:

Step 1:

a) Select the cell with the lowest transportation cost among all the rows or columns of the transportation table.

b) If the minimum cost is not unique, then select arbitrarily any cell with this minimum cost.

Step 2: Allocate as many units as possible to the cell determined in Step 1 and eliminate that row (column) in which either supply is exhausted or demand is satisfied.

Step 3: Repeat Steps 1 and 2 for the reduced table until the entire supply at different plants is exhausted to satisfy the demand at different warehouses. Consider again the transportation Table 1 for the illustration of least cost method. As stated in the algorithm, we start with the cell which has the minimum cost, i.e., cell (P_2, W_3) in which the cost is 3. We allocate min (S_2, D_3) = min (28, 19) = 19 units to fulfil the complete requirement of warehouse W_3. Since the demand of warehouse W_3 is satisfied, therefore, column W_3 will not be considered any more. In the reduced table, again locate the minimum cost cell, i.e., (P_3, W_1). Allocate min (S_3, D_1) = min (17, 21) = 17 units to this cell which exhaust the supply at plant P_3. Thus row P_3 will also not to be considered any more. Proceeding in the same way, the search for minimum cost cell will continue till all supply and demand conditions are satisfied. The initial solution by this least cost method is shown in Table 3.

Table 3

		Warehouse			Supply
		W_1	W_2	W_3	S_i
Plant	P_1	7	6 (20)	9	20 0
	P_2	5 (4)	7 (5)	3 (19)	28 9 5 0
	P_3	4 (17)	5	8	17 0
Demand	D_j	21 4 0	25 5 0	19 0	65

The total transportation cost associated with this method is given below:

Total cost = 20 x 6 + 4 x 5 + 5 x 7 + 19 x 3 + 17 x 4 = Rs. 300

This total cost is less than the total cost obtained by NWCM. Therefore this method is generally preferred over the NWCM.

Activity D

Use LCM for getting the initial solution for the problem given in Activity C. Also find the total transportation cost for this solution.

. . . .

. . . .

. . . .

. . . .

Vogel's Approximation Method (VAM)

This method is preferred over the other two methods because the initial basic feasible solution obtained is either optimal or very close to the optimal solution. Therefore, the amount of time required to arrive at the optimal solution is greatly reduced. Various steps of this method are summarised as under:

Step 1: Compute a penalty for each row and column in the transportation table. The penalty for a given row and column is merely the difference between the smallest cost and the next-smallest cost in that particular row or column.

Step 2: Identify the row or column with the largest penalty. In this identified row or column, choose the cell which has the smallest cost and allocate the maximum possible quantity to this cell. Delete the row or column in which supply is exhausted or demand is satisfied.

Whenever the largest penalty among rows and columns is not unique, make an arbitrary choice.

Step 3: Repeat steps 1 and 2 for the reduced table until the entire supply at plants are exhausted to satisfy the demand at different warehouses.

Consider again the transportation Table 1 for the explanation of Vogel's approximation method. Calculate the difference between the smallest and the next smallest cost in each column and row. Record them as shown in Table 4. In the first round, column W_3 for which the penalty (i.e. 5) is largest in both columns and rows will be considered for allocation. This allocation is made in that cell of column W_3 for which the cost is minimum, i.e., cell (P_2, W_3). Allocate min (S_2, D_3) = min. (28, 19) = 19 units to this cell so that the demand at warehouse W_3 is completely satisfied and therefore column W_3 will not be considered for further allocation. Adjust the supply at plant P_2 which now can supply only 9 units. In the second round, row P_2 has the largest penalty (i.e. 2) and therefore row P_2 is selected for allocation to that cell for which the cost is minimum, i.e., cell (P_2, W_1). Allocate min (9, 21) = 9 units to the cell (P_2, W_1). This exhausts the supply at plant P_2 and therefore row P_2 will not be considered any more. Similarly in the third round, column W_1 has the largest penalty (i.e. 3) and therefore column W_1 is selected for allocation to that cell for which the cost is minimum, i.e. cell (P_3, W_1). Allocate min (17,12) = 12 units to this cell. This allocation satisfies the demand at warehouse W_1 and therefore column W_1 will not be considered any more. Now only two cells, viz, (P_1, W_2) and (P_2, W_2) are left for allocation. Out of the two cells, cell (P_1, W_2) has the least cost, therefore we allocate the remaining 20 units to this cell. It may be observed that the supply and demand constraints are met exactly. The initial solution by this method is shown below in Table 4.

Table 4

		Warehouse W_1	Warehouse W_2	Warehouse W_3	Supply S_i	Row Penalties
Plant	P_1	7	6 (20)	9	~~20~~ 0	1 1 1
	P_2	5 (9)	7	3 (19)	~~28~~ ~~9~~ 0	2 ~~2~~
	P_3	4 (12)	5 (5)	8	~~17~~ ~~5~~ 0	1 1 1
Demand	D_j	~~21~~ ~~12~~ 0	~~25~~ ~~20~~ 0	~~19~~ 0	65	
Column Penalties		1 1 3	1 1 1	5		

The total transportation cost associated with this method is:

Total cost = 20 x 6 + 9 x 5 + 19 x 3 + 12 x 4 + 5 x 5 = Rs. 295

It may be pointed out that the initial solution of Vogel's method is supposed to be better than NWCM and LCM as the total trasportation cost obtained by this method is less than the other two methods. It may be emphasised at this stage that this may not be always true.

Activity E

Use Vogel's method for finding the inital solution for the problem given in Activity C. Also find the total transportation cost by this method.

....

....

....

....

12.5 FINDING THE OPTIMAL SOLUTION

Once an initial solution has been found, the next step is to test that solution for optimality. The following two methods are widely used for testing the solution:

1 Stepping Stone Method
2 Modified Distribution (MODI) Method

The two methods differ in their computational approach but give exactly the same results and use the same testing procedure. The procedure being used is to test each unoccupied cell one at a time, by computing the cost change. If the inclusion of any unoccupied cell can decrease the trasportation cost then this unoccupied cell will be considered for allocation in the improved solution. We select that unoccupied cell for allocation for which the cost change is most negative. This procedure is continued till we get the optimal solution.

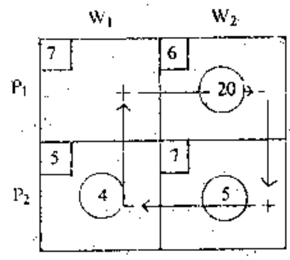
Stepping-Stone Method

In this method we calculate the net cost change that can be obtained by introducing any of the unoccupied cells into the solution. The important rule to keep in mind is that every increase (or decrease) in supply at one occupied cell must be associated by a decrease (or increase) in supply at another. The same rule holds true for demand. Thus there must be two changes in every row or column that is changed-one change increasing the allocation (or quantity) and one change decreasing it. This is easily done by evaluating re-allocations in a closed path sequence with only right-angle turns permitted.

The criterion for making a re-allocation is simply to know the desired effect upon costs. The net cost change is determined by listing the unit costs associated with each cell and then summing over the path to find the net effect. Signs are alternate from positive (+) to negative (-) depending upon whether shipments are being added or subtracted at a given point. A negative sign on the net cost change indicates that a cost reduction can be made by making the change. The positive sign on the net cost change indicates a cost increase.

To demonstrate the application of this method, consider the initial solution given by least cost method as shown in Table 3. Let us evaluate the unoccupied cell (P_1, W_1). The shipment of one unit to this cell will incur an additional cost of Rs. 7. This requires in turn that one unit be decreased from cell (P_1, W_2) which decreases cost by Rs. 6. But to keep the balance between supply and demand we have to add one unit to cell (P_2, W_2) which increases cost by Rs. 7 and finally one unit is decreased from cell (P_2, W_1) which decreases cost by Rs. 5. To determine the net cost change, let us list down the changes as shown below:

Cell	Changes in allocation	Cost change (Rs.)
(P_1, W_1)	+1	+7
(P_1, W_2)	−1	−6
(P_2, W_2)	+1	+7
(P_2, W_1)	−1	−5
	Net cost change =	+3



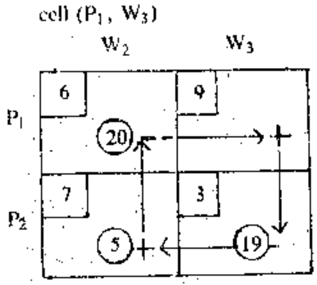
This indicates that if the unoccupied cell (P_1, W_1) is made occupied then the total transportation cost will be increased by Rs. 3 per unit supplied. This transfer of shipment of one unit is also shown in the right hand side table by making a closed path.

Similarly other unoccupied cells can also be evaluated in the same manner as shown below:

Unoccupied cell	Closed path	Net cost change (Rs.)	Remarks
(P_1, W_3)	(P_1, W_3)→(P_2, W_3)→(P_2, W_2) →(P_1, W_2)	$9-3+7-6=+7$	cost increases
(P_3, W_2)	(P_3, W_2)→(P_3, W_1)→(P_2, W_1) →(P_2, W_2)	$5-4+5-7=-1$	cost decreases
(P_3, W_3)	(P_3, W_3)→(P_3, W_1)→(P_2, W_1) →(P_2, W_3)	$8-4+5-3=+6$	cost increases

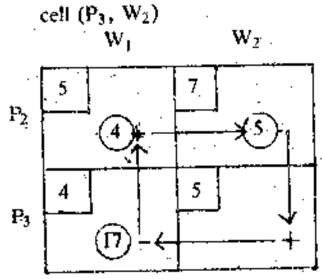
This transfer of shipment of one unit to make it closed path is also shown below for the other unoccupied cells:

For unoccupied cell (P_1, W_3)



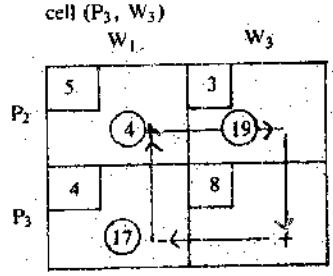
Cost Change: $9-3+7-6=+7$

For unoccupied cell (P_3, W_2)



$5-4+5-7=-1$

For unoccupied cell (P_3, W_3)



$8-4+5-3=+6$

Thus we observe that only unoccupied cell (P_3, W_2) for which the cost change is-1 will decrease the total transportation cost by Re 1 per unit. Therefore, the unoccupied cell (P_3, W_2) will be considered for further reduction in the cost. The next question which comes to our mind is, how much quantity can be shipped to cell (P_3, W_2). The maximum quantity we can ship to cell (P_3, W_2) is exactly the minimum quantity of those cells with the minus sign in the closed path as shown above. In this case, cell (P_3, W_1) has 17 and cell (P_2, W_2) has 5 as the quantity to be shipped. Therefore the minimum of 5 and 17 is 5 which has to be the maximum quantity to be shipped to cell (P_3, W_2). It should be obvious that if we ship more than 5 units to cell (P_3, W_2), we have to assign some negative value to cell (P_2, W_2) in order to meet supply and demand requirements. Since we cannot ship negative quantity, therefore the maximum quantity to be shipped to cell (P_2, W_2) has to be 5. The new solution thus obtained is shown below in Table 5.

Table 5

		Warehouse			Supply
		W	W_2	W_3	S_i
Plant	P_1	7	6 (20)	9	20
	P_2	5 (9)	7	3 (19)	28
	P_3	4 (12)	5 (5)	8	17
Demand	D_j	21	25	19	65

The total transportation cost of the improved solution is:

Total cost = 20 x 6 + 9 x 5 + 19 x 3 + 12 x 4 + 5 x 5 = Rs. 295

The next step is to evaluate again all the unoccupied cells of the improved solution and see whether the total cost can be further reduced. The unoccupied cells of this improved solution are evaluated in the same manner as shown below:

Unoccupied cell	Closed path	Net cost change (Rs.)	Remark
(P_1, W_1)	(P_1, W_1)→(P_1, W_2)→(P_3, W_2) (P_3, W_1)	7 − 6 + 5 − 4 = +2	cost increases
(P_1, W_3)	(P_1, W_3)→(P_2, W_3)→(P_2, W_1) (P_3, W_1)→(P_3, W_2)→(P_1, W_2)	9 − 3 + 5 − 4 + 5 − 6 = +6	cost increases
(P_2, W_2)	(P_2, W_2)→(P_3, W_2)→(P_3, W_1) (P_2, W_1)	7 − 5 + 4 − 5 = +1	cost increases
(P_3, W_3)	(P_3, W_3)→(P_3, W_1)→(P_2, W_1) (P_2, W_3)	8 − 4 + 5 − 3 = +6	cost increases

Since all the unoccupied cells have positive values for the net cost change, therefore, there is no way to improve the solution any more. Hence we have reached the optimum solution.

The transportation schedule as shown in Table 5 and the total transportation cost of the optimal solution are as given below:

From Plant	Transported to warehouse	Quantity	Unit cost	Total cost
P_1	W_2	20	6	120
P_2	W_1	9	5	45
P_2	W_3	19	3	57
P_3	W_1	12	4	48
P_3	W_2	5	5	25
				Total transportation cost = Rs. 295

The stepping stone method that we have applied can be summarised as follows:

1 Make sure that the number of occupied cells is exactly equal to m+n - 1, where m = number of rows and n = number of columns.

2 Evaluate each unoccupied cell by following its closed path and determine its net cost change. If all net cost changes have zero or positive sign, then the solution is optimum. If there exist negative net cost changes (change), find the unoccupied cell with the largest negative value. This is the unoccupied cell where a shipment should be made.

3 Determine the quantity to be shipped to the selected unoccupied cell. Trace the closed path for the unoccupied cell and identify the minimum quantity by considering the minus sign in the closed path. Ship this quantity and find the new solution. Go to step 1.

Activity F

Use stepping stone method to arrive at the optimal solution for the problem given in Activity C.

....

....

....

....

The MODI method is a more efficient procedure of evaluating the unoccupied cells. Instead of evaluating the unoccupied cells one at a time by tracing the closed path, as in the stepping stone method, in the MODI method it is possible to evaluate all of the unoccupied cells simultaneously. Consequently in the MODI method we do not have to trace all the closed paths. Rather, we trace the closed path of only one unoccupied cell for which the cost reduction is maximum.

In order to demonstrate the MODI method, let us go back to the initial solution we obtained by Vogel's method as shown in Table 4. To illustrate the MODI method, we have to alter the transportation table by assigning an additional row and column. The modified transportation table of the initial solution is shown in Table 6.

Table 6

	Warehouse W₁	W₂	W₃	Supply S_i	U_i
Plant P_1	[7]	[6] (20)	[9]	20	U_1
P_2	[5] (9)	[7]	[3] (19)	28	U_2
P_3	[4] (12)	[5] (5)	[8]	17	U_3
Demand D_j	21	25	19	65	
V_j	V_1	V_2	V_3		

Note that we have added column U_i to indicate row values and row V_j to indicate column values. Now let us define these variables before we proceed further.

U_i = Value for the ith row (plant)
V_j = Value for the jth column (warehouse)

For the occupied cells, the following relationship exists:

$$C_{ij} = U_i + V_j$$

For example, the unit transportation cost for the five occupied cells can be described as:

$$C_{12} = U_1 + V_2 = 6$$
$$C_{21} = U_2 + V_1 = 5$$
$$C_{23} = U_2 + V_3 = 3$$
$$C_{31} = U_3 + V_1 = 4$$
$$C_{32} = U_3 + V_2 = 5$$

In the above equations, we have six unknown variables (row and column values) and five equations. In order to obtain a solution to determine the row and column values, one of the variables must be chosen and given an arbitrary value of zero. We select U_2 and assign a zero value to it. With $U_2 = 0$, we can identify the values of the remaining variables in the above relationships which are shown below:

$$U_2 + V_1 = 5 \Rightarrow 0 + V_1 = 5 \Rightarrow V_1 = 5$$
$$U_2 + V_3 = 3 \Rightarrow 0 + V_3 = 3 \Rightarrow V_3 = 3$$
$$U_3 + V_1 = 4 \Rightarrow U_3 + 5 = 4 \Rightarrow U_3 = -1$$
$$U_3 + V_2 = 5 \Rightarrow -1 + V_2 = 5 \Rightarrow V_2 = 6$$
$$U_1 + V_2 = 6 \Rightarrow U_1 + 6 = 6 \Rightarrow U_1 = 0$$

We now proceed to evaluate cost change for all the unoccupied cells by using the following cost change formula:

$$\text{cost change} = C_{ij} - U_i - V_j$$

As in the stepping stone method, if an unoccupied cell has a negative cost change, it indicate that an improved solution is possible. When all cost changes have zero or positive values, the optimal solution is reached.

Table 7 presents the initial solution of the problem with the corresponding row and column values. The net cost change for unoccupied cell is evaluated as follows:

Unoccupied cell	$C_{ij} - U_i - V_j$	net cost change
(P_1, W_1)	7 - 0 - 5 = 2	+2
(P_1, W_3)	9 - 0 - 3 = 6	+6
(P_2, W_2)	7 - 0 - 6 = 1	+1
(P_3, W_3)	8 + 1 - 3 = 6	+6

Table 7

		Warehouse			Supply	
	W_1	W_2	W_3		S_i	U_i
Plant P_1	7 / +2	6 / (20)	9 / +6		20	$U_1 = 0$
P_2	5 / (9)	7 / +1	3 / (19)		28	$U_2 = 0$
P_3	4 / (12)	5 / (5)	8 / +6		17	$U_3 = -1$
Demand D_j	21	25	19		65	
V_j	$V_1 = 5$	$V_2 = 6$	$V_3 = 3$			

The net cost changes derived above corresponds with those we calculated in the stepping stone method. The cost change calculated for all the unoccupied cells are also shown in the right hand side bottom of the unoccupied cells and these values of cost change indicate that there is no negative value. Therefore, we have reached the optimal solution. The solution is identical with the one we derived through the stepping stone method. Again the total trasportation cost by MODI method is Rs. 295.

Activity G

Use MODI method to arrive at the optimal solution for the problem given in Activity C.

. . . .

. . . .

. . . .

. . . .

. . . .

12.6 SPECIAL CASES IN TRANSPORTATION PROBLEMS

Now that we have seen how to solve transportation problems, we next need to consider some special situations that can arise. An understanding of how to handle these situations in the transportation problem framework will allow you to apply this method to a wide variety of practical situations.

Multiple Optimal Solutions

As with other linear programming problems, a transportation problem can have more than one optimal solution. This would be indicated when one or more of the unoccupied cells have zero value for the net cost change in the optimal solution. Thus a reallocation to the cell having a net cost change equal to zero will have no effect on the total transportation cost.

This reallocation will provide another solution with the same total transportation cost, but the routes employed will be different from those for the original optimal solution. Multiple optimal solutions are important because they provide management with added flexibility in management decision-making.

Activity H

Obtain the optimal solution for the following transportation problem.

Plant	Warehouse W_1	W_2	W_3	Supply S_i
P_1	8	5	6	120
P_2	14	10	12	80
P_3	3	9	10	80
Demand D_j	150	70	60	280

Also indicate the alternative optimal solution for the given problem, if any.

Unbalanced Transportation Problems

So far we have been dealing with transportation problems in which the total supply at origins was equal to the total demand at destinations. In real life problems, however, this may not be always true. Thus the transportation problems with unequal supply and demand is said to be unbalanced transportation problems. With a little modification, an optimal solution of such problems can also be obtained by the methods discussed earlier.

Supply Exceeds Demand: When the total supply at origins is greater than the total demand at destinations, i.e.,

$$\sum_{i=1}^{m} S_i > \sum_{j=1}^{n} D_j$$

We add a dummy destination (column) in the transportation table with zero transportation costs associated with this dummy column so that the problem becomes balanced. Then the problem is solved by the methods used earlier for a balanced transportation problem.

To illustrate the application of unbalanced transportation problem, consider the following transportation problem where supply exceeds demand.

Since the total supply of 76 units at plant P_1, and P_2, and P_3 exceeds the total demand of 71 units at warehouse W_1, W_2 and W_3 by 5 units, the given problem is unbalanced. We add a dummy warehouse W_4 with its demand of 5 units and transportation costs zero from all plants to this dummy warehouse.

Now the balanced trasportation problem is given as shown in Table 8:

Table 8.

		Warehouse				
		W_1	W_2	W_3	W_4	Supply S_i
Plant	P_1	11	21	16	0	14
	P_2	7	17	13	0	26
	P_3	11	23	21	0	36
Demand D_j		18	28	25	5	76

Once the unbalanced transportation problem is converted into a balanced transportation problem, then the problem can be solved in the usual manner.

Demand Exceeds Supply: When the total demand at destinations is greater than the total supply at origins, i.e.,

$$\sum_{j=1}^{n} D_j > \sum_{i=1}^{m} S_i$$

we add a dummy plant (row) in the transportation table with zero transportation costs, so that the problem becomes balanced. Then the problem is solved in the usual manner. Therefore, in an unbalanced transportation, once the dummy facility is added, the problem is solved exactly as described earlier. The dummy facility is treated no differently than a real facility. The interpretation of the optimal solution should include an adjustment for the dummy facility. If dummy supply is added, the amounts scheduled to be shipped from the supply points have to be treated as unfulfilled demand by the demand destinations that are to receive those shipments. In a similar manner, amounts scheduled to be received by a dummy demand represent unused capacity at those supply points that are to make the shipments.

Activity I

For the foll~wing unbalanced transportation problem, derive the optimal solution.

		Warehouse			Supply
		W_1	W_2	W_3	S_i
Plant	P_1	8	5	6	120
	P_2	15	10	12	80
	P_3	3	9	10	80
Demand	D_j	150	70	90	310/280

Degeneracy in the Transportation Problem

It has been already established that to an m origin and n destination transportation problem, the number of occupied cells at any solution stage must be (m + n - 1). But if the number of occupied cells is less than (m + n - 1) at any stage of solution, then the problem is said to have a degenerate solution. Degeneracy can occur at two stages: either (i) at the initial solution or (ii) during testing of optimal solution.

Stage 1: Degeneracy occurs at the initial solution: To resolve degeneracy at the initial stage, we make use of an artificial quantity, denoted by the Greek letter ϵ (epsilon). If ϵ is placed in the unoccupied cell, then that cell is considered to be occupied. The quantity ϵ is so small that it does not affect the supply and demand constraints (rim conditions). For calculation purposes, the value of ϵ is assumed to be zero. Its use and significance are confined solely to the evaluation of unoccupied cells. Once a ϵ is introduced into the solution, it will remain there until degeneracy is removed or a final solution is arrived at, whichever occurs first. To explain the use of ϵ consider the following transportation problem:

		Warehouse				Supply
		W_1	W_2	W_3	W_4	S_i
Plant	P_1	23	42	33	11	2
	P_2	17	25	45	20	3
	P_3	3	12	8	18	12
Demand	D_j	4	1	5	7	17

Let us use NWCM to get the initial solution. The initial solution is shown below in Table 9.

Table 9

		Warehouse				Supply
		W_1	W_2	W_3	W_4	S_i
Plant	P_1	23 (2)	42	33	11	2
	P_2	17 (2)	25 (1)	45	20	3
	P_3	3 (ϵ)	12	8 (5)	18 (7)	12
Demand	D_j	4	1	5	7	17

From the initial solution table, we observe that there are only 5 occupied cells, whereas to get the solution it is necessary to have 6 occupied cells. Therefore the given initial solution is degenerate. In order to remove degeneracy, we need an additional allocation to eliminate degeneracy. Therefore, we make an artificial allocation of small quantity ϵ to any of the unoccupied cells. Let us put ϵ to that unoccupied cell which has the minimum transportation cost, viz., cell (P_3, W_1) as shown in the table and now this cell will be considered as the occupied cell. Then the problem can be solved in the usual manner.

Stage 2: Degeneracy Occurs During the Testing of the Optimal Solution:
Degeneracy may occur when we are testing the optimal solution. Degeneracy can be resolved by allocating ϵ to one or more of the recently vacated cells to complete the required (m+n-1) number of occupied cells. The problem is then solved in the usual manner. To illustrate the use of ϵ during the testing of the optimal solution, let us consider the following example:

		Warehouse			Supply
		W_1	W_2	W_3	S_i
Plant	P_1	7	3	6	5
	P_2	4	6	8	10
	P_3	5	8	4	7
	P_4	8	4	3	3
Demand	D_j	5	8	10	23/25

In the given example, since the total supply (25) is more than the total demand (23), therefore introduce a dummy warehouse W_4 with 25 - 23 = 2 items as its demand.

Using Vogel's method, the initial solution to the given problem is obtained as shown in Table 10.

Table 10

		Warehouse				Supply
		W_1	W_2	W_3	W_4	S_i
Plant	P_1	7	3 (5)	6	0	5
	P_2	4 (5)	6 (3)	8 (2)	0	10
	P_3	5	8	4 (5)	0 (2)	7
	P_4	8	4	3 (3)	0	3
Demand	D	5	8	10	2	25

The intial solution is non-degenerate and the transportation cost with the initial solution is:

Total cost = 5 x 3 + 5 x 4 + 3 x 6 + 2 x 8 + 5 x 4 + 2 x 0 + 3 x 3 = Rs. 98

Determine $U_{i's}$ rows and $V_{j's}$ for columns as usual by using the relationship

$C_{ij} = U_i + V_j$ for all occupied cells.

To find out whether an allocation in an unoccupied cell will increase or decrease the cost of transportation, let us calculate for each unoccupied cell the net cost change by using the relationship

Cost change = $C_{ij} - U_i - V_j$

These cost changes are shown in the right hand side bottom of the unoccupied cells in Table 11.

Table 11

		Warehouse			Dummy	Supply	
		W_1	W_2	W_3	W_4	S_i	U_i
	P_1	[7] +6	[3] (5)	[6] +1	[0] -1	5	$U_1 = -3$
	P_2	[4] (5)	[6] (3)	[8] (2)	[0] -4	10	$U_2 = 0$
Plant	P_3	[5] +5	[8] +6	[4] (5)	[0] (2)	7	$U_3 = -4$
	P_4	[8] +9	[4] +3	[3] (3)	[0] +1	3	$U_4 = -5$
Dummy	D_j	5	8	10	2	25	
	V_j	$V_1 = 4$	$V_2 = 6$	$V_3 = 8$	$V_4 = 4$		

Since the cell (P_2, W_4) has the largest negative (-4) value of cost change, therefore, introduce this cell in the new solution. Tracing the closed path, we find that 2 units from cell (P_2, W_3) or (P_3, W_4) should be transported to the cell (P_2, W_4). We observe that both the occupied cells, viz., (P_2, W_3) and (P_3, W_4) become unoccupied cells. Thus the total number of occupied cells are less than the required number (m + n - 1) = (4 + 4 - 1) = 7. This means degeneracy has developed during the test of optimality. To remove this we allocate $\in$ to any of the recently vacated cell with least transportation cost. Because cell (P_3, W_4) has least transportation cost, we allocate $\in$ to this cell as shown in Table 12.

Table 12

		Warehouse				Supply
		W_1	W_2	W_3	W_4	S_i
	P_1	[7]	[3] (5)	[6]	[0]	5
	P_2	[4] (5)	[6] (3)	[8]	[0] (2)	10
Plant	P_3	[5]	[8]	[4] (7)	[0] ($\in$)	7
	P_4	[8]	[4]	[3] (3)	[0]	3
Demand	D_j	5	8	10	2	25

Again to test the solution for optimality, we calculate U_i and V_j numbers and value of cost change for each unoccupied cell as shown in Table 13.

Table 13

		Warehouse				Supply	
		W_1	W_2	W_3	W_4	S_i	U_i
Plant	P_1	7	3 (5)	6	0	5	$U_1 = -3$
	P_2	4 (5)	6 (3)	8	0 (2)	10	$U_2 = 0$
	P_3	5	8	4 (7)	0 (ϵ)	7	$U_3 = 0$
	P_4	8	4	3 (3)	0	3	$U_4 = -1$
Demand	D_j	5	8	10	2	25	
	V_j	$V_1 = 4$	$V_2 = 6$	$V_3 = 4$	$V_4 = 0$		

As the value of cost change in all the unoccupied cells is positive therefore the current solution is optimal.

The total minimum transportation cost is:

Total cost = 5 x 3 + 5 x 4 + 3 x 6 + 2 x 0 + 7 x 4 + 3 x 3 = Rs. 93.

Activity J

For the following transportation problem, determine the optimum distribution plan.

		Warehouse			Supply
		W_1	W_2	W_3	S_i
Plant	P_1	8	5	6	100
	P_2	15	10	12	120
	P_3	3	9	10	80
Demand	D_j	100	100	100	300

. . . .

. . . .

. . . .

. . . .

12.7 MAXIMISATION IN A TRANSPORTATION PROBLEMS

Although the transportation problems we have been dealing with were of minimisation, there may be cases when you have a transportation problem with maximisation objectives. Maximisation problems of the transportation models are quite easy to solve. Several methods are available but we will consider that method which converts the problem from a maximisation to a minimisation problem which can then be solved in the normal manner. The conversion can be done by subtracting each of the profit elements associated with the transportation routes from the largest profit element. The resulting values (elements) so obtained represent opportunity costs because they correspond to the difference in profit earned by that route and the largest profit that could be earned by any of the routes. After converting the profits to opportunity costs, we attempt to minimise these opportunity costs by solving the problem in the same manner as the other minimisation problems. Thus, this approach requires only a simple adjustment to the data of the initial transportation problem.

Prohibited Routes

In certain situations, some of the routes in a transportation are prohibited for some external reason. This occurs whenever one or more routes are not available as shipment options. This can happen for a number of reasons, such as bad weather conditions, weight or size conditions, transportation strikes etc. Such restrictions or prohibitions can be handled in the transportation problem by assigning a very high cost to the prohibited routes. This ensures that these routes will not be included in the optimal solution.

12.8 SUMMARY

In this unit we have considered allocation models in which transportation models have been highlighted. The transportation problem is one of the most frequently encountered applications in real life situations. Solution of the transportation problem facilitates the specification of how many units should be shipped (transported) from each origin to each destination in order to satisfy all destination demands while minimising the total costs of transportation. The special cases in transportation problems have also been discussed.

12.9 KEY WORDS

Balanced Transportation Problem: A Transportation Problem in which the total supply available at the origins exactly satisfies the total demand required at the destinations.

Degenerate Solution: A feasible solution to a transportation problem is said to be degenerate if the number of occupied cells is less than (m + n - 1).

MODI Method: A procedure for the solution of the transportation problem.

North West Corner Method: Systematic allocation to cells beginning in the upper left hand corner of the transportation table to obtain an initial feasible solution.

Stepping Stone Method: A procedure for improving the cost in the transportation method.

Transportation Problem: A special type of linear programming problem that involves the transportation or physical distribution of goods and services from several supply origins to several demand destinations.

Unbalanced Transportation Problem: A transportation problem in which total supply does not equal total demand.

Vegel's Method: A procedure used to obtain an initial feasible solution to the transportation problem.

12.10 SELF-ASSESSMENT EXERCISES

1 What do you understand by transportation problems? Describe any two methods of determining the initial feasible solution for a transportation problem.

2 Discuss the similarities and differences between the Stepping Stone Method and the MODI method in solving a transportation problems.

3 How does the problem of degeneracy arise in a transportation problem? How does one overcome it?

4 A company has three warehouses A,B and C and four stores W, X, Y and Z. The warehouses have altogether a surplus of 150 units of a given commodity as follows:

A	B	C
50	60	40

The four stores need the following amounts.

W	X	Y	Z
20	70	50	10

Cost (in rupees) of shipping one unit of commodity from various warehouses to different stores is as follows:

		Store			
		W	X	Y	Z
Warehouse	A	50	150	70	60
	B	80	70	90	10
	C	15	87	79	81

i) Formulate the above problem using Linear Programming model.
ii) Work out the transportation schedule by using Vogel's method.
iii) Find the associated transportation cost.

5 A company has factories at F_1, F_2 and F_3 which supply warehouses at W_1, W_2 and W_3. Weekly factory capacities are 200, 160 and 90 units, respectively. Weekly warehouse demand are 180, 120 and 150 units, respectively. Unit shipping costs (in rupees) are as follows:

		Warehouses		
		W_1	W_2	W_3
Factories	F_1	16	20	12
	F_2	14	8	18
	F_3	26	24	16

Determine the optimal distribution for this company to maximise shipping costs.

6 Solve the following transporation problem whose cost matrix, availability at each plant and requirements at each warehouse are given as follows:

		Warehouses				
		W_1	W_2	W_3	W_4	Availability
	P_1	190	300	500	100	70
Plant	P_2	700	300	400	600	90
	P_3	400	100	600	200	180
Requirement		50	80	70	140	

7 A company has four factories F_1, F_2, F_3 and F_4 from which it ships its product units to four warehouses W_1, W_2, W_3 and W_4 which are the distribution centres. Transportation costs per unit between various combinations of factories and warehouses are as given below:

		Warehouses				
		W_1	W_2	W_3	W_4	Availability
	F_1	48	60	56	58	140
Factory	F_2	45	55	53	60	260
	F_3	50	65	60	62	360
	F_4	52	64	55	61	220
Requirement		200	320	250	210	

Find the transportation schedule which minimises the distribution cost.

8 A company manufacturing air coolers has two plants located at Bombay and Calcutta with a weekly capacity of 200 units and 100 units, respectively. The company supplies air coolers to its 4 show rooms situated at Ranchi, Delhi, Lucknow and Kanpur which have a demand of 75,100,100 and 30 units, respectively. The cost per unit (in Rs.) is shown in the following table:

	Ranchi	Delhi	Lucknow	Kanpur
Bombay	90	90	100	100
Calcutta	50	70	130	85

Plan the production programme so as to minimise the total cost of transportation.

9 A manufacturer wants to ship 8 loads of his product as shown below. The matrix gives the kilometers from origin to the destination.

		Destination			
		A	B	C	Availability
	X	50	30	220	1
Origin	Y	90	45	170	3
	Z	50	200	50	4
Requirement		3	3	2	

Shipping costs are Rs. 10 per load per kilometer. What shipping schedule should be used?

10 A company has three factores F_1, F_2 and F_3 which supply warehouses at W_1, W_2 and W_3. Monthly factory capacities are 120 units, 80 units and 80 units, respectively. Monthly warehouse requirements are 150 units, 80 units and 50 units, respectively. Unit shipping costs (in Rs.) are as follow:

		Warehouses			
		W_1	W_2	W_3	Availability
Factory	F_1	8	8	15	120
	F_2	15	10	17	80
	F_3	3	9	10	80
Requirement		150	80	50	

Determine the optimum distribution for this company to minimise shipping cost.

12.11 FURTHER READINGS

Cook, T.M. and R.A. Russell, 1981. *Introduction to Management Science,* Prentice Hall International, Inc.: Englewood Cliffs.

Cook, W.P. 1985. *Quantitative Methods for Management Decisions,* McGraw-HillBook Company: New York.

Dunn, R.A. and K.D. Ramasingh, 1981. *Management Science,* Macmillan Publishing Co. Inc: Bombay.

Gupta, M.P. and J.K. Sharma, 1987, *Operations Research for Management,* National Publishing House: New Delhi.

Loomba, N.P. 1978. *Management-A Quantitative Perspective,* Macmillan Publishing Co. Inc.: New York.

Turban, E and J.R. Meredith, 1985. *Fundamentals of Management Science,* Business Publications, Inc.: New York.

UNIT 13 ASSIGNMENT MODELS

Objectives

After studying this unit, you should be able to:

- recognise an assignment problem
- convert an assignment problem into a transportation problem
- solve assignment problems by using Hungarian method
- describe special cases in assignment problems such as maximisation case, multiple optimal solution, unbalanced assignment problem, and prohibited assignments.

Structure

13.1 INTRODUCTION

In this unit, we introduce a special type of linear programming problem known as assignment problem. The assignment problem deals in allocating the various items (resources) to various receivers (activities) on a one to one basis in such a way that the resultant effectiveness is optimised. These types of problems are linear programming applications that we can solve using the simplex method. Assignment problems can also be solved by transportation method. But simpler and more efficient methods such as Hungarian method have been developed for getting the solution of assignment problems.

13.2 ASSIGNMENT MODELS

The assignment problem is a special case of the transportation problem. In assignment problems, supply in each row represents the availability of a resource such as man, vehicle, product, salesman, etc. and demand in each column represents different activities to be performed, such as jobs, routes, factories, areas, etc., for each of which only one man or vehicle or product or salesman respectively is required. Entries in the square being costs, times, or distances. The essential characteristic of the assignment problem is: n resources are to be assigned to n activities such that each resource is allocated to each activity and each activity is performed by one resource only. The allocation is to be done in such a way so as to maximise the resultant affectiveness.

A slightly different tabular form than that of transportation problems will be convenient for handling assignment problems. To show that an assignment problem is a special case of the transportation problem, consider an instance where three qualified men are available to perform three jobs. Because of individual's training and experience, the cost of successful completion of the given job is different for each man. Find the assignment which optimises the performance characteristics of men with minimum cost of assignment. It is assumed that each job is completed independently by one man. The unit cost (in rupees) to complete each job by each man is given in the following Table:

		Job J_1	J_2	J_3	Supply
Plant	M_1	4	8	10	1
	M_2	6	9	7	1
	M_3	11	12	5	1
	Demand	1	1	1	3

The format of the above table is the same as that of a transportation problem, except for each row, the total supply is 1 and for each column, the total demand is 1. The objective is to optimise the total performance characteristics by assigning three men to three jobs. Let X_{ij} represent assignment of jth man to ith job. Then the problem can be stated mathematically in the linear programming form as follows:

Minimise $Z=4X_{11} + 8X_{12} + 10X_{13} + 6X_{21} + 9X_{22} + 7X_{23} + 11X_{31} + 12X_{32} + 5X_{33}$

Subject to the following constraints

i) Each man must be assigned to one and only one job, i.e.,

$X_{11} + X_{12} + X_{13} = 1$ for man M_1

$X_{21} + X_{22} + X_{23} = 1$ for man M_2

$X_{31} + X_{32} + X_{33} = 1$ for man M_3

or $\sum_{j=1}^{3} X_{ij} - 1$ for i = 1,2,3

ii) Each job must be assigned to one and only one man, i.e.,

$X_{11} + X_{21} + X_{31} = 1$ for job J_1

$X_{12} + X_{22} + X_{32} = 1$ for job J_2

$X_{13} + X_{23} + X_{33} = 1$ for job J_3

or $\sum_{j=1}^{3} X_{ij} - 1$ for j = 1,2,3

iii) X_{ij} = assignment of ith man to jth job

= 1, if the ith man is assigned to jth job

= 0, if the ith man is not assigned to jth job.

This problem is a special case of the transportation problem, where each $a_i = 1$ and $b_j = 1$. Thus, every assignment problem can be solved by the transportation method but solution by assignment method will be more efficient and requires less computation time.

Every basic feasible solution of a general assignment problem having a square pay-off matrix (say n x n) should be (n + n - 1) assignment cells. But due to the special structure of the assignment problem, any solution of such a problem cannot have more than n assignment cells. Thus, the assignment problem is inherently 'degenerate'. If the given pay-off matrix is not a square matrix, we can convert it to a square matrix by adding dummy row or column. This concept of unbalanced assignment problem will be discussed later.

Activity A

Explain the difference between a transportation problem and an assignment problem.

....
....
....
....
....

Activity B

Explain the conceptual justification that an assignment problem can be considered as a linear programming problem.

....
....
....
....
....

13.3 HUNGARIAN METHOD OF ASSIGNMENT PROBLEM (MINIMISATION CASE)

An efficient method for solving an assignment problem, known as Hungarian method of assignment is based upon the following two important properties:

1 In an assignment problem, if a constant quantity is added or subtracted from every element of any row or column in the given cost matrix, an assignment that minimises the total cost in one matrix also minimise the total cost in the other.

2 In an assignment problem, a solution having zero total cost is an optimum solution. Hungarian method of assignment problem (minimisation case) can be summarised in the following steps:

Step 1: In the given matrix, subtract the smallest element in each row from every element of that row.

Step 2: In the reduced matrix obtained from Step 1, subtract the smallest element in each column from every element of that column.

Step 3: Make the assignment for the reduced matrix obtained from steps 1 and 2 in the following way:

a) Examine the rows successively until a row with exactly one zero is found. Make an assignment to this single zero by putting square □ around it and cross out (X) all other zeros appearing in the corresponding column as they will not be used to make any other assignment in that column. Proceed in this manner until all rows have been examined.

b) Examine the columns successively until a column with exactly one zero is found. Make an assignment to this single zero by putting square □ around it and cross out (X) all other zeros appearing in the corresponding row as they will not be used to make any other assignment in that row. Proceed in this manner until all columns have been examined.

c) Repeat steps 3(a) and 3(b) until all zeros in rows/columns are either marked or crossed out (X). If the number of assignments (Marked □) made are equal to the number of rows/columns (being square matrix), then it is an optimal solution. Otherwise go to step 4.

Step 4: Draw the minimum number of horizontal and vertical lines to cover all zeros in the reduced matrix obtained from step 3 in the following way:

a) Mark (✔) all rows that do not have assignments
b) Mark (✔) all columns that have zeros in marked rows [step 4 (a)]
c) Mark (✔) all rows that have assignments in marked columns [step 4 (b)]
d) Repeat steps 4 (a) to 4 (c) until no more rows or columns can be marked.
e) Draw straight lines through all unmarked rows and marked columns.

It may be pointed out here that you may also draw the minimum of lines to cover all zeros by inspection.

Step 5: If the number of lines drawn [step 4 (c)] are equal to the number of rows or columns, then it is an optimal solution, otherwise go to step 6.

Step 6: Select the smallest element among all the uncovered elements. Subtract this smallest element from all the uncovered demands and add it to the element which lies at the intersection of two lines. Thus, we obtain another reduced matrix for fresh assignments.

Step 7: Go to step 3 and repeat the procedure until the number of assignments become equal to the number of rows or columns. In such a case, we shall observe that every row/column has an assignment. Thus, the current solution is the optimal solution.

To illustrate the application of Hungarian method of assignment problem, let us consider an example where a company is faced with the problem of assigning five jobs to five machines. Each job must be done on only one machine. The cost (in Rs.) of processing each job on each machine is given below in Table 1:

Table 1

Machine

Job	M_1	M_2	M_3	M_4	M_5
J_1	7	5	9	8	11
J_2	9	12	7	11	10
J_3	8	5	4	6	9
J_4	7	3	6	9	5
J_5	4	6	7	5	11

The problem is to determine the assignment of jobs to machines so that it will result in minimum total cost.

As suggested in the solution procedure, we select the minimum element in each row and subtract this element from every element in that row. The resultant reduced matrix is shown below in Table 2:

Table 2

Machine

Job	M_1	M_2	M_3	M_4	M_5
J_1	2	0	4	3	6
J_2	2	5	0	4	3
J_3	4	1	0	2	5
J_4	4	0	3	6	2
J_5	0	2	3	1	7

Next we select the minimum element in each column and subtract this element from every element in that column. You may note that only colomn M_4 and M_5 will change. The resultant reduced matrix is shown below in Table 3:

Table 3

Machine

Job	M_1	M_2	M_3	M_4	M_5
J_1	2	0	4	2	4
J_2	2	5	0	3	1
J_3	4	1	0	1	3
J_4	4	0	3	5	0
J_5	0	2	3	0	5

Now we attempt to make a complete set of assignments using only a single zero element in each row or column. Since row J_1 contains only single zero, therefore the assignment is made in the cell (J_1, M_2) and the zero appearing in the corresponding column M_2 is crossed out. Similarly the assignment is made in the cell (J_2, M_3) and the zero appearing in the corresponding column M_4 is crossed out. Now row J_4 has only single zero, therefore the assignment is made in cell (J_4, M_5). Since there are two zeros in row J_5, we cannot make assignment in this row J_5. Looking columnwise, we find that column M_1 has only single zero, therefore we make an assignment in cell (J_5, M_1) and cross out the zero appearing in the corresponding row J_5. The assignments so made are shown in Table 4.

Table 4

Machine

Job		M_1	M_2	M_3	M_4	M_5
	J_1	2	[0]	4	2	4
	J_2	2	5	[0]	3	1
	J_3	4	1	~~0~~	1	3
	J_4	4	~~0~~	3	5	[0]
	J_5	[0]	2	3	~~0~~	5

Thus, it is possible to make only four of the five necessary assignments using the zero element position. We therefore create one more zero element by drawing the minimum number of horizontal and vertical lines. Usually the minimum number of lines to cover all of the zeros can be obtained by inspection. However we shall use the method given earlier in explaining the various steps. The various steps for drawing the minimum number of lines are:

a) Mark the row J_3 which has no assignment

b) Mark column M_3 which has zero in the marked row J_3

c) Mark row J_2 which has assignment in marked column M_3

d) Repeat steps (a) and (b) until no more rows or columns can be marked.

e) Draw the lines through unmarked rows and marked columns.

The minimum number of lines drawn are shown in Table 5. You must check that the number of lines drawn is equal to the number of assignments made. But we require five assignments. To create one more zero, we examine the elements not covered by these lines and select the smallest element, viz, I among these uncovered lines.

Table 5

Machine

Job		M_1	M_2	M_3	M_4	M_5
	J_1	2	0	4	2	4
	J_2	2	5	0	3	1
Job	J_3	4	1	0	1	3
	J_4	4	0	3	5	0
	J_5	0	2	3	0	5

Subtract this smallest element 1 from all the uncovered elements and add it to the element where the two lines intersect. The reduced matrix so obtained is shown below in Table 6:

Table 6

Machine

Job		M_1	M_2	M_3	M_4	M_5
	J_1	2	0	5	2	4
	J_2	1	4	0	2	0
Job	J_3	3	0	0	0	2
	J_4	4	0	4	5	0
	J_5	0	2	4	0	5

Now we make fresh assignment. Proceeding in the usual way, the set of assignments made are shown in Table 6. Hence the optimal solution is:

Assign Job	To Machine	Cost (Rs.)
J_1	M_2	5
J_2	M_3	7
J_3	M_4	6
J_4	M_5	5
J_5	M_1	4
	Minimum total cost =	Rs. 27

Activity C

A certain equipment needs seven repair jobs which have to be assigned to seven machines. The estimated time (in hours) that each mechanic requires to complete the repair job is given in the following Table:

Job

Machine		J_1	J_2	J_3	J_4	J_5
	M_1	2	9	2	7	1
	M_2	6	8	7	6	1
Machine	M_3	4	6	5	3	1
	M_4	4	2	7	3	1
	M_5	5	3	9	5	1

Assuming that each mechanic can be assigned to only one job, determine the minimum time assignment.

....

....

....

....

....

....

....

....

13.4 SPECIAL CASES IN ASSIGNMENT PROBLEMS

1. Maximisation Case in Assignment Problem: In some cases, the pay off elements of the assignment problem may represent revenues or profits instead of costs so that the objective will be to maximise the total revenue or profit. The Hungarian method explained earlier can also be used for maximisation case. The problem of maximisation can be converted into a minimisation case by selecting the largest element among all elements of the profit matrix and then subtracting it from all other elements in the matrix. You can then proceed as usual and obtain the optimal solution by adding the orignal values of these cells to which the assignments have been made.

2. Multiple Optimal Solutions: Sometimes, it is possible to have two or more ways to cross out all zero elements in the final reduced matrix for a given problem. This implies that there are more than the required number of independent zero elements. In such cases, there will be multiple optimal solutions with the same total cost of assignment. In such type of situations, management may exercise their judgment or preference and select that set of optimal assignments which is more suited to their requirement.

To demonstrate the applicability of a maximisation case and also that of multiple optimal solutions, consider the example where a company has four sales territories and four salesmen available for assignment. The sales territories are not equally rich in their sales potential and the salesmen also differ in their selling ability. The sales (in thousand rupees) for each salesman to be assigned to each territory is given below in Table 7:

Table 7
Territory

		I	II	III	IV
Salesman	A	42	35	28	21
	B	30	25	20	15
	C	30	25	20	15
	D	24	20	16	12

How should the territories be assigned to salesmen so as to maximise the total sales?

Since the problem involves maximisation of total sales, the problem of maximisation can be converted into a minimisation problem by subtracting all the elements from the highest element (i.e. 42) in the given matrix. The matrix so obtained is shown below in Table 8:

Table 8
Territory

		I	II	III	IV
Salesman	A	0	7	14	21
	B	12	17	22	27
	C	12	17	22	27
	D	18	22	26	30

As this problem is reduced to that of a minimisation problem, we shall therefore apply the Hungarian method of assignment as discussed earlier.

Subtracting the smallest element in each row from every element in that row and then, columnwise, subtracting the smallest element in each column from every element in that column, we get the reduced matrix as shown in Table 9:

Table 9
Territory

		I	II	III	IV
Salesman	A	0	3	6	9
	B	0	1	2	3
	C	0	1	2	3
	D	0	0	0	0

Now make assignments by using single zero elements in each row and column. The number of assignments made are only two as shown in Table 10. In order to create more zeros, we shall draw the minimum number of lines to cover all zeros as explained earlier. Since there are only two assignments, the number of lines drawn are only two as shown in Table 10:

Table 10
Territory

		I	II	III	IV
Salesman	A	[0]	3	6	9
	B	~~0~~	1	2	3
	C	~~0~~	1	2	3
	D	~~0~~	[0]	~~0~~	~~0~~

Selecting the smallest element viz., 1 among all uncovered elements and subtracting it from each uncovered element and adding it to the intersection of two lines, we get the matrix as shown in Table 11.

Table 11
Territory

		I	II	III	IV
Salesman	A	0	2	5	8
	B	0	0	1	2
	C	0	0	1	2
	D	1	0	0	0

Now when we make fresh assignments, we observe that there are only three assignments made and thus it is not an optimal solution. This is shown in Table 12. Again draw the minimum number of lines to cover all zeros in such a way that the number of lines drawn are equal to number of assignments made. This is also shown in Table 12.

Table 12
Territory

		I	II	III	IV
Salesman	A	[0]	2	5	8
	B	~~0~~	[0]	1	2
	C	~~0~~	~~0~~	1	2
	D	1	~~0~~	[0]	~~0~~

Subtracting the smallest element, viz., 1 among all uncovered elements and adding it to the inter section of two lines we get the reduced matrix as shown in Table 13.

Table 13
Territory

		I	II	III	IV
Salesman	A	[0]	2	4	7
	B	~~0~~	0	0	1
	C	~~0~~	0	0	1
	D	2	1	~~0~~	[0]

Now make the assignments in reduced matrix of Table 13. We observe that there is only single zero element in row A and column IV, so we make the assignment by putting☐ and cross out other zeros in column I and row D. for the remaining zeros, we can make the assignments in two ways. First if we make an assignment considering the zero element in row B and column II, then the zero, appearing in column III and row B is to be crossed out and then the other assignment is automatically made in row C and column III. In the other case, if we make an assignment considering the zero element in row B and column III, then the zero appearing in column II and row C is to be crossed out and then the other assignment is made in row C and column II. Thus this problem has two sets of optimal solutions as shown in Table 14 (a) and (b).

Table 14

Territory (a)

Salesman	I	II	III	IV
A	[0]	2	4	7
B	✗	[0]	✗	1
C	✗	✗	[0]	1
D	2	1	✗	[0]

Territory (b)

Salesman	I	II	III	IV
A	[0]	2	4	7
B	✗	✗	[0]	1
C	✗	[0]	✗	1
D	2	1	✗	[0]

The two optimal solutions are

Solution I

Salesman	Territory	Sales (in 000's Rs.)
A	I	42
B	II	25
C	III	20
D	IV	12
	Total	99

Solution 2

Salesman	Territory	Sales (in 000's Rs.)
A	I	42
B	II	20
C	III	25
D	IV	12
	Total	99

Both the solutions show that the best salesman A is assigned to the richest territory I and the worst salesman to the poorest territory IV. The salesman B and C being equally good, may be assigned to either territory II or III. In both the cases, the optimal total sales is Rs. 99,000.

Activity D

Four salesmen have completed the company sales course and are to be assigned to four different sales territories. Based on their experience, course performance, product knowledge, and potential customers, the company has rated each salesman's expected success in each sales territory. The ratings, on a scale from 1 (low) to 10 (maximum) are shown in the following table:

		Territory			
		A	B	C	D
Salesman	S_1	7	9	10	9
	S_2	8	7	9	9
	S_3	7	10	9	8
	S_4	6	8	8	7

If the objective is to maximise total rating, who should be assigned to which sales territory?

Are other optimal solutions possible?

....

....

....

....

....

....

....

....

3. Unbalanced Assignment Probelm: Whenever, the payoff matrix of an assignment problem is not a square matrix (i.e. number of rows are not equal to the number of columns), the assignment problem is called unbalanced assignment problem. In such cases, dummy rows and/ or columns are added in the matrix to make it a square matrix. Then, we can apply the Hungarian method to this resulting balanced (square matrix) assignment problem. For example, if five workers, are to be assigned to six machines, a dummy row is simply added to transform the assignment problem into a square (6x6) matrix. Creating dummy rows or columns will give us a matrix of equal dimensions and allow us to solve the problem as discussed earlier. The cost (or time) associated with this dummy row or column is assigned zero element in the matrix.

To illustrate the application of unbalanced assignment problem, consider an example where a company is faced with the problem of assigning six different machines to five different jobs. The costs (in,-ooo s Rs.) are estimated as shown in Table 15.

Table 15

		Job				
		J_1	J_2	J_3	J_4	J_5
Machine	M_1	6	2	5	2	6
	M_2	2	5	8	7	7
	M_3	7	8	6	9	8
	M_4	6	2	3	4	5
	M_5	9	3	8	9	7
	M_6	4	7	4	6	8

The objective of the company is to make assignments in such a way that the total cost is minimum.

Since the number of machines is not equal to the number of jobs, a dummy job J_6 is created. The cost associated with any machine for this dummy job is zero. This square matrix is shown in Table 16.

Table 16

Machine \ Job	J_1	J_2	J_3	J_4	J_5	J_6
M_1	6	2	5	2	6	0
M_2	2	5	8	7	7	0
M_3	7	8	6	9	8	0
M_4	6	2	3	4	5	0
M_5	9	3	8	9	7	0
M_6	4	7	4	6	8	0

We observe that every row has a zero element, therefore by subtracting the smallest element in each column from every element of that column, we get a reduced matrix as shown in Table 17.

Table 17

Machine \ Job	J_1	J_2	J_3	J_4	J_5	J_6
M_1	4	0	2	0	1	0
M_2	0	3	5	5	2	0
M_3	5	6	3	7	3	0
M_4	2	0	0	2	0	0
M_5	7	1	5	7	2	0
M_6	2	5	1	4	3	0

Now let us make assignments. We can make only four assignments as shown in Table 18. To create more zeros, we draw the minimum number of lines as explained earlier and are shown in Table 18.

Table 18

Machine \ Job	J_1	J_2	J_3	J_4	J_5	J_6
M_1	4	[0]	2	0	1	0
M_2	[0]	3	5	5	2	0
M_3	5	6	3	7	3	[0]
M_4	2	0	[0]	2	0	0
M_5	7	1	5	7	2	0
M_6	2	5	1	4	3	0

Subtracting the smallest uncovered element, viz., 1 from all uncovered elements and adding it to the intersection of two lines, we get the reduced matrix as shown in Table 19.

While making assignments in single zero element, we get six assignments as shown in Table 19. Hence the optimal assignments made are:

Table 19

		Job					
		J_1	J_2	J_3	J_4	J_5	J_6
Machine	M_1	4	~~0~~	2	[0]	1	1
	M_2	[0]	3	5	5	2	1
	M_3	4	5	2	6	2	[0]
	M_4	2	~~0~~	~~0~~	2	[0]	1
	M_5	6	[0]	4	6	1	~~0~~
	M_6	1	4	[0]	3	2	~~0~~

Assign	To	(Cost in 000's Rs.)
M_1	J_4	2
M_2	J_1	2
M_3	J_6	0
M_4	J_5	5
M_5	J_2	3
M_6	J_3	4
	Total =	16

This solution also suggests that machine M_3 will not be assigned to any job. The total minimum cost incurred will be Rs. 16,000.

Activity E

A company is faced with the problem of assigning six operators to five jobs. The time taken (in hours) by operators in completing the job is given below:

		Job				
		J_1	J_2	J_3	J_4	J_5
Operator	O_1	26	22	15	22	16
	O_2	22	25	18	17	17
	O_3	17	18	26	19	18
	O_4	16	22	23	24	25
	O_5	19	23	18	19	17
	O_6	24	17	24	16	28

Determine the assignment of operators to various jobs in such a way that the total time taken in minimised. Which operator will be left unassigned?

....

....

....

....

4. Prohibited Assignments: Sometimes due to certain reason an assignment cannot be made in a particular cell. For Example, a particular machine cannot be installed at a particular place or a worker cannot be given a particular job to perform. To resolve this, we put either a very large cross or dash (--) to avoid assignments in those cells where there is a restriction of assignment.

To explain prohibited assignments, let us consider a case where three workers are available to work with the machine and the respective cost (in Rs.) associated with each worker assignment is shown in Table 20. Further due to certain reasons, worker W_1 and worker W_2 cannot be assigned to machine M_4 and machine M_3 respectively.

Table 20
Machine

		M_1	M_2	M_3	M_4
Worker	W_1	12	3	6	—
	W_2	4	11	—	5
	W_3	8	2	10	9

Determine the optimal assignments and associated cost.

Add one dummy worker (row) to convert the given problem into a balanced one. Write zero assignment cost in this row. This is shown below in Table 21.

Table 21
Machine

		M_1	M_2	M_3	M_4
Worker	W_1	12	3	6	—
	W_2	4	11	—	5
	W_3	8	2	10	9
	Dummy worker	0	0	0	0

The problem has been converted into a balanced transportation problem, therefore now we may proceed in the usual manner to solve it.

Activity F

Five workers are available to work on six machines and the respective cost (in Rs.) associated with each worker on different machines is given below. The dash (-) in the matrix refers to those workers who cannot be assigned a particular machine.

Machine

Worker		M_1	M_2	M_3	M_4	M_5	M_6
	W_1	12	3	6	—	5	9
	W_2	4	11	—	5	—	3
Worker	W_3	8	2	10	9	7	5
	W_4	—	7	8	6	12	10
	W_5	5	8	9	4	6	—

Determine the optimal assignments and its associated costs. Which machine will be unassigned?

....

....

....

....

13.5 SUMMARY

In this unit the application of assignment models have been highlighted. The assignment problem is a special type of transportation problem in which allocations are made on one to one basis. Assignment of jobs to machines, workers to various tasks, salesman to territories are some of the examples of real life situations. Solution of assignment problem enables us to determine the one to one assignment of resources to activities which optimises the appropriate objective function.

13.6 KEY WORDS

Assignment Problem: The problem of assigning n resources to n activities in such a way that only one resource is assigned to one activity, every activity has a resource assigned to it, and the cost of completing all jobs is minimised.

Balanced Assignment Problem: An assignment problem in which the number of activities is exactly equal to the number of resources.

Hungarian Method: An efficient solution procedure for the assignment problem that was developed by a Hungarian mathematician.

Unbalanced Assignment Problem: An assignment problem where the number of resources is not equal to the number of activities.

13.7 SELF-ASSESSMENT EXERCISES

1 What is an assignment problem? Discuss its method of solution. Give two applications in management.

2 Explain the difference between a transportation problem and an assignment problem.

3 In a textile sales emporium, four salesmen A, B, C and D are available to handle four

counters W, X, Y and Z. Each salesman can handle any counter. The service time (in hours) of each counter when manned by each salesman is given below:

Counters / **Salesmen**

	A	B	C	D
W	41	72	39	52
X	22	29	49	65
Y	27	39	60	51
Z	45	50	48	52

How should the salesmen be assigned to counters so as to minimise the total service time? Each salesman should handle only one counter.

4 A computer centre has three expert programmers. The centre needs three 'application' programmes to be developed. The head of the computer centre, after carefully studying the programmes to be developed, estimates the computer time (in minutes) required by the experts for the application of programmes as follows:

Programmer / **Programme**

	A	B	C
1	120	100	80
2	80	90	110
3	110	140	120

Assign the programmers to the programme in such a way that the total computer time is least.

5 Five salesmen are to be assigned to five territories. Based on past performance, the following table shows the annual sales (in thousand rupees) that can be generated by each salesman in each territory.

Salesman / **Territory**

	T_1	T_2	T_3	T_4	T_5
S_1	26	14	10	12	9
S_2	31	27	30	14	16
S_3	15	18	16	25	30
S_4	17	12	21	30	25
S_5	20	19	25	16	10

Find the optimal assignment to maximise sales.

6 Six salesmen are to be allocated to six sales regions so that the cost of allocation of the job will be minimum. Each salesman is capable of doing the job at different costs in each region. The cost (in rupees) matrix is given below:

Salesman

Region		I	II	III	IV	V	VI
	A	15	35	0	25	10	45
	B	40	5	45	20	15	20
	C	25	60	10	65	25	10
	D	25	20	35	10	25	60
	E	30	70	40	5	40	50
	F	10	25	30	40	50	15

i) Find the allocation to give minimum cost. What is the minimum cost?

ii) If the figures given in the above table represent the earning of each salesman at each region, then find an allocation so that the earning will be maximum. Also work out this maximum possible earning.

7 The owner of a small machine shop has four machinists available to assign to jobs for the day. Five jobs are offered with expected profit (in Rs.) for each machinist on each job as follows:

Job

Machinist		A	B	C	D	E
	1	62	78	50	101	82
	2	71	84	61	73	59
	3	87	92	111	71	81
	4	48	64	87	77	80

Determine the assignment of machinists to jobs that will result in a maximum profit. Which job should be declined?

8 A company is faced with the problem of assigning six different machines to five different jobs. The costs are estimated as follows (hundreds of rupees):

Jobs

Machines		1	2	3	4	5
	1	2.5	5	1	6	1
	2	2	5	1.5	7	3
	3	3	6.5	2	8	3
	4	3.5	7	2	9	4.5
	5	4	7	3	9	6
	6	6	9	5	10	6

Solve the problem assuming that the objective is to minimise total cost.

9 At the end of the cycle of schedules, a trucking firm has a surplus of one vehicle in each of the cities 1, 2, 3, 4 and 5 and a deficit of one vehicle in each of the cities A, B, C, D, E and F. The costs (in rupees) of transportation and handling between the cities with a surplus and the cities with a deficit are shown in the following table:

		To City					
		A	B	C	D	E	F
From City	1	134	116	167	233	194	97
	2	114	195	260	166	178	130
	3	129	117	48	94	66	101
	4	71	156	92	143	114	136
	5	97	134	125	83	142	118

Find the assignment of surplus vehicles to deficit cities that will result in a minimum total cost. Which city will not receive a vehicle?

10 A firm wants to purchase three different types of equipment and five manufacturers have come forward to supply one or all the three machines. However, the firm's policy is not to accept more than one machine from any of the manufacturers. The data relating to the price (in lakhs of rupees) quoted by the different manufacturers are given below:

		Machines		
		1	2	3
Manufacturers	A	2.99	3.11	2.68
	B	2.78	2.87	2.57
	C	2.92	3.05	2.80
	D	2.82	3.10	2.74
	E	3.11	2.90	2.64

Determine how best the firm can purchase three machines.

13.8 FURTHER READINGS

Cook, T.M. and R.A. Russell, 1981. *Introduction to Management Science*, Prentice-Hall International, Inc.: Englewood-Cliffs.

Cooke, W.P. 1985. *Quantitative Methods for Management Decisions*, McGraw-Hill Book Company: New York.

Dunn, R.A. and K.D. Ramsingh, 1981. *Management Science*, MacMillan Publishing Co. Inc.: London

Gupta, M.P. and J.K. Sharma. 1987. *Operations Research for Management*, National Publishing House.: New Delhi.

Loomba, N.P. 1978. *Management-A Quantitative Perspective*, MacMillan Publishing Co., Inc.: London.

Turban, E and J.R. Meredith. 1985. *Fundamentals of Management Science*, Business Publications, Inc.: New York.

UNIT 14 *PERT/CPM*

Objectives

After studying this unit, you should be able to:

- appreciate problems involved in planning, scheduling and controlling projects
- list and discuss the special terms developed for this unit, namely: activity, event, dummy activity, critical activity, slack, critical path and float.
- develop simple network diagrams with acitivities and events
- identify critical path through the calculation of the earliest expected time and the latest allowable time
- compute slack and float
- estimate the probability of completing a project by a certain target date.

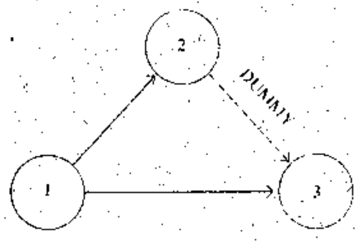
Structure

14.1 INTRODUCTION

Program Evaluation and Review Technique or PERT and Critical path Method or CPM are two of the most widely used techniques in project management. The objectives of project management can be described in terms of a successful project which has been finished on time, within the budgeted cost and to technical specifications which satisfy the end users. A project is any human undertaking with a clear beginning and a clear ending. Planning, scheduling and controlling the work during any worth-while project is the main task for any project manager. Project planning calls for detailing the project into activities, estimating resource requirements and time for each activity, and describing activity inter-relationships. Scheduling requires the details of starting and completion dates for each activity. Control requires not only current status information but insight into possible trade-offs when difficulties arise. Normally for any project, we may be interested in answering questions such as

The above network diagram is incorrect because it breaks the rule of assigning unique numbers to each activity for the purpose of identification. The following network diagram demonstrates the principle of using a dummy activity for overcoming the problem of parallel activities with identical start and finish events.

Figure 1



Therefore, a dummy activity is created to make activities with common starting and finishing events distinguishable, and also to identify and maintain the proper precedence relationship between activities.

14.3 GUIDELINES FOR CONSTRUCTION NETWORK DIAGRAM

1 Each activity is represented by one and only one arrow in the network. Therefore, no single activity can be represented twice in the network.

2 No two activities can be identified by the same beginning and end events. In such cases, a dummy activity is introduced to resolve the problem.

3 Two events are numbered in such a way that the event of higher number can happen only after the event of a lower number is completed.

4 Dangling must be avoided in a network diagram. This happens when precedence and inter-relationships of the activities are not properly identified.

5 To ensure the correct logical sequence and inter-relationships, one has to answer the following questions satisfactorily.

   i) Which activities precede this activity? This is, what other activities must be completed before this activity can be started?

   ii) Which activities follow this activity? Or, what activities cannot be started until this activity is completed?

   iii) Which activities can take place concurrently with this activity? Or, what activities can be worked on at the same time when this one is being performed?

14.4 DETERMINISTIC TIME ESTIMATES

The main determinant of the way PERT and CPM networks are analysed and interpreted is whether activity time estimates are deterministic or probabilistic. If time estimates can be made with a high degree of confidence so that actual time will not differ significantly from estimates, we say the time estimates are **deterministic.** On the other hand if estimated times are subject to variation, we say the time estimates are probabilistic. First, we shall describe the analysis of network with deterministic time estimates and at a later stage with probabilistic time estimates.

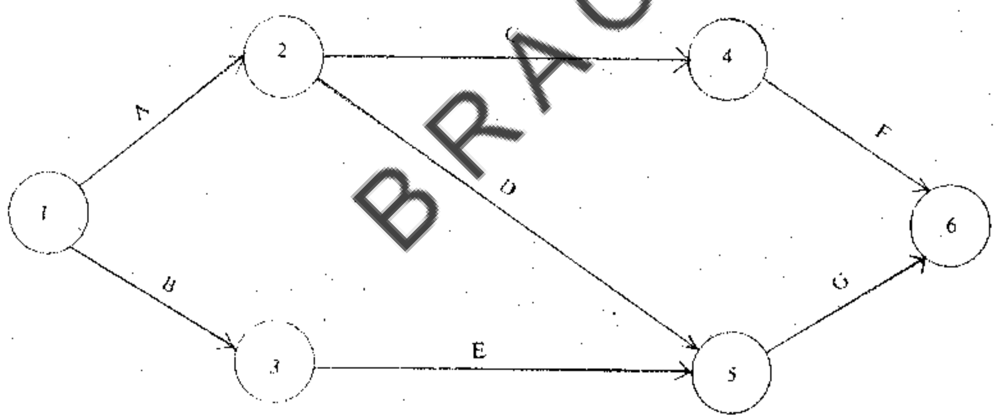
14.5 DEVELOPING A PROJECT NETWORK

Network of PERT/CPM consist of two basic elements: activities and events. The network clearly shows the sequence and inter-relationships of all activities in the project. To illustrate how a project network can be developed, let us consider an example where a project involves several activities which are listed in the following table along with their predecessor activities:

Table 1

Activity	Predecessor Activity
A	—
B	—
C	A
D	A
E	B
F	C
G	D, E

Figure II



All the activities associated with the project can be combined into an integrated network of events and activities as shown in the following network diagram.

The above network diagram gives the complete description of the project. For example activities A and B have no predecessor activities, they can begin immediately and are shown coming out at the start event 1. You may observe that acitivity G has two predecessor activities D and E. Similarly other relationships are also satisfied.

Activity A

A car manufacturing company has decided to redesign its fuel pump for their new car model. This project involves several activities which are listed in the following table. First activity is that the engineering department must finish the design of fuel pump. Second, the marketing department must develop the marketing strategy for its promotion. Third, a new manufacturing process must be designed. Fourth, advertising media must be selected. Fifth, an initial production run must be completed. Finally, the fuel pump must be released to the market.

Activity	Description of activity	Predecessor Activity	Time estimate (weeks)
A	Finish the design of fuel pump	—	5
B	Develop marketing strategy	A	4
C	Design manufacturing process	A	7
D	Select advertising media	B	8
E	Initial production run	C	9
F	Release fuel pump to market	D, E	4

Draw a network diagram for the given project.

14.6 PROJECT DURATION AND CRITICAL PATH

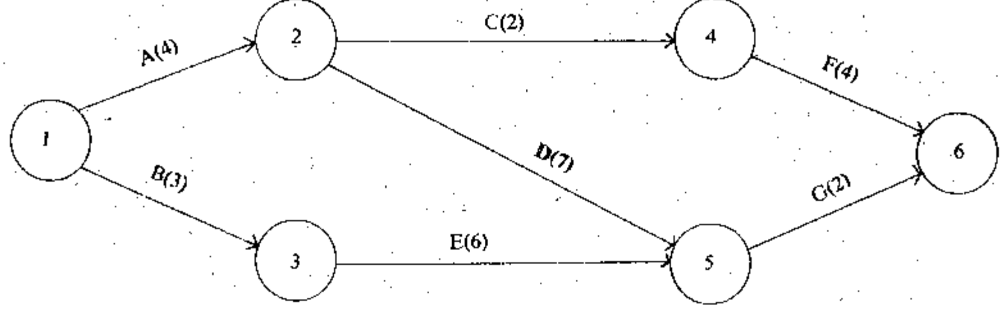
The longest path in the network is called the **critical path.** Indentifying the critical path is of great importance as it determines the duration of the entire project. If any activity on the critical path is delayed, then the entire project will be delayed. Every network has a critical path. It is possible to have multiple critical paths if there are ties among the longest paths. For finding the project duration and critical path, let us consider the example discussed earlier (Table 1). For this example, the time estimates (in weeks) for each activity are as shown in Table 2.

Table 2

Activity	Predecessor activity	Time estimates (weeks)
A	—	4
B	—	3
C	A	2
D	A	7
E	B	6
F	C	4
G	D, E	2

In this network diagram, the time estimates (weeks) are specified inside the bracket along with the activity as shown below:

Figure III



There are three possible paths for this network. For this simple network, the critical path is found by enumerating all of the possible paths to the completion time. These paths are listed below in Table 3:

Table 3

Path	Length of time
i) A→C→F	4+2+4 = 10 weeks
ii) A→D→G	4+7+2 = 13 weeks
iii) B→E→G	3+6+2 = 11 weeks

The second path (A→D →G) is the critical path because it requires the longest period of time i.e. 13 weeks for completion of the project. For this network, the project duration time to complete the project is 13 weeks. The activities on the critical path are called critical activities because a delay in any of these results in a delay of the entire project. In other words, there is no slack time in the activities on the critical path. Slack time is defined as the latest time an activity can be completed without delaying the project minus the earliest time the activity can be completed. Therefore slack time is the amount of time an activity can be delayed without delaying the entire project.

For this small network, it is a simple process to identify the critical path by comparing all possible paths. As the number of activities increases, it may become very difficult and time consuming to find the critical path by complete enumeration or inspection. Therefore we need to develop an algorithm (a systematic approach) to determine the critical path. The critical path calculations proceed in two phases. The first phase or forward pass begins from left to right through the network. The calcuation begins at the start event and moves towards the end event of the project network. The second phase or backward pass begins from right to left through the network. In this phase, the calculation begins from the end event and moves backward to the start event.

14.7 FORWARD PASS (EARLIEST EXPECTED TIME)

In forward pass, we compute the earliest time an event can be expected to occur which in turn depends upon the latest completion time of an activity terminating at that point. Thus, the longest path, in terms of duration times, is the earliest expected time for that event to occur. During the forward calculations, we need to compute the earliest expected time (ET) for each of the events which can be calculated as follows:

$$ET_j = Max\ (ET_i + d_{ij})$$

where

ET_j = the earliest expected time of event j.

ET_i = the earliest expected time that an activity can be started leading to event j.

d_{ij} = duration time of an activity from event i to event j.

Max = Maximum (of)

Let us use this computational procedure to determine the earliest expected time for each event for the network diagram shown in figure III. The use of this procedure suggests that the earliest expected time for a given event is primarily a function of the previous events plus the activity times of all prior activities. Applying this procedure, we get

$ET_1 = 0$ (starting event set at zero)

$ET_2 = ET_1 + d_{12} = 0 + 4 = 4$

$ET_3 = ET_1 + d_{13} = 0 + 3 = 3$

$ET_4 = ET_2 + d_{24} = 4 + 2 = 6$

$ET_5 = Max\ [ET_2 + d_{25}, ET_3 + d_{35}] = Max\ [4 + 7, 3 + 6]$

$= Max\ [11,9] = 11]$

$ET_6 = Max\ [ET_4 + d_{46}, ET_5 + d_{56}] = Max\ [6 +4, 11 + 2]$

$= Max\ 10,13 = 13$

Note that the end event 6 occurs at the end of 13 weeks.

14.8 BACKWARD PASS (LATEST ALLOWABLE TIME)

In Backward Pass we compute the latest allowable time (LT). The LT for an event is the latest time that the event can be delayed without delaying the completion of the entire project. The procedure we use in computing LT is to start from the end event of the network and proceeding backward to the starting event. The latest allowable completion time (LT) for a given event is calculated by subtracting the duration times of all activities coming into the event. In cases where two or more activities start from an event, we must select the smaller of LT values. The LT value for an event in a network can be calculated as follows:

$$LT_i = Min\ (LT_j - d_{ij})$$

where

LT_i = the latest allowable time of event i.

LT_j = the latest allowable time of event j toward which activity (i, j) is headed

d_{ij} = duration time of an activity from event i to event j.

Min = Minimum (of)

Let us use this procedure to determine the LT value for each event in the network shown in Figure III. To find LT values we begin at the end event of the network. The expected completion time for the project is 13 weeks.

In our example, since event 6 is project completion, it must occur no later than 13 weeks or else the project will be completed later than expected. Therefore for the end event, ET = LT = 13 weeks. The computation of other LT values are shown below:

$LT_6 = 13$

$LT_5 = LT_6 - d_{56} = 13 - 2 = 11$

$LT_4 = LT_6 - d_{46} = 13 - 4 = 9$

$LT_3 = LT_5 - d_{35} = 11 - 6 = 5$

$LT_2 = \text{Min}\ [LT_4 - d_{24}, LT_5 - d_{25}]$

$= \text{Min}\ [9-2, 11 - 7] = \text{Min}\ [7, 4] = 4$

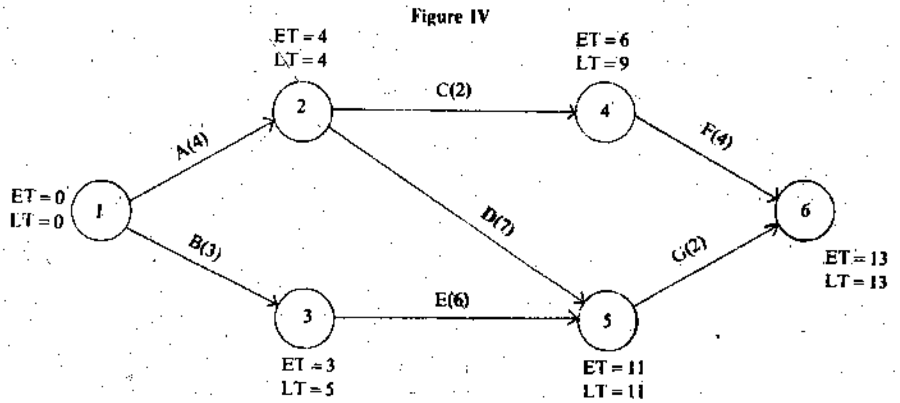
$LT_1 = \text{Min}\ [LT_2 - d_{12}, LT_3 - d_{13}]$

$= \text{Min}\ [4-4, 5 - 3] = \text{Min}\ [0, 2] = 0$

It should be noted here that, by definition, at the beginning point of the network (i.e. event 1), we must have $ET_1 = LT_1 = 0$.

Once the values of ET and LT for all the events are determined, we can easily identify the critical path. These values of ET and LT are listed around each event in Figure IV. If the values of ET and LT of an event are equal, then such an event is referred to as the critical event. If the values of ET and LT of an event are not equal, then such an event is referred to as noncritical events.

Figure IV



Critical activities can also be identified from the project network diagram. A critical activity is an activity which joins two critical events and has a duration which equals the difference between the times of these critical events. A critical path consists only of such activities. It may be pointed out again that critical activities are important because if they exceed their estimated durations, the whole project will be delayed to that extent.

An event that is not critical is said to have slack. **Slack** is the calculated time span within which the event must occur. The term slack is used only for referring to events.

As you would expect, every event on the critical path has no slack time. The critical path is shown by thick lines in Figure IV. The importance of identifying the critical path is that it points out those activities and events which are critical and, as such, must be carefully monitored and controlled.

Activity B

Consider the data of activity A. Compute the earliest expected time and latest allowable time for the events in the given project. Also determine the critical path and slack time. Interpret your slack time values.

....

....

....

....

....

....

....

....

14.9 FLOAT

The concept of float is of great importance for a project manager. It is the time available for an activity in addition to its duration time. Since both start and end events of an activity have earliest and latest times, an activity has four assiciated times. Thus, there are four possible types of float but in practice only three of these are used.

Total float: This is the time by which an activity may be delayed or extended without affecting the total project duration. This is computed as follow:

$TF_{ij} = LT_j - ET_i - d_{ij}$

where TF_{ij} = total float for activity (i, j)

LT_j = latest allowable time for event j

ET_i = earliest expected time for event i

d_{ij} = the time duration for activity (i, j)

Free float: This is the time by which an activity may be delayed or extended without delaying the start of any succeeding activity. This is calculated as follows:

$FF_{ij} = ET_j - ET_i - d_{jj}$

where FF_{ij} = free float for activity (i, j)

ET_j = the earliest expected time for event j

ET_i = the earliest expected time for event i

d_{ij} = the time duration for activity (i, j)

Independent float: This is the time by which an activity may be delayed or extended without affecting the preceding or succeeding activities in any way. This is obtained as follows:

$$IF_{ij} = ET_j - LT_i - d_{ij}$$

where

IF_{ij} = independent float for activity (i, j)

ET_j = the earliest expected time for event j

LT_i = the latest allowable time for event i

d_{ij} = the time duration for activity (i, j)

Activity C

Compute total float, free float and independent float from the results you obtained Activity B.

14.10 PROBABILISTIC TIME ESTIMATES

Uptil now, we have discussed cases where the activity duration times were known with certainty. It is obvious that for most projects these activity times are random variables. PERT is more effective in handling cases in which activity duration times are uncertain. The PERT technique makes the following basic assumptions:

1. Activity times are statistically independent and usually associated with a 'beta' distribution.
2. There are enough activities involved in the network that the totals of activity times based on their means and variances will be 'normally' distributed.
3. The three estimates of the activity duration can be obtained for each activity.

The three time estimates are referred to as

i) Optimistic estimate denoted by a

ii) Most likely estimate denoted by m

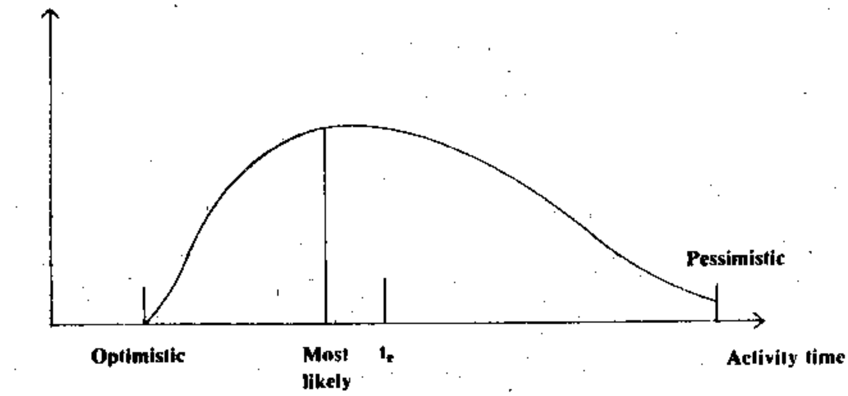
iii) Pessimistic time estimate denoted by b

The useful property of the beta distribution is that if we know the three time estimates (a, m and b) for an activity, we can compute mean or expected duration time (t_e) and the variance of duration (σ_{te}^2) as follows:

$$t_e = \frac{a + 4m + b}{6} \text{ and } \sigma_{t_e}^2 = \left(\frac{b-a}{6}\right)^2$$

The shape of the beta distribution is skewed. It can either be skewed left or skewed right. The following figure depicts a beta distribution which is skeward to the right.

Figure V



To demonstrate the use of PERT, let us continue with the same example. Instead of activity times to be known with certainty, let the three time estimates be as shown in Table 4:

Table 4

Activity	Predecessor activity	Time Estimates (weeks)		
		Optimistic a	Most likely m	Pessimistic b
A	—	2	3	10
B	—	2	3	4
C	A	1	2	3
D	A	4	6	14
E	B	4	5	12
F	C	3	4	5
G	D, E	1	1	7

In order to find the critical path, we need to determine the mean or expected duration for each activity. Once this is done, the procedures you have already learnt can be applied to find critical path. The expected time (t_e), is shown in Table 5:

Table 5

Activity	Predecessor activity	Time (weeks)			$t_e =$
		a	m	b	$(a + 4m + b)/6$
A	—	2	3	10	4
B	—	2	3	4	3
C	A	1	2	3	2
D	A	4	6	14	7
E	B	4	5	12	6
F	C	3	4	5	4
G	D, E	1	1	7	2

Note that the expected time (t_e) for each activity turned out to be the same as the single time estimate used earlier in this example. Obviously no coincidence, this was intentionally done for the sake of simplicity. The calculations for standard deviation and variance are shown in Table 6.

Table 6

Activity	Expected (mean) time (t_e)	Standard deviation $(\sigma t_e) = \frac{b-a}{6}$	Variance $\sigma^2 t_e$
A*	4	8/6 = 4/3	16/9
B	3	2/6 = 1/3	1/9
C	2	2/6 = 1/3	1/9
D*	7	10/6 = 5/3	25/9
E	6	8/6 = 4/3	16/9
F	4	2/6 = 1/3	1/9
G*	2	6/6 = 1	1

*Critical activity

The distribution of each activity completion time is 'normally' distributed. Thus, the expected activity times for critical activities are also normally distributed. The following table shows the critical activities along with expected time (t_e), standard deviation (t_e) and variance (t^2_e).

Table 7

Critical activity	expected time (σt_e)	standard deviation (σt_e)	Variance ($\sigma^2 t_e$)
A	4	4/3	16/9
D	7	5/3	25/9
G	2	1	1
	$\sigma t_e = 13$		$\Sigma \sigma^2_{te} = 50/9$

$$\sigma t_e = \sqrt{\Sigma \sigma^2 t_e} = \sqrt{50/9} = 2.357$$

14.11 PROBABILITY OF PROJECT COMPLETION BY A TARGET DATE

Sometimes, the management would also like to know the probability of completing the project by a particular date. Let us assume that in our example, we are required to complete the project within 11 weeks.

We know that the expected activity times for critical activities are also normally distributed (central limit theorem).

Therefore in order to find the probability of project completion by a target date, we can use the following formula:

$$Z = \frac{x - t_e}{\sigma t_e}$$

where X = target project completion time
t_e = expected project completion time
σt_e = standard deviation of activities on the critical path

In our example expected activity times follow a normal distribution with mean time σt_e =13 weeks and standard deviation t_e = 2.357 weeks. The target due date is 11 weeks. Using the formula, we get

$$Z = \frac{11 - 13}{2.357} = - 0.85$$

We can find the probability to any value of Z from the standard normal distribution table. The probability for the value of Z =-0.85 is 0.8023. Since Z = - 0.85, we must subtract 0.8023 from 1.0. Thus we obtain 1-0.8023 = 0.1977. Therefore the required probability of completing the project within 11 weeks is 0.1977 or 19.77 per cent.

Suppose now we are interested in finding the probability of completing the project in 16 weeks. Again using the formula, we get

$$Z = \frac{16 - 13}{2.357} = - 1.27$$

Thus the probability of completing the project is 0.8980 or 89.8 per cent.

Activity D

Consider a project having the following activities and the associated time estimates.

Activity	Predecessor activity	Time (days)		
		Optimistic	Most likely	Pessimistic
A	—	2	4	6
B	A	8	12	16
C	A	14	16	30
D	B	4	10	16
E	B, C	6	12	18
F	E	6	8	22
G	D	18	18	30
H	F, G	8	14	32

a) Draw the project network diagram

b) Identify the critical path and compute the expected project completion time.

c) What is the probability that the project will be completed on or before 55 days?

d) What is the probability that the project will be completed after 70 days?

14.12 SUMMARY

PERT/CPM is a network technique that is very useful to a project manager throughout all phases of a project. An understanding of events and activities and an appreciation of the inter-relationships between them are necessary before a network for the project can be constructed. A network can provide information such as earliest expected time, latest allowable time, slack and critical path. Activity times may be deterministic or probabilistic in nature. PERT introduces probabilitic aspects to the project network. It uses three time estimates: Optimistic, most likely, and pessimistic. The random characteristics of activity times are considered to follow beta distribution. The use of normal distribution assists the manager in determining the probability of project completion within a certain specified time period.

14.13 KEY WORDS

Activity: A clearly definable portion of a project that requires for its completion, the consumption of resources, and time in particular.

Critical activity: An activity becomes critical, if delay in its estimated time duration delays the whole project to that extent.

Critical path: The longest path through the network, consisting of critical activities. The length of the critical path is the shortest time allowable for project completion.

Dummy activity: Dummy activity is an activity which does not consume resource or time. It is used in network to show logical links between other real activities.

Event: An event represents a specific accomplishment in the project and takes place at a particular instant of time and therefore does not consume resources or time.

Earliest expected time: The earliest time that an event can occur is on the latest completion of an activity.

Float: It is the amount of time available for an activity in addition to its duration time. Float is computed in relation to activity.

Latest allowable time: The latest time that the event can be delayed without delaying the completion of the entire project.

Slack: The amount of time by which the start of an activity may be delayed without affecting the overall duration of the project. Slack is computed in relation to events.

14.14 SELF ASSESSMENT EXERCISES

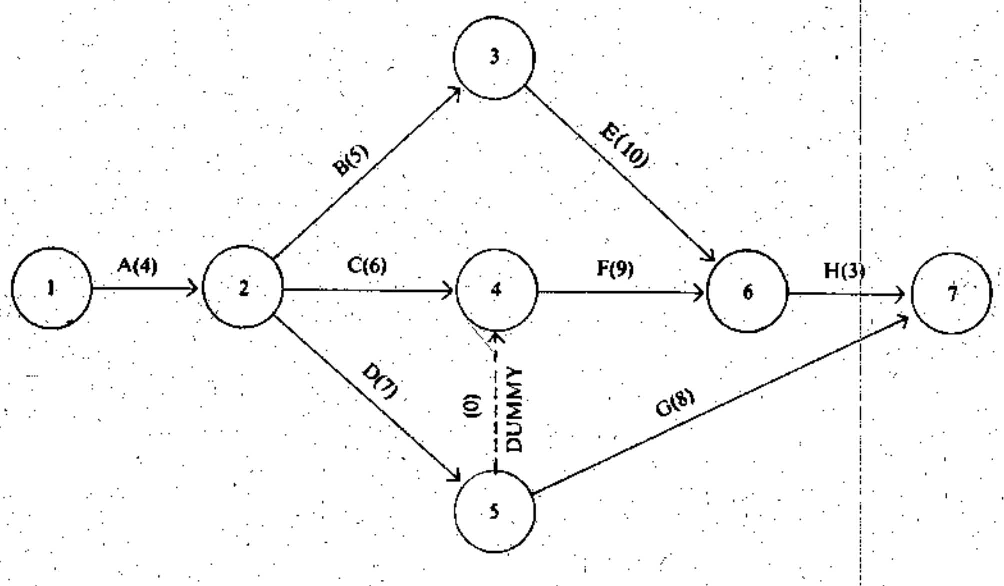
1 How does network analysis help in large complex projects?

2 What purpose in served by including dummy acitivities in network diagram?

3 Explain PERT and its importance in network analysis. What are the requirements for applications of PERT techniques?

4 Illustrate with examples the essential difference between PERT and CPM techniques.

5 Find critical path method and project duration for the following project network.



6 For processing a job at a data-processing centre, certain steps need to be taken. These jobs can be described as follows:

Job	Description	Immediate Predecessors	Time (minutes)
A	Design flowchart and write fortran statements.	—	180
B	Punch control cards	A	30
C	Punch comment cards	A	20
D	Punch programme cards	A	60
E	Obtain brown folder	B, C, D	10
F	Put deck together	B, C, D	20
G	Submit deck	E, F	10

Draw a critical-path arrow diagram and indicate the critical path. What is the minimum time required for completion?

ii) What is the free float of job C?

7 Draw the arrow diagram, identify the critical path and compute total and free floats for the activities in the project of planning a rural piped-water supply.

Activity identification	Activity description	Immediate predecessor(s)	Expected duration (weeks)
a	Excavation of well	—	8
b	Collection of 10% popular contribution	a	9
c	Completion of well	b	7
d	Detailed plans of supply system	a	15
e	Pump house construction	c, d	4
f	Standpost construction	e	1
g	Construction of reservoir	d	4
h	Laying of pipelines	e, g	5
i	Roadside taps	h,f	4

8 A project comprises eight independent activities. Diagram the project and identify its critical path. What is the expected time to complete the project? Calculate the total and free floats for non-critical activities. What is the probability of completing the project in 20 weeks or less? Time estimates (in weeks) are as follows where a = most optimistic time, m = most likely time, b = most pessimistic time.

Activity	Predecessor Activities	a	m	b
A	—	1	3	5
B	—	2	3	4
C	—	3	4	5
D	A	2	9	10
E	C	4	5	6
F	B, D, E	5	6	13
G	A	2	4	6
H	C	1	3	6

14.15 FURTHER READINGS

Cook, T.M. and R.A. Russell, 1981. *Introduction to Management Science*, Prentice Hall: Englewood-Cliffs.

Cooke, William P., 1985. *Quantitative Methods for Management Decisions*, McGraw Hill Book Company: New York.

Dannenbring, D.G. and M.K. Starr, 1981. *Management Science*, McGraw Hill Book Company: New York.

Gordon, G. and I. Pressman, 1983. *Quantitative Decision Making for Business*, Prentice Hall: Englewood-Cliffs.

Gupta, M.P. and J.K. Sharma, 1987. *Operations Research for Management*, National Publishing House: New Delhi.

Lock, D., 1987. *Project Management Handbook, Gower:* London.

BLOCK 5 MANAGEMENT INFORMATION SYSTEM

This block comprises of three units.

Unit 15 describes the evolution of MIS in a way, leading to fair understanding of its status in the organisation. It also explains different frameworks for effective creation and successful implementation of MIS through adequate involvement and proper development of the system and skills within available resources.

Unit 16 deals with basic ingredients of MIS i.e. data and information processing, along with a historical perspective. It presents a brief account of Data life cycle leading to information through processing or manipulation. The concepts cost, value and quality have been discussed under Information Economics.

The last unit deals with the systems view of information and its control. Structure and designing of a desirable MIS has been discussed, to facilitate the internal operation of the organisation and information flow therein.

UNIT 15 AN MIS PERSPECTIVE

Objectives

After going through this unit, you should be able to:

- understand the main features of a Management Information System (MIS)
- have an idea about the development of MIS and its present status in the organisations
- appreciate different viewpoints regarding applications of MIS in organisations.

Structure

15.1 MANAGEMENT INFORMATION SYSTEMS - AN INTRODUCTION

The subject of management information systems evokes emotions on a wide spectrum ranging from disdain to rapture. It would be no exaggeration to say that few areas of management have stirred more acrimonious debate. A large part of the controversy, however, is a matter of semantics. It boils down to sharply differing notions of what constitutes a Management Information System. Some people have used the term to describe systems in which a manager has instantaneous access to detailed pieces of information regarding the entire organisation. Some see it merely as an appendage to the accounting system which blends into the usual financial type of summary statements. Usage of the term sometimes includes any information processed by computers. Definitions that are most useful in practice are those which emphasise the use that is made of the information provided by an MIS rather than the technology or methodology employed in collecting and disseminating the information. We will, therefore, adopt the following definition, which closely parallels those given by Kanter (1) and Davis (2).

A Management Information System is an integrated man-machine system that provides information to support the planning and control functions of managers in an organisation.

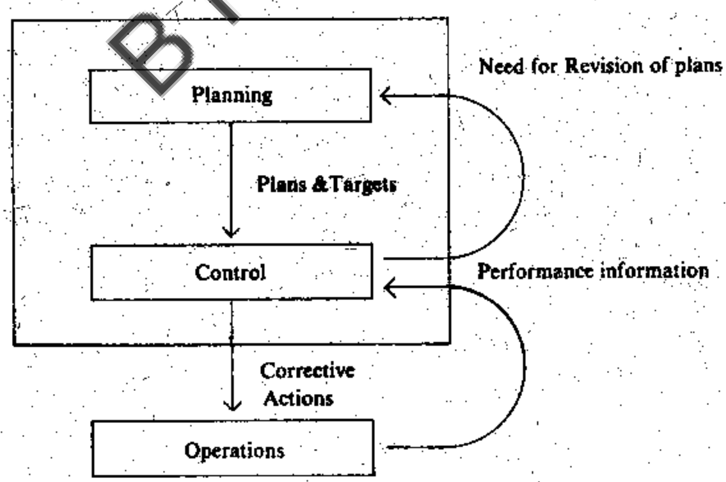
Let us make a few remarks in elaboration of this definition:

1) The output of an MIS is information that subserves managerial functions. If a system provides information to persons who are not managers we will not consider it as part of an MIS. For example, an organisation often processes a lot of data which it is required by law to furnish to various government regulatory agencies. Such a system, while it may have interfaces with an MIS, would not be a part of it. Instances of such systems are salary disclosures and excise duty statements. By the same token to sophisticated computer-aided design system for engineering purposes would also not be a part of an MIS.

2) An MIS deals with information that is systematically and routinely collected in accordance with a well-defined set of rules. This implies that an MIS is a part of the formal information network in an organisation. Information that has major managerial planning significance is sometimes collected at golf courses. Such information is not part of MIS, however. One-shot market research data collected to gauge the potential of a new product does not come within the scope of an MIS by our definition because although such information may be very systematically collected it is not collected on a regular basis.

3) The information provided by an MIS assists managers to make planning and control decisions. Let us clarify what we mean by planning and control. Every organisation in order to function must perform certain operations. For example, a car manufacturer has to perform certain manufacturing activities, a wholesaler has to receive and dispatch goods, a municipal corporation has to provide water to its area of jurisdiction. All these are operations that need to be done. In addition to performing these operations, an organisation must make plans for them. In other words it must decide on how many and what type of cars to make next month or what commissions to offer retailers or what pumping stations to instal in the next five years. Also an organisation must control the operations in the light of the plans and targets developed in the planning process. The car manufacturer must know if manufacturing operations are in line with the targets and if not, he must make decisions to correct the deviation or revise his plans. Similarly the wholesaler will want to know the impacts that his commissions have had on sales and make decisions to correct adverse trends. The municipal corporation will need to control the tendering process and contractors who will execute the pumping station plans. The diagram below depicts the relationship between operations, planning and control.

Figure I : Relationship Between Operations, Planning and Control



By our definition an MIS is concerned with planning and control. Often there are elaborate systems for information that assists operations. For example, the car manufacturer will have a system for providing information to the workers on the shop floor about the job that needs to be done on a particular batch of material. There may be route sheets which accompany the rate materials and components in their movement through various machines. This system per se provides only information to support operations. It has no managerial decision-making significance. It is not part of an MIS. If, however, the system does provide information on productivity, machine utilisation or rejection rates, then we would say that the system is part of an MIS.

To take another example, the wholesaler may have a computer system to send out bills to his retailers. If this was all that the system did, it would only be supporting operations. If, however, the system uses the bills to produce information on sales, profitability and retailer performance, it would qualify under our definition as part of an MIS. We will see that often systems that provide information on operations can be the basis for information for planning and control. However, we must be careful to realise that in and of themselves, they are not management information system no matter how sophisticated the technology employed.

4) An MIS includes all the ingredients that are employed in providing information support to managers in making planning and control decisions. Managers often use historical data on an organisation's activities as well as current status data to make planning and control decisions. Such data comes from a data base which is contained in files (paper or electronic) maintained by the organisation. This data base is an essential component of an MIS. Manual procedures that are used to collect and process information and computer hardware are obvious ingredients of an MIS. Less obvious, but equally important are the computer programme used to process information, and operations research models employed to marshall the data to provide highly processed information to support decisions. These also form part of the MIS. In summary, when we say that "an MIS is an integrated man-machine system that provides information to support the planning and control functions of managers in an organisation" we mean that it is a system which:

- subserves managerial functions
- collects information systematically and routinely
- supports planning and control decisions
- includes files, hardware, software and operations research models.

15.2 HISTORICAL BACKGROUND

Having clarified what we mean by an MIS let us examine its historical roots. Surprisingly, Management Information Systems are in fact as old as the oldest writings in the world. The oldest evidence of writing by man discovered so far consists of clay tablets excavated at Sumer in Mesopotamia and dated approximately 3000 B.C. These contain records from an inventory system carrying information on receipts and issues made to individuals from a temple grain store. In fact, many historians believe that writing arose in response to the need for such management information. For example, W.H. McNeill, Chairman of the Department of History at the University of Chicago, says:

> " At first, Sumerian priests used writing mainly to record deposits and withdrawals from temple store houses. A persistent problem here was how to find ways to record the names of the men who engaged in these transactions. Eventually the effort to record individual men's names in recognizable form induced the priests to resort to equivalences between syllables in men's names and the sound of some easily pictured word. Then by developing enough standard syllable pictures the scribes would easily learn to record all the sounds of ordinary speech."

He goes on to add:

> "It seems probable that all known forms of writing derive directly or indirectly from the Sumerian invention."

But why is MIS a subject of great interest today? There are two principal reasons for this.

Firstly, organisations have grown in complexity to levels which are unprecedented and information plays a vital role in holding together and coordinating organisations. Information is the mortar that holds together the edifice of the modern multi-division, multi-location, multi-product organisations. The role of the management information system is very similar to that of the nervous system in animals.

One notices that with evolution of creatures of greater structural complexity and functional specialisation, the nervous system too grows greatly and becomes critical in the survival of the animal. It is said that one of the reasons for the extinction of dinosaurs was the poor capability of their nervous system at low temperatures. One has only to imagine an organisation in which one department, say production, does 'not speak to' another department, say marketing, to realise the fundamental importance of information in coordinating the activities of an organisation. Indeed, management is nothing if it is not coordination and control. Despite this fact it has only been lately that management has paid any explicit attention to its information processes. Until recently information occupied the status that oxygen did before lavousier's discovery of the gas-it was both vital and unrecognised.

The second reason is the advent of the computer. Computers are able to both access and record information and perform calculations at speeds which are almost unbelievable. Even older (1960 vintage) computers can access information at a rate higher than 20,000 characters per second from magnetic tape which would correspond to reading and writing about 400 pages of an average sized book in one minute! When it comes to calculation, the computer is even more of a whiz kid. It can do about 50,000 multiplications in one second! These figures are not for unusual computers but for very modest-sized older models. More recent computers have add times in nanoseconds. A nanosecond is the time taken by light travelling at 186,000 miles per hour to move a distance of one foot! But it is not only the speed of computers but rather the reduced cost of information handling made possible with computers that has launched them into the management environment. It is a striking fact of computer technology that between its arrival on the management scene in the mid fifties and today, the cost of information processing (for arithmetic as well as filing and retrieval) has decreased by at least two orders of magnitude. It is as if the cost of an airplane ticket from Bombay to New York had dropped from $ 800 to about $ 8 in 20 years! Or the cost of the average automobile had dropped from Rs. 20,000 to Rs. 200.

It is clear that the sort of tehnological environment provides a challenge to management. Traditionally management had to think of managing four resources: money, materials, men and machines. It must add a fifth resource to this list, namely 'information'. In fact, some researchers in management have gone so far as to define a manager as a translator (transducer) of information into decisions.

15.3 STATUS OF MIS IN ORGANISATIONS

Traditionally management information systems have not really been designed at all. They are the end product of a process in which manual systems are automated in a piecemeal fashion.

The choice of applications is largely dictated by what similar companies are doing with computers or else such areas are taken up where the operational problems are most visible. Generally what may be called as MIS is nothing but a limited analysis of data captured through routine data processing applications. Thus analysis of invoice data leads to a sales analysis system in which region-wise, outlet-wise, product-wise sales is reported periodically. Similarly a few reports on overstocked, or out of stock items from the stores accounting data pass for an inventory control system.

Such applications may result in benefits which outweight costs of computerisation in a marginal way. However, this does not represent a use of computers which will have the maximum pay off for the organisation.

15.4 FRAMEWORK FOR UNDERSTANDING MANAGEMENT INFORMATION SYSTEMS

The information processes in an organisation ar labyrinthine and without some overall map to guide our steps in studying them we should soon be lost in a mass of unstructured detail. Such maps are provided by general frameworks that seek to sharpen important distinctions in the kinds of information that support managerial decisions. A number of such frameworks which, in our opinion, provide the most insight from a pragmatic standpoint are discussed below.

Robert Anthony has delineated a framework which distinguishes between the different types of planning and control process that typically occur in organisations. His basic thesis is that thinking of planning and control as two separate and homogeneous activities in an organisation is not only meaningless but positively dysfunctional. Instead of this segmentation of management planning and control activities into two categories of planning and control, Anthony suggested that the area of management planning and control be segmented into three categories, resisting the "natural temptation to use as the two main divisions: (1) planning (roughly), deciding what to do, and (2) control (roughly), assuring that desired results are obtained". The three categories suggested by Anthony are:

1) Strategic planning
2) Management control
3) Operational Control

Essentially, according to Anthony, planning and control activities are so closely inter linked as to make the separation of these activities undersirable and meaningless. Instead, according to him, it makes much more conceptual and practical sense to link together planning and control activities which are similar and inter-twined.

Anthony's definitions of these three sub-species of planning and control are:

1) **Strategic Planning** is the process of deciding on objectives of the organisation, on changes in these objectives, on the resources used to attain these objectives, and on the policies that are to govern the acquisition, use and disposition of these resources.
2) **Management Control** is the process by which managers assure that these resources are obtained and used effectively and efficiently in the accomplishment of the organisation's objectives.
3) **Operational Control** is the process of assuring that specific tasks are carried out effectively and efficiently.

Anthony's departure from the traditional concept of separate planning systems and control systems is diagrammed in Figure II (a) and (b) respectively.

Figure II (a): Traditional Segmentation of Planning and Control Systems

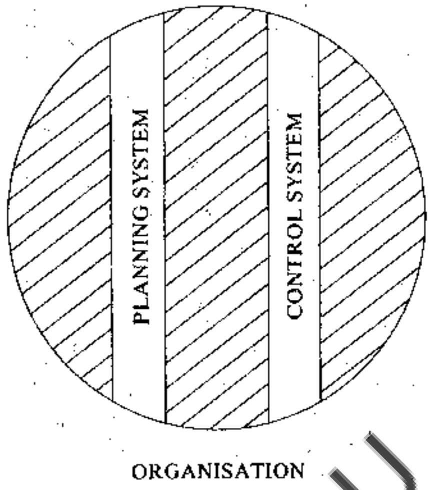
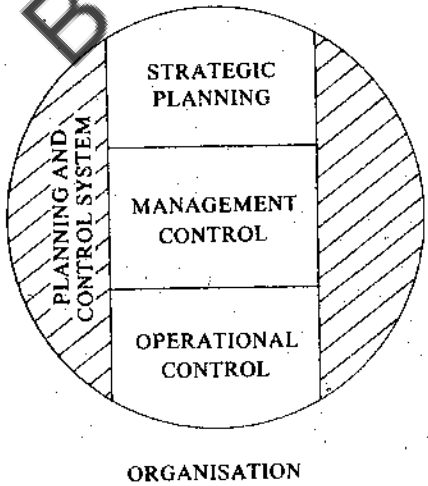


Figure II (b): Anthony's Framework of Planning and Control



It is useful to clarify the above definitions with some examples. The table below gives instances of planning and control activities in different functional areas classified according to the above definitions.

	Strategic Planning	Management Control	Operational Control
Production	Location of a new factory	Determining the product mix for a monthly production programme	Scheduling specific jobs on specific machines in a shift
Marketing	Entering the Export market	Media planning for advertising expenditures	Planning sales contacts to be made by a sales man in the next week
Finance	Raising capital by issuing new shares	Determining maximum levels of credit for customers	Determining what action to take againist nonpayment by a specific customer
Personnel	Deciding on changes to be made in the organisation structure	Determining who will be promoted to fill a vacated post at middle and lower levels in the organisation	Determining which workers will be on each shift

Anthony's framework enables us to understand the characteristics of information needed to support the three types of planning and control processes.

The table below depicts these characteristics and highlights the substantial differences in information required for strategic planning, management control and operational control.

Information Characteristic	**Strategic Planning**	**Management Control**	**Optional Control**
1. Volume	Low	Intermediate	High
2. Level of Aggregation	High	Intermediate	Low
3. Frequency of use of a particular type of data	Low	Intermediate	High
4. Currency requirement	Low	Intermediate	High
5. Accuracy	Low	Intermediate	High
6. Scope	Wide	Intermediate	High
7. Source	Significant amount from external sources	Mostly Internal	Entirely Internal

8.	Predictability of use*	Low	Fairly High	Very High
9.	Variability with user**	High	Intermediate	Low
10.	Distance of user (in Organisational terms) from sorces within organisation	Far	Fairly close	Close

* How far in advance can the information that will be needed for a decision be stipulated?

** For a given decision, how much of the information considered necessary likely to vary from one individual to another.

Another framework which is useful in structuring our understanding of an MIS is one provided by Simon. Whereas Anthony's framework is concerned with the objective of the decision-maker, i.e. what the manager is trying to do, Simon's framework examines the process of decision-making i.e. how does the manager make decisions.

Simon breaks down the process of making a decision into three stages.

1) **Intelligence:** This is the stage in which the decision-maker recognises that there is a problem or opportunity that requires him to make a decision.

2) **Design:** This stage covers the determination of the alternative actions that he could take to resolve the problem or exploit the opportunity.

3) **Choice:** This stage is concerned with the process by which one of the alternatives generated in stage-2 is singled out to be pursued.

With this framework we can distinguish between three major classes of decisions. Programmed decisions are those in which all stages are handled by following a preset well-defined procedure. These decisions are repetitive and routine which arise often and are capable of being modelled mathematically in their entirety. The classic example would be inventory ordering decisions.

Non-programmed decisions are those where none of the stages is amenable to handling by a well-defined, pre-specified procedure. (These decisions are novel and difficult to structure in logical-mathematical terms. They have to be treated de novo whenever they arise.) An example would be the decisions to set up a new factory or launch a new line of products.

Semi-programmed decisions are those in which at least one and no more than two of the above stages can be handled by a well-defined preset procedure. An example where the intelligence phase is well-structured would be the diverse kinds of variance analysis. Here comparison with a budget or standard is undertaken in a well-defined way to signal the need for a decision. Subsequent stages of design and choice, however, are not handled by a set procedure.

Zani's framework draws upon the earlier two frameworks. He argues that effective MIS can only be designed in a top down fashion, viewing the organisation's information needs from the vantage point of managers who will use it rather than in a 'bottom up' manner which automates existing clinical procedures.

According to Zani, the important determinants of MIS design are:

- Opportunities and risks
- Company strategy
- Company structure
- Management and decision-making processes
- Available technology
- Available information sources

Opportunities, risks, competencies and resources, plus the strategy derived from them, yield the company's organisational structure. This structure sub-divides the essential tasks to be performed, assigns them to individuals, and spells out the interrelationships of these tasks. These tasks, and the organisational structure they compose, determine the various information needs of the company.

Every organisation must understand its key success variables which are activiities on which the company must score high for it to succeed. For example, a textile company in fashion market must focus on product promotion, understanding customer response to product and monitoring competitive changes. Whereas a mill selling grey cloth (unprocessed) must focus on manufacturing and distribution costs.

The key success variables name the key tasks of the company and thus help identify the priorities for information system development. The system must provide information that makes the individual manager's performance of these tasks easier and better.

These tasks could be related to strategic planning, management control or operational control. The content and frequency of reports that will provide the necessary information for key task must be identified through an analysis of the decision-making processes. Here the frameworks proposed by Anthony and Simon prove to be useful.

Zani advocate participative process of MIS design where top management and functional managers help in understanding critical areas of operations, identification of specific information requirements. To fulfil this role properly, managers must be aware of the major sources of information, of alternative methods of supplying data, and of the impact of the major changes of information techonology. The major contribution to information systems in these areas, of course, must come from the information and data-processing specialists.

Using the framework, then, encourages understanding of the critical areas of operations, identification of specific information requirements, and recognition of the technological, economic and personnel constriants within which an MIS develops. As important as anything else, perhaps, is the fact that systems are of necessity dynamic, changing with the environment and the organisation.

In designing an MIS there are two types of situations one may come across.

If the organisation has no experience of computing, applications which will create the maximum impact on the organisation can be identified by using Zani's framework. Key success variables are however seldom obtained through a questionnaire survey of managers. Data on environment, past company performance must be analysed and discussed to identify key success variable. It is sometimes useful to pen down a quantitative measure of such variable.

For example the performance of a textile unit can be summed up through two indicators-contribution per loom shift and fixed cost per loom shift. Similarly the performance of a shipping company may be measured as gross operating profit per day per voyage. Precise definitions of performance indicators enables the analyst to understand and quantify the likely impact of improvement in different tasks of planning and monitoring.

An analysis of the company's key success variables can be done only after a thorough understanding of the company's operations. Consultants and vendors who do not spend adequate time in understanding the operations are unlikely to throw up applicaiton areas which will create the maximum impact. They are likely to suggest 'off-the-shelf' applications. For such applicaitons use standard software which is available.

For a company getting into computerisation for the first time, a list of applications would have to be generated, keeping in view a 4-5 year perspective on the basis of which a suitable configuration would be decided. However the development and implementation of the applications would have to be done in a phassed manner. The first few applications must be those, which can create an impact on the performance of the organisation, are quick to implement with the least amount of changes in the existing procedures and systems. Initial success can make the later implementation of complex and more involved systems easier.

For organisations which have been into data processing and would like to graduate to MIS, the choices are somewhat limited. Existing computer technology, manpower and past experience with computer applications all such factors will condition the future growth.

By and large an effort is made to create useful data bases which capture data during the execution of routine data processing systems. Such data are then analysed to produce periodic planning report for monitoring.

Examples of such systems are the sales analysis based on invoice processing; inventory control based on stock accounting; costing and profitability analysis on the basis of financial accounting systems. Of course marginal additions to data fields, new coding structure, revised procedures are introduced to make the data base and reporting more useful.

Factors Facilitating Implementation of MIS

A few factors which will increase the chances of a successful implementation of MIS are:

1) Involvement of top management in the computerisation effort, in defining the purpose and goals of computers within the organisation.

2) Selection of an EDP Manager who has the political skills to involve managers in choosing application areas, identifying information needs and designing reports.

3) A computer staff which has interdisciplinary skills in computers, management and operations research.

4) A balanced expenditure on hardware and software.

15.5 SUMMARY

This unit has given you a fair understanding of the main features of a Management Information System in Organisational Context, describing its various functions, importance and relationship with planning, control and operations in an organisation i.e. what an MIS is and what it is not.

Further, the unit systematically leads you to the prevailing status of MIS in organisations, discussing the ever growing need of information and its proper handling (processing), which in turn led to the development of MIS and advent of computers therein to cope with the hazards faced in coordinating and managing the organisational challenges. We have also discussed different viewpoints about the MIS given by some management scientists.

15.6 SELF-ASSESSMENT EXERCISES

1 Critically evaluate the status of MIS prevailing in your organisation in the light of the understanding you have developed from this unit, and also propose adequate measures to improve it.

2 In what respects do the three models of MIS, discussed in this unit, differ significantly? Describe in brief.

3 What do you understand by programmed, semi-programmed and non-programmed decisions; describe with the organisational examples, you have come across?

4 How do you differentiate between planning and control in operations of an organisation?

15.7 FURTHER READINGS

Kanter, J., 1972. *Management Oriented MIS,* Prentice Hall Inc: Englewood-Cliffs.

Davis, G.B., 1974. *MIS Conceptual Foundations, Structure and Development,* McGraw Hill: New York.

Zani, W.S., 1973. "A blue print for MIS", *Harvard Business Review.*

UNIT 16 INFORMATION NEEDS AND ITS ECONOMICS

Objectives

After going through this unit, you should be able to:

- have a historical perspective on data and information processing
- distinguish between data and information
- conceptualise the data life-cycle
- identify both the logical and physical ways in which data can be processed or manipulated to produce information.
- appreciate the primary aspects of cost, value and effectiveness as they relate to data and information processing
- appreciate mathematical information theory.

Structure

16.1 INTRODUCTION

Some the dawn of civilisation, people have required information to aid them in their personal battle of survival as well as in their attempts to manage their organisations. Though for centuries man lived on earth without keeping records, it became necessary for man to adjust with the growth of social organisations. As tribes grew into nations, trade and commerce developed. Historians have traced some type of record keeping and data processing back to about 3500 BC. when in Babylonia, merchants were keeping records on clay tablets. At about the same time, the ancient Egyptians made a great improvement in record keeping when they developed papyrus (the fore-runner of paper) and a sharp pointed pen called a 'calmus'. Today merchants and all of us still keep records and process data, but the technology by which data is processed has had a quantum leap forward especially in the last fifteen to twenty years. The world has gone through the agricultural revolution, industrial revolution and we are in the midst of the so called 'information' revolution.

16.2 GROWING NEED FOR INFORMATION

There were two primary reasons for processing data before eighteenth century. First there was a natural desire of men to keep an account of their possessions and wealth as exhibited by the Babylonian merchants. Second reason for processing data before eighteenth century was essentially because of governmental requirements for compiling administrative survey data for raising taxes and conscripting soldiers. Around fifteenth century Luca Pacioli developed the double entry book keeping system, the fore-runner of our modern financial accounting systems.

With the dawn of the industrial revolution, the basic tasks of production shifted from home and small shops to the factories. With the advent of factories, other service industries like marketing and transportation grew. The increasing size and complexity of the organisation prohibited any one individual from effectively managing them without some data and its appropriate processing. As factories grew in size it necessitated the need for more sophisticated capital goods, equipment, plant and machinery, requiring large investments. On one hand, management needed more information for internal decisions while investors, on the other hand, needed information about the organisation, its soundness, health and credibility of its managerial performances. In spite of the technological developments, there is an ever increasing pressure for more and more information.

Granting of loans, credits etc. created a need to maintain accounts receivable, accounts payable etc. It has now become statutory for any type of organisation to display certain financial statements like a balance sheet, profit and loss statement etc. duly audited and certified by chartered accountants.

There was a swing from integration to fragmentation and now we are trying to integrate once again. Most organisations were run by single individuals. In government, this figure was the ruler/king, but in business it was the owner. However, with the increasing size and complexity in government, parliaments evolved and took away many of the king's powers and duties. Bureaucratic groups were established and made responsible for specific areas, thereby resulting in erosion of authority of the central figure and fragmentation of responsibility. In business, the owner began by managing his own concern. But as the business diversified and grew, it became necessary to delegate authority to other men. The communication lines and their length increased. The result was that the owner often had very little knowledge and control of what decisions were made at the lower levels. More people at the lower levels got involved in their immediate problems losing sight of the organisation's overall goals. Fragmentation had thus become a necessary evil. The situation as of today, however, is that there is a visible movement back to some type of an integration because of the advent of technology in the form of computers and electronic data processing.

The executives in Indian industry, though they complain of many things, have three areas in common. The first is personal income tax. The second is the increasing amount of paper work and the third is the paucity of 'information' available for their decision-making. At the first glance the second and third complaints seem contradictory. On the one hand, information is pouring in through paper submerging the executive while on the other hand there is not adequate information for rational decision-making. "Water, water, everywhere but not a drop to drink!" This contradiction is present because there is abundant 'data' but a great paucity of 'information'.

In general usage, the terms data and information are often used interchangeably. However, there is a difference between data and information.

16.3 DATA

Data are a set of isolated raw facts, figures, statistics, unrelated and uninterpreted. Data could be defined as a collection of numbers, letters or symbols that can be processed, maintained or produced by a management information system (MIS). Data may not be organised, may not add to our knowledge and may not surprise us. Organisational activities like sales, production, billing, collecting amounts due, and assigning personnel are examples of activities that use data. Data items have value only if they are useful in the performance of organisational activities. A tray of cards representing sales invoices by itself means very little to the sales manager. Sometimes many reports are generated through data processing systems and placed on the management desk. This could be compared to a daily newspaper printed without headlines or spacing between words-an avalanche of data, but a paucity of information. Now what then is information?

16.4 INFORMATION

Information is the result or product of processing data. It provides knowledge or intelligence. It contains an element of surprise, reduces uncertainty and triggers off action. Information is an occurrence or a set of occurrences which carry messages and when perceived by the recipients via any of the senses, increase their state of knowledge.Information is the behaviour-initiating stimuli between sender and receiver. Information is in the form of signs that are coded representations of data. Data becomes information when it is processed and placed in some context. Information is the 'stuff' of paperwork systems just as material is the 'stuff' of production systems. In fact the relation of data to information is that of raw material to finished product. Information for one person may be the raw material for another. Information is now considered a resource on the level of money, materials, men, machines, methods, markets and managements. Thus information resources could supplement the existing resource list of M's by introducing messages and moments in the context of information resource management. Information resources (in the sense of stored data of all types) are re-usable.

Realistically, information communicates the state of a situation, but the perception can differ significantly between users. Information containing the same message content and quantity communicated in the same way and at the same time does not mean that it will be used similarly by the receivers. This could depend largely upon the perception, background and prejudice of the users.

Information is information only to the extent it serves the needs of the manager. It is significant for him only if it can add to his knowledge in planning the operations and help him in discharging his personal responsibility. If the data is properly organised, the manager will be able to react to it. If the data is not properly organised, it will not be helpful to him in solving various managerial problems. Since a manager has multifarious responsibilities the time at his disposal is limited. Therefore, his information needs should be identified and only that data which can be helpful to him should be accumulated and supplied to him.

Activity A

List down five elements in an organisation which you would term as data and/or information and why?

1. ....

....

2. ....

....

3 ....

....

4 ....

....

5 ....

....

Information Classification

We can classify information found and used in an organisation into the five groups as follows:

i) **Action Versus Non-action Information:** Here a non-action information is synonymous to data. Action information requires the recipient to do something when information is received.

ii) **Recurring Versus Non-recurring Information:** Recurring information is one which is generated at regular intervals such as periodic reports. Non-recurring information is typified by some special kind of studies to aid in management decision. This is of a non-repetitive type nature.

iii) **Documentary Versus Non-documentary Information:** Documentary information is that which is preserved in some written form or available on microfilms, punch cards, magnetic tapes, floppy disks, accounting reports etc. Non-documentary information is transmitted orally or received by an individual observation.

iv) **Internal Versus External Information:** The distinction here is obvious. It will be seen in the subsequent unit that different managers at different hierarchies in the organistaion require different blend of internal and external information.

v) **Historical Information Versus Future Projections:** The distinction here is again obvious. The historical information could be futile, unless it can be used for future projection.

16.5 INFORMATION FROM DATA

Data Life-cycle

We can think of data having their own life-cycle namely, data generation, data manipulation, transmission of data (and communication of information) and storing/retrieving and reproduction of data.

The **generation** of data could take place internally and/or externally. This data has to be captured by recording of data from an event or occurrence in some form such as sales slips, personnel forms, purchase orders etc.

The captured data would have to be **stored** either in person's mind or in a document or in a 'mechanical' or electronic device, microfilm, punched cards/tapes or in a device of some suitable form before they may be operated upon or utilised.

Stored data would have to be **retrieved** by searching out and gaining access to specific data elements from the medium where it is stored.

Retrieved data may be **converted** or **reproduced** to a different form of storage or presentation format by way of documents reports etc.

Data are also constantly being transported to the user in processed form. It is transferred to storage from the source, then processed and passed on to the user, who again return it to storage after working on it, which becomes available for further retrieval.

The randomly accumulated data has to be **sorted** and **classified** to reveal appropriate information. For example, sales data can be classified product-wise, territory-wise, salesperson-wise etc. Such a classification will give the sales data more meaning.

Sometimes aggregation or **synthesis** of many pieces of data to structure a meaningful whole or complete report is often required.

Processing of data might entail quite a bit of manipulation and calculations involving addition, subtraction, multiplication, division etc. based on certain formulae. Computations might have to be performed for deriving employer's pay, customers bill, financial ratios etc. Management science/operational research models might be used for determining optimal product mix, aggregate planning and economic order quantity determination.

Data stored must be utilised on some occasion by some one at some point of time, otherwise there is no point in putting in on inventory. When data is finally put in a usable form it can be retrieved and turned into information at appropriate time for decision-making.

Some type of a continuous **verification** and **evaluation** of data ought to be taken because there is also an economic aspect of cost of processing data versus the value of the information. Therefore, data files should be continously monitored to eleminate useless data.

It is important to destroy data following its evaluation or use. Destruction of data records may be on a purely routine basis following one time use or may occur in review of old records. Destruction is the terminal stage or the end of the data life-cycle.

The data life-cycle is portrayed in Figure I.

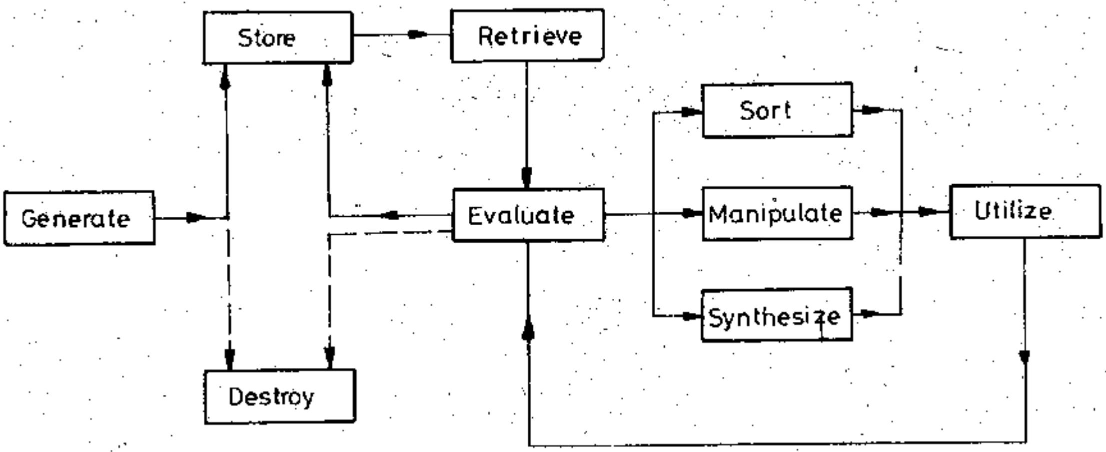


Figure I: Data Life Cycle (Source: R.G. Murdick & J.E. Ross Information System for Modern Management, Prentice Hall Int.)

Data Processing Methods

Every organisation whether business, government or social requires a certain amount of paper work or data processing which can be done in a variety of ways. In the past, data processing consisted of manual procedures whereby the data operations were performed by hand with the aid of basic devices such as pencil, paper, slide rule etc. Then came the electro-mechanical method which is actually a symbiosis of man and machine. Then came the sue of typewriters, cash registers, time clocks etc.

The punched card equipment method came into use along with the 'unit record system', the principle being that data concerning a person, object, or event is normally recorded (punched) on a card. A number of cards placed logically and sequentially in a deck was termed a 'file'. There was considerable reduction in the manual intervention. However, it was only after the development of the electronic computer that one machine became capable of performing most of the data operations without intermittent human intervention. Today data processing is generally assumed to be computer or **electronic data processing** (EDP). Computer systems were used initially in many organisations to perform essentially the same functions as ADP (Automatic Data Processing) systems.

The computer simply means a configuration of input devices, a central processing unit (CPU) and output devices. A major innovation in the development of CPU was the stored programme concept. A computer executes only the instructions given to it at fantastic speed. They are highly suited to highly repetitive type of calculations. However, whenever judgement is to be exercised, humans remain far superior, though of late more researches are being conducted in Artificial Intelligence (AI) and the development of knowledge based expert systems. Recently significant technological developments (especially in the field of electronics) have reduced the cost and size of the computer to such an extent that organisations of almost any size can now invariably benefit by using computer data processing. Computers of various types- mainframes, minicomputers, micro-computers and personal computers are there for the asking. Though the hardware costs are coming down, the software costs, however. are rising. Let us now figure out the cost of information.

16.6 INFORMATION ECONOMICS

Information is a valuable resource in any organisation. However, the preparation of formal information is not free; it costs money. How much should an organisation spend for information. Some type of a cost-benefit analysis ought to be undertaken. It is more easily said, than done. Difficulties occur in measuring the cost of providing the information and measuring the value of information. Information is conceptual in nature and possibly has hardly any tangible characteristics, except symbolic representations. However, let us attempt to understand some aspects of information economics.

Cost of Information

The costs of operating the information system could be categorised as follows;

i) **Hardware Costs:** This is normally a fixed or sunk cost over a relevant range in case of computer based information system. With the quantum jumps in the field of electronics, the hardware costs are coming down appreciably.

ii) **System Analysis, Design and Implementation Costs:** This again is a sunk cost. This function includes formulating a methodology for overall electronic data processing procedure. This should include the cost for preparation of programmes and the so called software costs. With computers being the 'in thing' these days, software people are pitching their costs higher and higher.

iii) **Conversion Costs:** This is a sunk cost and includes any kind of change from one method of data processing to another.

iv) **Cost of Space and Environmental Control Factors:** This cost is semi-variable. Examples of this cost are floor space, air-conditioners and dehumidifier systems, power control units, standly generator, security and so on. The size of the computers has shrunk considerably from the first generation down to the fifth generation computers. Sizes have shrunk whereas power and capabilities have increased manifold. Many mini-and-microcomputers of today are rugged enough and do not even require an air-conditioned room (like the BBC Micro's etc. which have been introduced in schools both in rural and urban settings in India).

v) **Operation Costs:** This is basically a variable cost and includes a variety of personnel, facilities and systems maintenance, supplies, utilities and support facilities costs.

These costs are often classified as either variable or non-variable costs.

Value of Information

Let us now discuss what we understand by the term 'value' of information. To be of 'value' there are certain desirable characteristics or attributes of information, quantitative and descriptive.

Information must primarily possess attributes of relevance, availability and timeliness, to have value and thus to qualify as information. Objectivity, sensitivity, comparability, consciousness, and completeness are desirable and necessary only in varying degrees. No doubt quantifiability is desired to the maximum extent possible. The quality attribute refers to the presence or absence of ambiguities in information. No doubt all information should preferably possess 'quality'. Measures of quality are validity, accuracy and precision. These measures of quality are especially important and applicable to quantified information.

Quality is defined as excellence or fitness. It is not an absolute concept; it is defined within a context. An application has quality relative to its primary and secondary users, operations personnel, control personnel, maintenance personnel etc. Perfect quality is very costly and perhaps impossible. Since information is a critical organisational resource, low quality information has an adverse effect on organisational performance.

Even if information is presented in such a way as to be transmitted efficiently and interpreted correctly, it may not be used effectively. The quality of information is determined by how it motivates human action and contributes to effective decision-making. Information may be evaluated in terms of 'ultilities' which may facilitate or retard its use. These 'utilities' are explained below:

i) **Form Utility:** As the form of information more closely matches the requirements of the decision-maker, its value increases. If the manager is looking for a graph depicting the sales history, he appreciates receiving the data in the graphical form rather than in tabular form.

ii) **Time Utility:** Information has greater value to the decision-maker if it is available when needed. If you make it available well ahead of time, he might forget about it. Obviously any availability after the due time might have no relevance. Therefore, greatest value is at the appropriate time when the decision-maker needs it. The difficulty is that somethimes the appropriate time is not known to the decision-maker himself.

iii) **Place Utility (Accessibility):** Information has greater value if it can be accessed or delivered easily. On line systems maximise both time and place utility. It is very important to have the information available at the place desired.

iv) **Possession Utility (Organisation Location):** The possessor of information strongly affects its value by controlling its dissemination to others. Information is power or rather one who has information has the power.

Though it is difficult to quantify the contribution of the utility of information, one surrogate worth considering is **information satisfaction** viz. the degree to which the decision-maker is satisfied with the output of the formal information system.

Bias and Error

Managers or in fact anyone, would have strong bias towards quality rather than quantity of information. It is possible to estimate the biases of the decision-makers and provide him suitably adjusted information.

Here we would like to introduce the notion of errors in contrast to bias. Errors, a more serious problem, may result from:

a) Incorrect data measurement and collection methods

b) Failure to follow correct processing procedures

c) Loss or non-processing of data

d) Wrong recording of data

e) Incorrect history (master) file (or use of wrong history file)

f) Mistake in processing procedure

g) Deliberate falsification.

In most information systems, the receiver of information might have no knowledge of either bias or errors that may effect its quality. The difficulties with errors may be overcome by internal controls to detect errors, internal and external auditing, addition of 'confidence limits' to data and user instruction in measurement and processing procedures so that users can evaluate possible errors. The first two methods attempt to reduce the uncertainty about the data and therefore increase the information content. The last two remedies provide the user with confidence limits.

Value of Perfect Information

In the decision theoretic framework, there could be **decision-making under certainty** where we assume existence of perfect information regarding outcomes. In decision-making under uncertainty, we assume only a knowledge of possible outcomes but no information as to probabilities. In decision theory, the value of information is the value of the change in decision behaviour caused because of availability of information less the cost of obtaining the information. Given a set of possible decisions, a decision-maker will select one on the basis of the information at hand. However, if a new set of information input changes the decision, then the value of the new information is the difference in value between the outcome of the old decision and that of the new decision, less the cost of obtaining the new information. In this context, there is also the notion of the value of perfect information. The value of the perfect information can be computed as the difference between the 'optimal' policy without perfect information and the 'ideal optimal' policy with perfect information.

The concept of the value of perfect information is useful because it demonstrates how information has value as it influences or changes, decision. However, decisions are usually made without the right and appropriate information because the needed information is usually not available; the effort to acquire the information is too great or too costly; sometimes there is no knowledge of the availability of the information and often the information is not available in the form needed.

16.7 SUMMARY

In this unit we have discussed the concept of data and information. Over the years with the growing complexities and increasing competition, the pressures for information have stepped up manifold. There are various methods of processing data to give some type of information. A data life-cycle has also been explained. Data could be free, but information costs something. We have tried to explain the various costs of information. Now there must be some benefits, utility or value of information. We have discussed how the value or utility could increase depending on circumstances. It is quite difficult to qantify quite a few of the desirable characteristics of information. A brief introduction to the mathematical information theory in communication is also given. Information tries to reduce the uncertainty in a given situation. This unit is a pre-requistie to the next unit on management information systems as the concepts of desirable characteristics of information, would be particularly useful in understanding management information systems.

16.8 SELF-ASSESSMENT EXERCISES

1 List and explain the major reasons for ever increasing need for data processing.

2 What impact would (i) the technological revolution (ii) research and development (iii) product changes and (iv) the information explosions have on the need for information by management in a light commercial vehicle manufacturing company, a health care delivery system and an engineering college?

3 Define the terms data and information. Discuss the difference between data and information. Give examples.

4 Take any situation in an organisation and illustrate how a data life-cycle works in such a context.

5 List the attributes that add value to information and attempt to give a system of measurement for each attribute.

6 Find a user of information in any organisation of your choice and identify some information report that he or she receives. Ascertain or estimate the cost of this report and its value. Does the value exceed the cost?

16.9 FURTHER READINGS

Blumenthal, S., 1969. *Management Information Systems: A frame Work for Planning & Developing,* Prentice Hall: Englewood-Cliffs.

Davis, G.B. & M.H. Olson, 1985. *Management Information Systems: Conceptual Foundations, Structure & Development,* McGraw Hill: New York.

Kroeber, D.W. & H.J. Watson, 1984. *Computer based Information Systems, a Management approach,* Macmillan Pub. Co.: New York.

Kanter, Jerma, 1984. *Management Information Systems,* Prentice Hall of India: New Delhi.

Lucas Jr. H.C., 1982. *Information Systems Concepts for Management,* McGraw Hill: New York.

Murdick R.G. & R.E. Ross, *Information Systems for Modern Management,* Prentice Hall of India: New Delhi.

Senn, J.A., 1985. *Analysis and Design of Information Systems,* McGraw Hill Book Co.: New York.

Sanders D.H., 1984. *Computers in Business: An Introduction,* McGraw Hill: New York.

Srivastava S.S., *Management Information Systems & Computers,* Atma Ram & Sons: Delhi.

Tricker and Boland, 1982. *Management Information and Control Systems,* J. Wiley: London.

UNIT 17 MANAGEMENT INFORMATION AND CONTROL SYSTEMS

Objectives

After going through this unit, you should be able to:

- comprehend and conceptualise systems basically using a classification of input, processor, output, control and the environment
- appreciate and understand the management functions at various levels in the context of relationships between management and informational needs
- define clearly the concept of management information
- apply the systems approach to analysis of organisational problems.

Structure

17.1 INTRODUCTION

Information processing is a major societal activity. It has become an important resource in all walks of life today. Whether it is industry, commerce, defence, banking, education, economics or politics, information is needed everywhere. A significant part of an individual's working and personal time is spent on recording, searching for and absorbing information. As much as 70 to 80 per cent of a typical executive's time is spent in the processing and communication of information. Information is 'live' as it is required to be updated all the time, and it is renewable. Information is substitutable and transportable and can be made to travel nearly at the speed of light in a communication network. The growth of information in the last few decades has been fantastic. It used to be said that information gets doubled every ten years. But today this period of doubling seems to be just five years. The exponential growth of information all around makes it necessary that information is probably collected, stored and retrieved in various fields so that it could be usefully exploited where and when needed. In the previous unit, we have discussed in detail certain concepts about data and information. From information let us move on to management information. What information does the manager need to manage effectively? We are interested in a system for providing the necessary management information. Thus we need to conceptualise a management information system (MIS).

The MIS is a system that aids management in making, and implementing decisions. There is quite a lot of disparity and lack of precision in this regard as can be seen by scanning literature on MIS. Whether the MIS is computer based or non-computer based, it ought to focus on managerial effectiveness.

17.2 SYSTEMS VIEW

A Change that has occurred in recent years is the adoption of the so called 'systems approach'. In the past, managers, decision-makers and problem-solvers attempted piecemeal solutions, thinking in an isolated compartmentalised fashion independent of other operational units in the organisation. Today besides professional managers, political administrators have also become aware of the need for adopting an integrated holistic perspective by adopting the systems approach to problem-conceptualisation and decision-implementation.

Systems Concepts

Today we find everyone talking of systems-the transport system, educational system, healthcare delivery system, defence system, economic system, communication system, management information system, transaction processing system, decision support system, computer systems, etc. We are in the midst of an era of systems so to say. But what exactly do we mean by a system?

What is a System?

A system is an organised or complex whole. It is an entity, conceptual or physical, which consists of interdependent parts or components. It is this interdependency which is a characteristic of the parts of the system. It is an interlocking complex of processes characterised by many reciprocal cause effect pathways. A system is a complex of elements or components directly or indirectly related in a casual network. This brings in the notion of some type of feedback and control to see whether or not the system is in a position to achieve the goals/purpose/objectives of the system. Any system must have an objective or a set of objectives or a hierarchical set of objectives. In a large context, a system is an assembly of procedure, processes, methods, routines techniques etc. united by some form of regulated interaction to form an organised whole. In fact no system, unless it be a totally closed system, can exist in isolation.

A system is made up of sub-systems which may be composed of further sub-systems. We could carry on this refinement till we arrive at the so called 'black box' level which is some perceptible manageable level. Just as a system is made up of sub or sub-sub-systems, it itself is part of a super or supra system. This could be termed as the environment in which the system operates. The forces in the environment impinge on the system while the system itself exerts pressure outwardly on the environment thereby having some sort of a dynamic equilibrium at the boundary which separates the environment from the system.

We can graphically depict the above narrative description in the form of Figure I.

Let us give an illustration in the context of Figure I. Well, one could think of an industrial system or a factory system. A factory system has various sub-systems like the production sub-system, the financial sub-system, the marketing sub-system and the personnel sub-system. Now a production sub-system could consist of sub-sub-systems of production control, materials control, quality control etc. The materials sub-sub-system can be further broken down into 'black boxes' say purchasing, stores, transportation etc. In turn, the factory system is part of the larger economic system of the country which would be the so called superior or supra system.

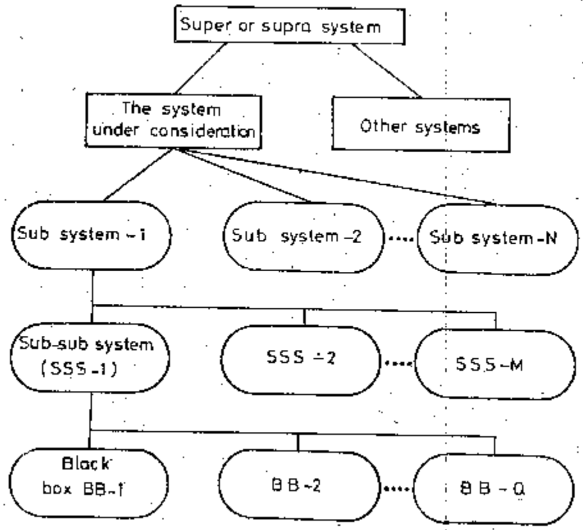


Figure I: Hierarchy of Systems

Activity A

Think of atleast three examples in the context of Figure I.

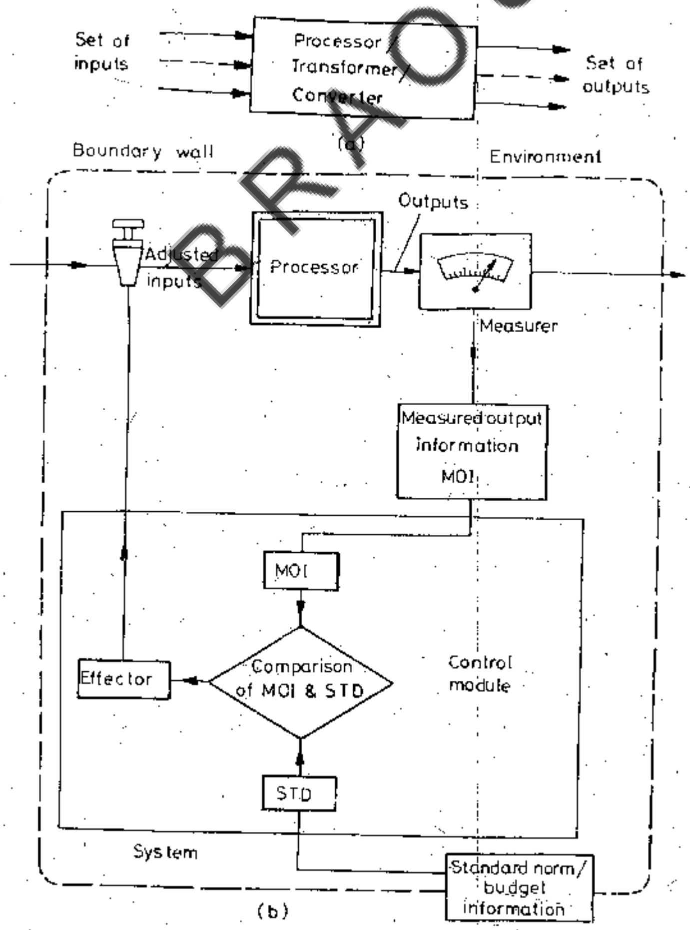


Figure II: Basic Systems Module

Further we could give yet another example in the context of Figure II. Data could be a set of input into a data processing system which would process/transform/convert the data into output or information. One could think of raw material entering as input into a production system which is converted/transformed or processed into an output i.e. some final product.

Activity B

In the context Figure II, give atleast three examples mentioning the inputs, the processor and the set outputs.

Input	Processor	Output
....	....	....
....	....	....
....	....	....

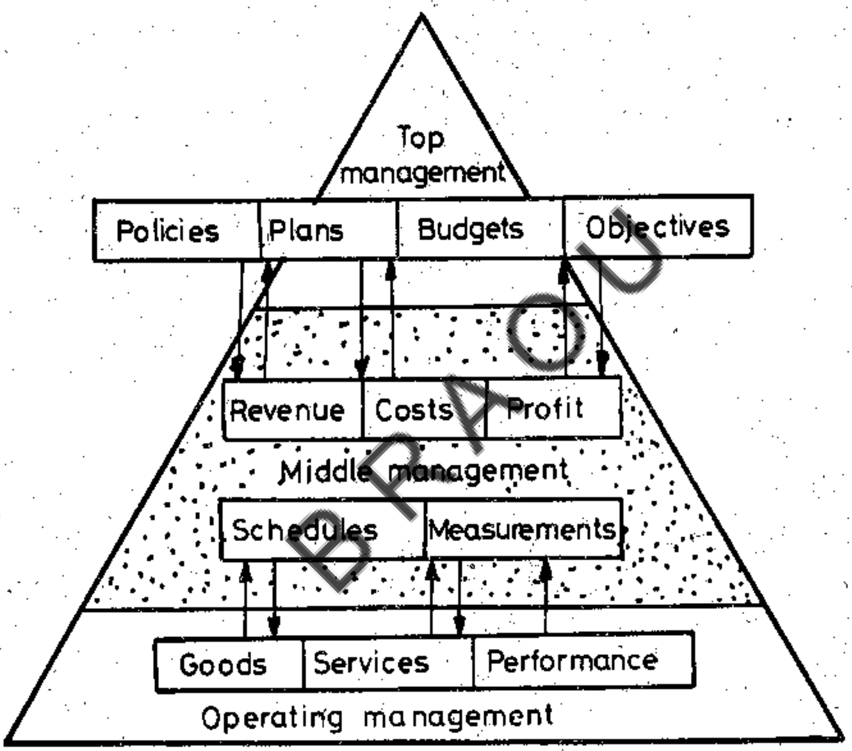


Figure III: Interaction of Management Levels

Continuing the example further in the context of Figure III this time, the quality of the finished product could be measured by comparing it with the standard specifications of the finished product. Depending on the deviations or variances the manager can then adjust the quality and quantity of the raw materials. The environment of the factory system under consideration could be other factories, competitors, customers, markets, sociopolitical and cultural factors, government etc.

17.3 ROLE OF MIS AT VARIOUS MANAGEMENT LEVELS

Some say that management can be understood by observing what managers do. Managers get the work done through others. Management can also be understood by the type of functions a manager performs. A manager usually performs the following functions: Planning, Organising, Staffing, Directing, Coordinating, Reporting and Budgeting. In fact management is a process of achieving an organisation's goals and objectives by judiciously making use of resources of men, materials, machines, money, methods, messages and moments (the last two in the context of information being a vital resources to the manager/decision-maker).

Management can also be seen as structured into three hierarchical levels namely, top level, middle level and bottom level or strategic, tactical and operational levels, respectively. Although lines of demarcation are not absolute and clear-cut, one can usually distinguish certain layers within the organisation which are characterised by the classical pyramidical type of structures as shown in Figure III. Top management establishes the policies, plans and objectives of the company as well as a budget framework under which the various departments will operate. These factors are promulgated and passed down to middle management. They are translated into specific revenue, cost and profit goals particularly if each department works under a cost or profit centre concept. These are reviewed, analysed and modified in accordance with the overall plans and policies until agreement is reached. Middle management then issues the specific schedules and measurement yardsticks to the operational management. The operational level has the responsibility of producing goods and services to meet the revenue, profit and other goals, which in turn will enable the organisation achieve its overall plans and objectives.

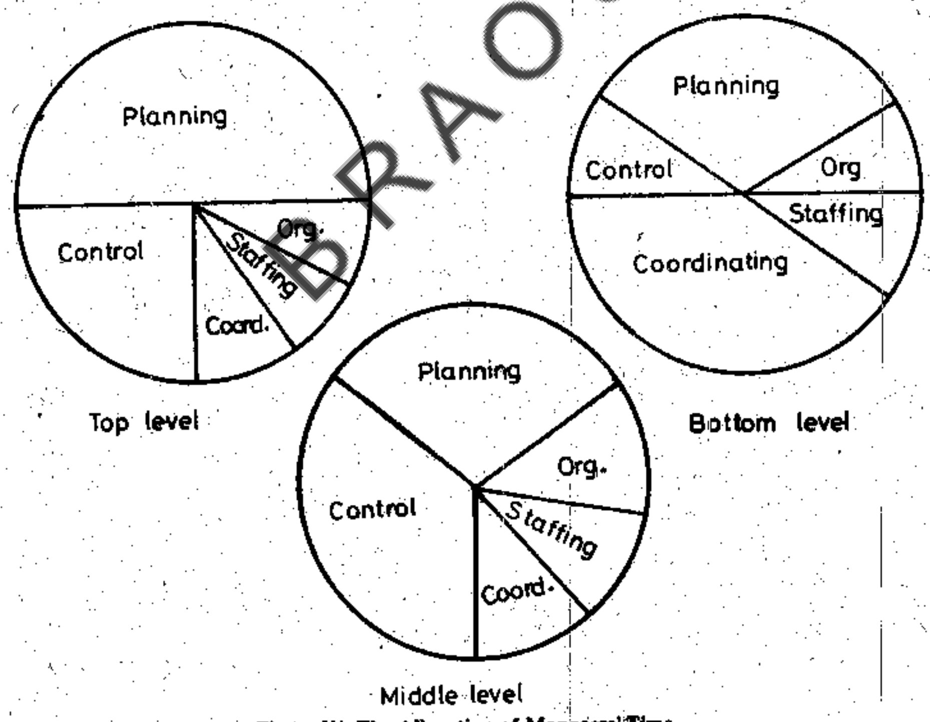


Figure IV: The Allocation of Managers' Time

The hierarchical view of management is important for two reasons: information needs tend to be different at different levels of management and the amount of time devoted to any given function varies considerably with the level as can be seen in Figure IV. The job content at various management levels is further elaborated in Table 1.

In the context of MIS, management can perhaps be best defined as a process of (i) selection of objectives (ii) judicious allocation of resources (iii) determining operational plans and schedules (iv) keeping control of progress and (v) evaluation through feedback. Each of these areas require certain decisions to be made.

Thus we take strategic decisions at the top level, tactical decisions at the middle and operational decisions at the junior level. As can be seen from Table 1, the type of problems and deicisions at the junior level are quite deterministic and structured, so we can have programmed decisions.

But as we move to higher level, situations become fuzzy, ambiguous and unstructured, and thus we are faced with non-programmed decisions. We find that with the introduction of computers, we have gone about routine EDP type of an activity for the essentially programmed decisions that take place at the operating level. Perhaps with the rapid advances that are taking place in the field of electronics, communication and computers, we might have good progress in the field of AI (Artificial intelligence) and accordingly devise knowledge based expert systems which would be helpful at the strategic level to cater to non-programmed complex type of decision-making situations.

Table 1

Job Content of Management Levels

Character		Top Management	Middle Management	Operating Management
1	Focus on planning	Heavy	Moderate	Minimum
2	Focus on Control	Moderate	Heavy	Heavy
3	Time Frame	One to Five Years	Up to a year	Day to day
4	Scope of activity	Broad	Entire functional area	Single sub-function or subtask
5	Nature of activity	Relatively unstructured	Moderately structured	Highly structured
6	Level of complexity	Very complex, many variables	Less complex, better defined variables	Straightforward
7	Job measurement	Difficult	Less difficult	Relatively easy
8	Result of activity	Plans, policies and strategies	Implementation schedules, performance Yardsticks	End product
9	Type of information utilised	External	Internal, reasonable accuracy	Internal, historical; high level of accuracy
10	Mental attributes	Creative innovative	Responsible, persuasive, administrative	Efficient, effective
11	Number of people involved	Few	Moderate number	Many
12	Department divisional interaction	Intra-division,	Intra-division, inter department	Intra department

Source: J.Kanters- "Management Information Systems". 3rd Ed., Prentice Hall: Englewood-Cliffs.

Though the classic pyramidical structure is generally acceptable, unfortunately in the modern complex organisation this neat, militaristic, configuration seldom (!) fits the reality. Under conditions facing modern management the strategy and the control tend to become more remote from the resources that are geographically spread and organisationally diverse. Between the decision-maker and the resources lie systems-of people and data handling equipment-that can distort, delay, amplify, and dampen messages. External to the enterprise, interest groups-in government, consumers, labour representatives, other national and international agencies are involved in an information exchange. The modern manager must be capable of managing his information systems for strategic planning, management control and operational control.

Let us now see what the structure of a management information system is.

17.4 STRUCTURE OF MIS

Few concepts have been more vague and misunderstood as the definition and scope of an MIS. Many experts think that an MIS is the computerisation of clerical work. This is incorrect. Others hope that some day it might be possible to have an all knowing expert computer system which will provide answers and decisions for complex problems when a manager/executive simply presses a few buttons. This view would perhaps remain a dream. An MIS is a means for connecting the managed operating systems by exchange of information. An MIS is more than a set of ideas or concepts. It is an operational system performing a variety of functions to produce outputs which are useful to the operating personnel and management of an organisation. Managers have always had 'sources' of information, the MIS provides a system of information. It is imperative to realise that a systems approach to managing is necessary to complete in today's world. The systems approach to management must precede the design and use of an MIS. The computer is only a compenent, or a tool of the MIS, not the MIS itself or the central focus of MIS. Management must take an active part in the design of the MIS. Participatory design process is advocated. Technical knowledge of the computer, though preferable, is not necessary for the manager to perform his role in the design of MIS. The essence of MIS is integrated planned systems are and not "the islands of mechanisation" or data processing systems.

Davis and Olson define MIS as an integrated user machine system for providing information to support operations, management and decision-making functions in an organisation. The system, if it is computer based, utilises computer hardware and software, manual procedures, decision models and preferably a data base. A data base is a centrally controlled integrated collection of logically organised data. The underlying concept of a data (bank) base is that data needs to be managed in order to be available for processing and have appropriate quality and value. Over time, the concept of a single highly integrated system was demonstrated to be too complex to implement. The MIS concept is now veering around to that of a federation of sub-systems developed and implemented as needed but conforming to the overall plan. Thus rather than a single global MIS, an organisation may have many related information systems which serve managerial needs in various ways.

MIS is an organised method of providing past, present and projected information relating to internal operations of an organisation and external intelligence by good environmental scanning techniques. All organisations have some kind of information system even though some systems might be nothing more than filing cabinets and an accounts ledger. An information system should have a systematic, formal assemblage of components that performs data processing operations to (a) meet the legal and transactional data processing requirements (b) provide information to managers at all levels for carrying out their functions effectively and (c) provide a variety of useful reports, as required to internal and external constituents. If a report does not have an element of utility, why generate it or ask for it, unless it is a statutory requirement. Since the facts are too many and keeping track of them means getting bogged down in day-to-day routine matters, the managers do not have sufficient time for creative and innovative work and for decision-making of a strategic nature. It is, therefore, imperative to ask for 'exception' reports. Perhaps it is well to introduce a brainstorming session of executives involved at different levels to specify what they are looking for, what is to be done, by whom, by what data and in what form. It would be worth creating confidence and mutual trust about not reporting normal things. Perhaps some type of a selective information management could be pursued somewhat similar to the selective inventory management of value based ABC analysis and critically based VED (Vital, Essential, Desirable) form.

17.5 INFORMATION NETWORK

Invariably we find information flowing from one place to another, from one decision-maker to another in an organisation. Perhaps it is this phenomenon which motivated Forrester to conceptualise an organisation as an **Information Network.** He observes that enterprises are very complex multiloop interconnected sytems. Decisions are made at multiple points throughout the system at various hierarchical levels. Each resulting action generates information that may be used at several, though not necessarily at all decision points. This structure of cascaded and interconnected information feedback loops, when taken together, describe the total system. The interlocking network of information channels emerges at various points to control physical processes such as the hiring of employees, the building of factories, and the production of goods and/or services. Every action point in the system is backed up by a local decision point whose information sources reach into other parts of the organisation and the surrounding environment. Management is a process of converting information into action-a process analogous to decision-making. Forrester conceptualises six information feedback networks in an industrial setting namely, materials, orders, money, personnel, capital equipment and information. A policy is a rule that states how the day-to-day operating decisions are made. Decisions are the result of applying the policy rules to the particular conditions that prevail at any moment.

In essence, therefore, there is essentially an **activity centre** where certain actions or activities take place which change the level or state of the system. These activities are carried out because of higher level direction received from the managers at the **decision centres** which have their own set of decision procedures or norms. The combination of an activity centre plus decision centre is termed a **functional unit** because it performs a function. The functional unit is represented in Figure V, which shows the flow of information. Hence the information system could be designed to provide information to each functional unit, in fact at each of the strategic planning level, management control level and operational level as shown in Figure VI. The information system for related functional units can be clustered into an information sub-system. An **operational function** is a class of any one or more types of actions, carried on by the same or different functional units which regulate the inflow and/or outflow to or from sequence of levels as a group. A **management control** centre is one or more management people together with their supporting staff which acts as a decision centre for a group of functional units or for a group of subordinate management control centres.

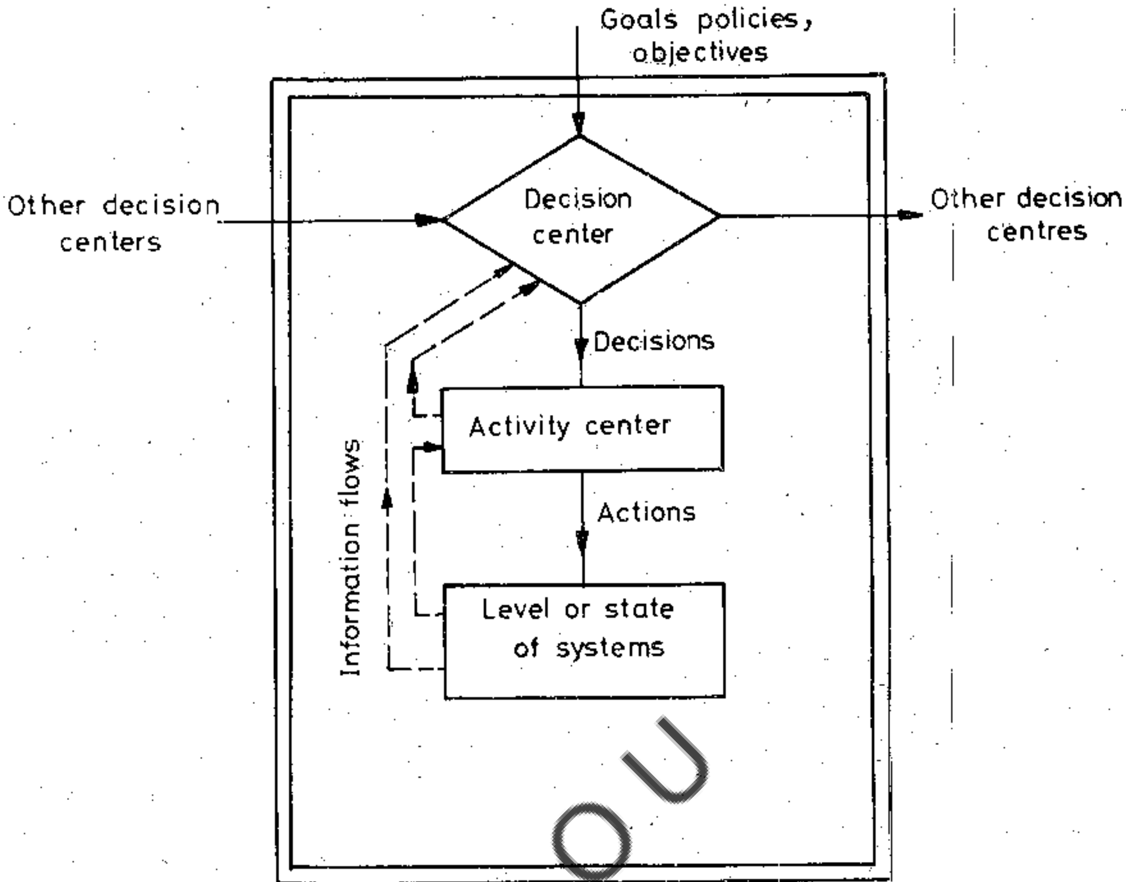


Figure V: Flow of Information in a Functional Unit

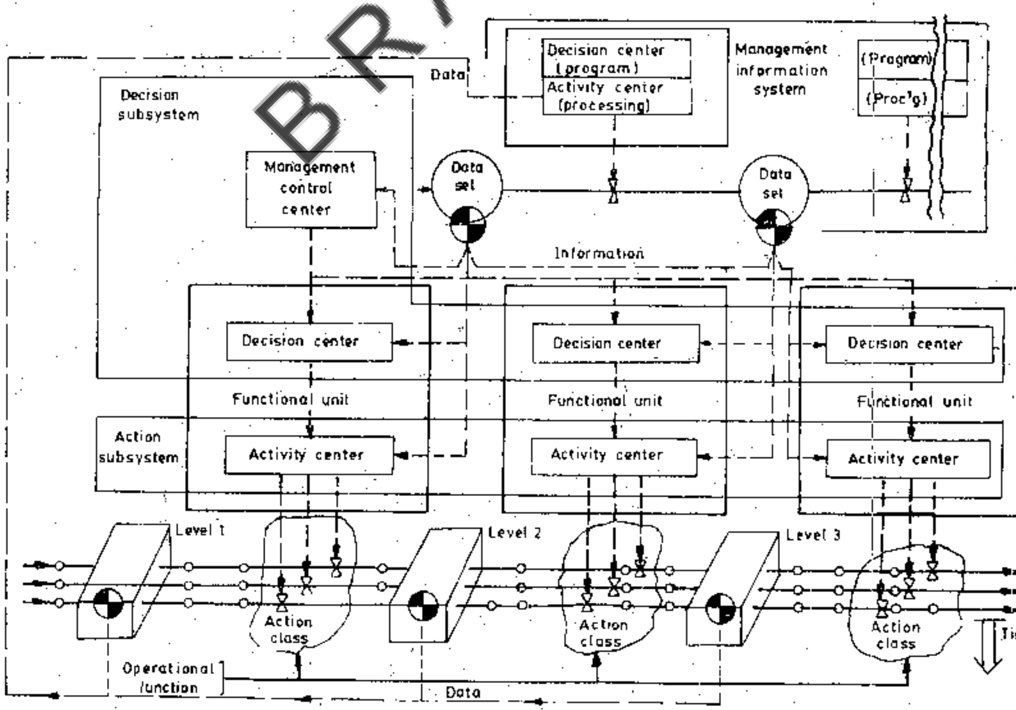


Figure VI: Schematic Representation of Management Information System(Source S.C. Bluementhal Management Information Systems, Prentic Hall Int. 1969)

An information sub-system is a special operational function unit and MIS is an operational function whose parts (corresponding to functional units) are information sub-systems of other operation functions. We could also think of an Operational Control Module as that part of an information sub-system which supports the functional units of an operational function. A Management control module is that part of an information sub-system which supports the management control centres of an operational function. Parts of information sub-systems, termed modules, are the basic entities which help in adopting a modular approach to an MIS design. The use of modules is desirable because it allows improved project control. The modules can be written and tested separately allowing more efficient planning and implementation in phases.

The Non-formal Element

An MIS model has to reflect formal and informal flow of communication of messages/data/information. At the bottom is the physical system-the workers and all equipment and facilities used to produce the products and/or services. Internal data generated here is passed on to the information processing resources (which could be manual or computer based, preferably making use of a data base) which generate internal information formally. The internal and external environmental information is passed on to the appropriate executives/managers who also acquire on their own internal and external information on an informal, non-formal manner. Gordon Davis has conceptualised the various MIS components chart as shown in Figure VII.

Part of the MIS is available for use by anyone in the organisation, and part is private, restricted to only the person establishing it. The computer resource (in case of computer based MIS) is a part of the public MIS, whereas the information that an executive receives from telephone calls, letters and memos sent on a personal basis are examples of the private MIS. Likewise, part of the MIS is formal and part is informal. The formal MIS is prescribed by procedure such as computer programme. The informal MIS has no spelled-out routine. The size of the four boxes in Figure VII are what it ought to be like-the largest being public-formal and the smallest

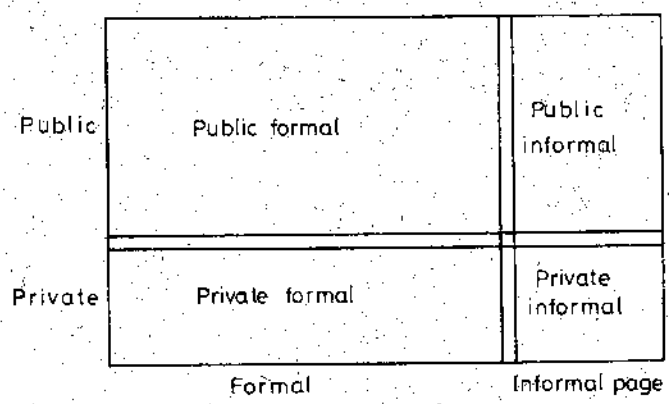


Figure VII: Davis' MIS Formal & Informal Components

being private-informal (which in an ideal case ought to be at the zero level which perhaps will never occur). Attention must be directed also to the informal organisation structure. Management should try to identify information needs that are not being fulfilled by the formal system and to incorporate as many of these flows into the formal MIS as possible. Certainly those informal power centres of the organisation that can influence the sucess of the MIS should be identified and included in the design effort. One often wonders whether an MIS should straightaway be superimposed on the existing organisation structure.

17.6 DESIRABLE CHARACTERISTICS OF MIS

It would be proper if we recall the desired characteristics of information i.e. accuracy, timeliness, objectivity, relevance, conciseness etc. You will discover that some of these characteristics are also the desirable characteristics of a good, effective and efficient MIS. The process of developing an MIS is never ending, as organisations strive to take advantage of **new technology and methodology.** The evolutionary process followed in achieving an MIS is called the MIS life-cycle which consists of phases in a sequence of planning, analysis and design, implementation, operation and control. The manager is ultimately responsible both for developing and using it. The information specialists ought to serve as valuable technical assistants. No doubt, the information specialists play a vital role in the development of MIS. They often trigger the manager's interest in a new system by informing the manager of a new technology or method. Though the specialists recommend a particular system design, it is the manager's responsibility to approve its implementation. Once the manager makes the decision, it is the information specialist' task to implement the system. The MIS would have to preferably cater to the management/leadership styles also. If there could be greater user-involvement at all stages of the MIS life-cycle, then the end results would tend to be superior.

The system should help each executive in his decision-making process for problem identification, generation and evaluation of alternative courses of action, to acquire necessary feedback on implementing his decision and help him to take corrective action. The MIS should develop the much needed management informaiton rather than facts. It should provide relevant data in a summarised form to the higher echelons viz. the system should be integrated through a centralised data base to cut down on redundancies, overlaps and costs. Apart from the appropriateness of information at different levels, it is important to recognise that different types of information are required for planning, control, and other managerial functions. It might become necessary to computerise your system especially if you are dealing with a large complex organisation or if you find that the existing manual based MIS is not in a position to provide you with timely information. Make sure that the reports churned out from the MIS are relevant and meaningful. They should be easy to understand and read with good documentation. It is a good practice to distribute the reports only to those that genuinely need and make use of it. The MIS should be robust but at the same time it should be flexible and sensitive enough to cater to relevant charges in technologies, methodologies and the changed parameters. There should be some type of a review undertaken periodically to assess the worth of the MIS.

17.7 SUMMARY

In this unit we have tried to conceptualise management information and control systems. The concepts of data and information having already been outlined in the previous unit, systems concepts, and systems approach were discussed in this unit. The systems approach advocates an integrated, holistic, systematic and scientific approach. Next certian concepts of management were visited again, but discussed in the context of MIS. A manager is primarily a decision-maker and a problem-solver. Information is one of the prime resource inputs to his decision-making. It becomes necessary to provide the right type of information to the right decision-maker at the right time. Thus the need for an effective system to provide information to management at all levels. The whole organisation could be thought of as an information network, both formal and informal, connecting various decision centres at various levels. Managers at different levels perform different managerial functions and hence require different types of information. A conceptual framework outlining the desirable features of MIS have been described which could lead you towards the design of an MIS, if you do a little more reading on the subject.

17.8 KEY WORDS

Analysis: The methodical study of a problem. Especially the modularising of a problem into smaller and simpler problems. Usually used in a nonbusiness sense (such as numerical analysis, operations analysis) unless combined with a business term.

Audit: Examination and verification of records.

Binary Digit: Either of the digits 0 or 1. A single digit of the binary number system. Bit is the contraction of binary digit.

Business: The purchase and sale of goods in order to make a profit. As used in this block, business encompasses the professions and industry, any trade of goods or service, manufacture or other activity engaged in for profit. Any institution or group of people engaged in such activity.

Business System: A system used to accomplish some business goal. Typical business system include accounts receivable systems, payroll systems, banking systems and so on.

Central Processing Unit: Directs the operation of other units in a computer and performs the data manipulation and computations needed to effect the desired transformation of the input data. Frequently written CPU. When the word computer is used to mean less than a complete computing system, it generally denotes the CPU. In computers referred to as main frame.

Communication System: Used to convey data from one point to another. Consists of a message source, message channel and message receiver. Also called data transmission sytem data communication system and the like.

Components: A part of system. An element of a larger thing. See system and sub-system. A component is generally a sub-system unless it is too simple to be considered a system at all.

Cycle: A sequence of operations or activities that are repeated regularly (though the operations may be varied from repetition to repetition.)

Data: Some representation of facts, instructions, or concepts. Commonly represented in symbols but in computerised buincss systems, it is generally represented with graphics.

Data Base: Some number of files. Most data bases have names (corporate base, financial data base, production data base and so on).

Implementation: The step in the system life-cycle where a system design is constructed as a real business system.

Information: The meaning humans assign to data. The purpose of an information systems is the production of information from data.

Installation: The step in the system life-cycle where a new business system is put into use. Also used to mean a business or computing facility.

Management by Exception: A principle of management where attention is focused only on performance. The phrase used in connection with the presumed action required where reports on performance differ from original projections. Characteristically these reports bring out 'favourable' and 'unfavourable' variances, the latter being the basis for 'Management by exception'.

Management Information System: A business application that collects, maintains, correlates, and selectively displays information to assist human being in decision-making. Generally denoted by the acronym MIS.

Message: The information to be transmissted by a communication system.

Microprogramming: Using a programme to specify the inter-relationship among the subsystems of a computer when the computer is being viewed at the architectural level. These programs and the hardware they control are often referred to as firmware. Microprogramming permits redefinition of a computer's hardware, machine language instructions without making any wiring changes.

Module: A part of a whole.

Organisation: Either (1) an institution, body of people organised for some purpose or the administrative personnel of a business or (2) synonymous with structure.

Storage: A mechanism into which data can be entered, in which it can be held, and from which it can be retrieved at a later time. Loosely any mechanism that can store data.

System: Some number of components that interact in such a manner that the whole is more than the sum of its parts.

Systems Analysis: The step in the system life-cycle were a detailed study is made of the need and its solutions and the environment in which they must exist.

Systems Life-Cycle: The sequence of steps through which a typical business system passes; need, conception, feasibility study, system analysis, design, implementation testing installation or conversion, production maintenance and cessation. This cycle has also been stated as

1) need,
2) conception,
3) wild enthusiasm,
4) total confusion,
5) search for the guilty,
6) punishment of the innocent,
7) promotion of nonparticipants, and
8) need.

17.9 SELF-ASSESSMENT EXERCISES

1 Discuss fully the purpose of an information system.

2 Discuss why management needs information. Is it possible for the management of an organisation to make effective decisions without the aid of an information sytem? Discuss.

3 "In the end, the informaiton system must be recognised only as a foundation for human judgement, insight and inventiveness". Discuss.

4 What impact does the implementation of programmed-decision-making have on the management system of an organisation?

5 "Formal information can't do much to help top level decision-makers". Discuss.

6 Sketch and explain a schematic diagram of a system with its management constituents, input relation to the environment and its information system. Choose any organisational system with which you are familiar.

7 Compare and contrast formal and informal information. Give five examples of each.

8 Visit an organisation of your choice and describe its information system. Is it based on the hierarchical or systems approach, or a variation thereof?

17.10 FURTHER READINGS

Burchand Strater, 1974. *Information Systems, Theory and Practice,* Hamilton Publishing Co.: New York.

Blumenthal, S., 1969. *Management Information System: A Framework for Planning & Development,* Prentice Hall Inc: Englewood-Cliffs.

Devis, G.B. & M.H. Olson, 1985. *Management Information System: Conceptual Foundations, Structure & Development,* McGraw Hill Inc: New York.

Kroeber, D.W. & H.J. Watson, 1984. *Computer based Information System, A Management Approach,* Macmillan Pub. Co.: New York.

Kanter, Jerma, 1984. *Management Information Systems,* 3rd Ed. Prentice Hall of India: Delhi.

Lucas Jr. H.C. 1982. *Information Systems Concepts for Management,* McGraw Hill: New York.

Matthews, D.R., 1971. *The Design of Management Information Systems,* AUERBA.

Senn, J.A. 1985. *Analysis and Design of Information Systems,* McGraw Hill Book Co.: New York.

Sander, D.H., 1984. *Computers in Business: An Introduction,* McGraw Hill: New York.

Tricker and Boland, 1982. *Management Information and Control Systems,* J. Wiley: London.

Trickers, R.I., 1984. *Effective Information Management,* Beaumount Executive Press: New York.

BLOCK 6 SYSTEMS ANALYSIS AND COMPUTER LANGUAGES

This block comprises three units. Unit 18 deals with the design and development of Computer Based Information System in an organisation, discussing extensively the System Life Cycle.

The second unit (19) gives the basic insight into computer programming with the help of programming illustrations, discussing higher languages and preparation of flow diagrams before writing programmes.

The third and last unit (20) specifically deals extensively with COBOL programming, explaining in details all the steps and other specifications involved in it.

UNIT 18 SYSTEMS ANALYSIS AND DISIGN

Objectives

After going through this unit you should be able to:

- understand the design and development of Computer Based Information System (CBIS) in an organisation
- know about the various aspects and components of System Life Cycle in a CBIS
- have an idea about the importance of flow diagrams in designing CBIS.

Structure

18.1 INTRODUCTION

Systems analysis and design provides a structured approach to the design and development of Computer Based Information System (CBIS). The tasks involved in the design and development of CBIS can be broadly classified into three categories, namely (i) Systems Analysis, (ii) Logical System Design, and (iii) Physical System Design. A Logical Design provides a logic for obtaining the desired outputs from available inputs. A Physical design maps the logical design onto the physical hardware of a computer system.

A Computer Based Information System (CBIS) is a set of software packages, which when executed, provides information for decision-making. Designing and developing a CBIS is one of the most important activities in any organisation, as it involves people at different levels in the organisation. Like any living organism, a CBIS also has a life cycle.

18.2 SYSTEM LIFE CYCLE

The various stages in the life cycle of a CBIS are described below:

Objectives: The first and the most important step is to provide a broad statement of organisational objectives, for the proposed CBIS. This is primarily a responsibility of the top management.

Feasibility: The next step is to examine the feasibility of the proposed system. This involves evaluating the costs and benefits of the system. Initially a rough cost-benefit analysis will be sufficient for the top management to take a decision either in favour of or against the proposed CBIS.

Costs include costs of design, development, implementation and maintenance of the system. Benefits will be realised from the timely and accurate generation of required information to meet the stated objectives of the organisation. It is to be realised that a data base created for a particular CBIS application usually serves other applications as well to a certain extent. For example, a data base for pay roll accounting can be used for applications such as Provident Fund (PF) accounting, Retirement Benefits accounting, Personnel Information Systems etc. Such indirect benefits also should be considered in the feasibility analysis.

Feasibility study is a part of Systems Analysis. Experience has shown that delaying or neglecting feasibility study is one of the major reasons for the failure of a CBIS.

Systems Analysis: Systems 'Analysis is the next stage in the life cycle of a CBIS. This step involves the following:

- understanding the organisation to identify the flow of information between different levels in the organisation
- a detailed examination of the proposed system (application area) for CBIS
- identifying alternative approaches to meet the stated objectives of the system
- evaluating the costs and benefits of each alternative in detail, and finally
- choosing the 'most appropriate' alternative.

Systems Analysis involves team work and a considerable amount of time. A clearer picture of costs and benefits of alternative approaches will emerge from a detailed study of the proposed system. Involvement of ultimate users of CBIS is very crucial at this stage itself, so that the acceptance of CBIS upon implementation will be relatively easy.

System Design (Logical): Providing Logical System Design for the chosen alternative involves:

- understanding user requirements
- identifying data requirements
- suggesting a logical organisation of data, and
- suggesting a logical procedure to produce the desired outputs from available inputs.

It is very important that we understand the requirements of various users in the organisation. Users at different levels in an organisation have different information requirements for decision-making. Information requirement can be broadly classified into three groups as follows:

- monitoring and control decisions
- planning decisions, and
- policy and strategy decisions.

Detailed discussions between system designers and ultimate users is very essential to estimate users' requirements clearly. This includes identifying report contents, frequency of reporting, formating of reports and presentation of reports (tabular vs. pictorial), for each user.

After estimating users' requirements, a system designer works backwards to identify data requirements. This includes identifying data sources, the nature and type of data that is available, and data gaps.

The next step is to establish a processing logic to produce the desired outputs from the available inputs. This step involves **data flow analysis** and a **data processing anlaysis.** Data flow analysis helps us to arrive at a logical organisation of data into **computer files.** A file is a collection of similar **records,** each record has a number of **data items** (fields) of informaion of a particular entity. For example, a payroll file will contain many payroll records, each record carrying the necessary data items (fields) of information of an employee, like name, designation, department, basic salary, etc.

A logical representation of **data flow analysis** and **data processing analysis** in a CBIS can be effectively provided through structured system design tools. These are:

- data flow diagrams
- data dictionary for data flow analysis
- decision tables
- decision trees, and
- structured English for data processing analysis.

These tools are highly recommended to present a well documented and self explained logical system design. These tools are explained in detail later.

System Design (Physical): While a logical design provides an estimate of processing requirements, a physical design involves mapping the logical design onto the physical hardware of a computer system. Upgrading the existing hardware and/or acquiring a new computer system if required to meet the processing requirements is also undertaken at this stage.

Subsequently, file organisation details are worked out and appropriate file organisation methods established for processing and storing data. File organisation methods can be broadly classified into two types - serial access organisation and random access organisation. These methods are discussed in detail in another unit.

A system flow chart is commonly used to present the physical system design. It is to be realised that a system flow chart is different from a programme flow chart. A detailed discussion on system flow chart is given later.

Implementation: Actual programming is undertaken at this stage to implement the proposed CBIS in the available hardware. This activity includes:

- programme development
- debugging of programmes: i.e. spotting and correcting errors in programming
- testing of individual programme modules on sample data
- implementing the entire system on the computer
- getting acceptance of the CBIS from its users
- providing user training, and
- preparing user manuals and documentations.

This is the most time consuming activity in the life cycle of a CBIS; and is also the costliest activity. Systems Analysts have a tendency to lose interest once all the programmes are developed and tested to their own satisfaction, and not necessarily to the satisfaction of the users. This tendency is dangerous and should be avoided at all cost. It is the responsibility of systems analysts to properly document their programmes, prepare user manuals, provide user training and getting acceptance for the CBIS from its users.

Maintenance: Maintenance follows a successful implementation of the CBIS. As users develop faith in a CBIS, their demands on the system will grow. The system design should be flexible enough to accommodate future requests, refinements, modifications, and changes to suit users' requirements. Well documented logical and physical designs of a CBIS will facilitate its maintenance considerably.

The following table summarises the various activities in the development of a CBIS.

Table 1: System Life Cycle

Stage	Description
Objectives	Broad statement of objectives
Feasibility	Rough cost/benefit analysis
System Analysis	Evaluating alternatives and choosing the most appropriate alternative to meet the objectives
Systems Design (Logical)	A broad logical design structured tools like DFD, Data Dictionary, Decision Tree, Decision Table, Structured English etc. and arrive at system requirements.
System Design (Physical)	Detailed system design, taking into account available and proposed hardware - use of system flow chart.
Implementation	Programming, implementing and user documentations
Maintenance	Modifications and revision for user needs.

18.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation to depict the flow of data in a CBIS. It should be realised that a DFD provides only **logical** data flow and **not** a physical data flow.

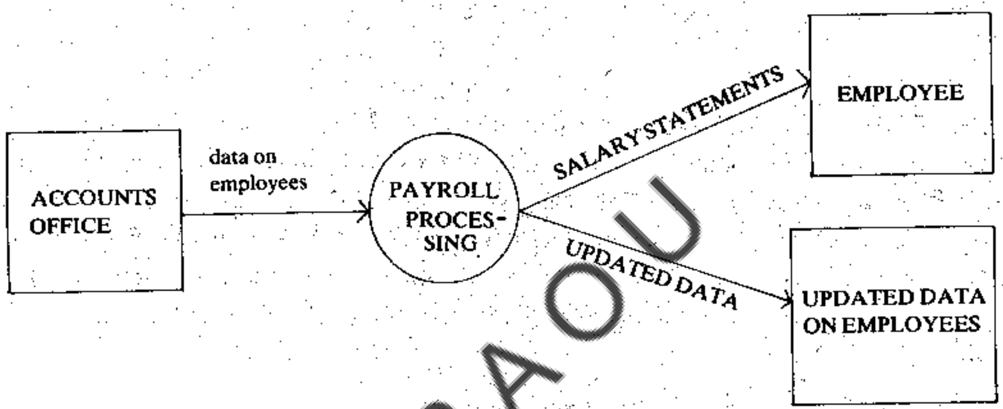
A logical data flow in a CBIS can be explained as follows: Data orginates from a **Source,** undergoes some **Processing** and terminates in a **Sink.** The processing step may require data stored elsewhere in **Data Stores,** over and above what is supplied by the source. Similarly the output of processing may be an intermediate data store which is used for subsequent processing.

These components of a DFD are presented graphically, using the following conventions:

- a closed square box to denote source/sink
- an open rectangular box to denote intermediate data store
- a circle to denote processing, and
- an arrow to denote the directional flow of data.

For example, consider the case of a 'pay roll accounting' system, to prepare salary statements for each employee of an organisation. Data flow for such a system can be represented as shown below in Figure I.

Figure I : A DFD for Payroll Processing



In Figure I, data on employees originate from accounts office (source), gets processed,salary statements go to employees (sink), and updated data on employees (e.g. total tax deducted, total contributions to provident fund, etc.) is stored in an intermediate computer file (data store) which is needed for subsequent processing next month.

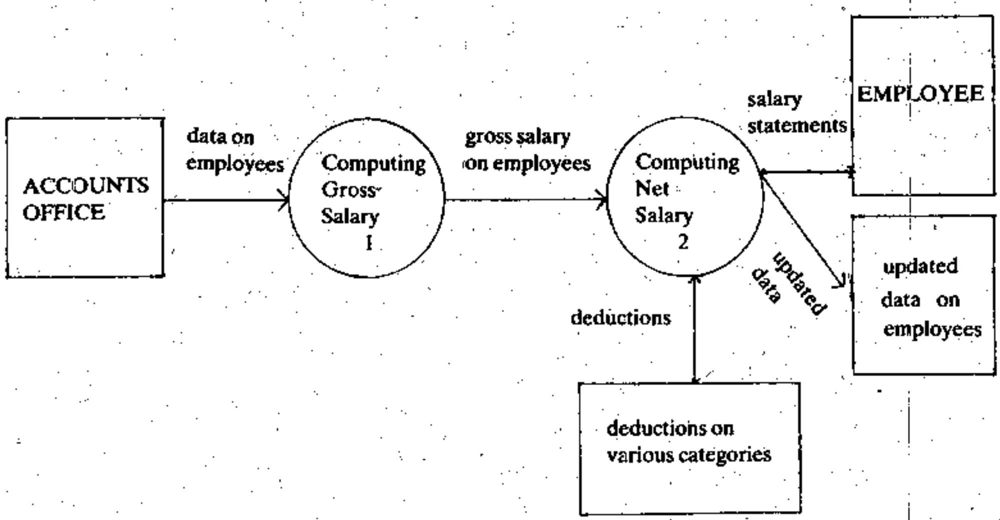
A DFD displays data flow in a **top-down approach.** Therefore we start with a macro DFD and 'explode it' into micro DFDs. Care has to be exercised to provide clarity for each level of DFD. Towards this, a standard practice is

a) not to use more than one page for one DFD, and

b) not to show more than 6 or 7 circles (processing steps) in each DFD.

The following Figure illustrates this practice.

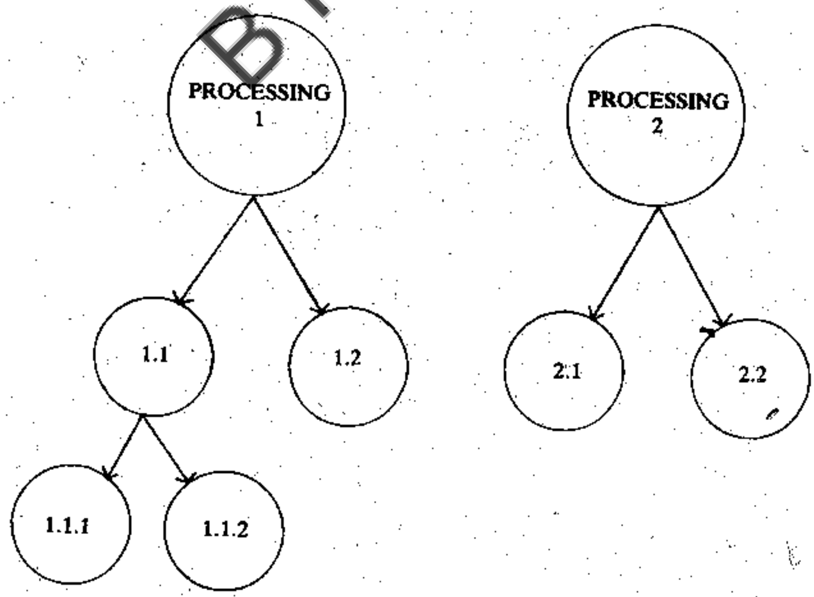
Figure II : A DFD Payroll Accounting



While exploding a DFD into lower levels, it is very essential to maintain a continuity and linkage between a DFD and its member DFDs. This is achieved by numbering each circle (processing step) by adopting the numbering system used in text books. In a text book, chapters are numbered 1,2,3,............... sections within a chapter have numbers as extensions of the chapter number, e.g. 1.1, 1.2, 1.3,.... and sub-sections within each section numbered as 1.1.1, 1.1.2, 1.1.3, etc. Similarly circles in DFDs are numbered as 1,2,3..... 1.1, 1.2, 1.3,....., 1.1.1, 1.1.2,......

The following Figure illustrates this numbering convention.

Figure III: Numbering Convention for Processing Steps in DFD.



Such a hierarchical approach to drawing DFDs provides a clear logic for system design, a proper documentation of processing steps and an estimate of processing requirement.

18.4 DATA DICTIONARY

A Data Dictionary (DD) is intended to provide a complete documentation of all the elements of a Data Flow Diagram, namely data items, data stores, and data flows. Data described in a DD carries the following details.

Data type	Data item/data store/data flow
Data name	Name of data item/data store/data flow
Data aliases	Alternate names used for convenience by multiple users.
Data description	A short description of data, explained in simple terms.
Data characteristics	Characteristics of each data type.
	A data item is characterised by its type (numeric/ alphanumeric), width, etc.
	A data store is characterised by its composition (set of data items), organisation (sequential, random), etc.
	A data flow is characterised by its origin, destination, etc.

Below we describe the contents of a DD for a sample data item, data store and data flow for a CBIS on payroll accounting.

DATA ITEM

- Data type — Data item
- Data name — G-SALARY
- Data aliases — Wages
- Data description — Monthly gross salary of an employee

Data characteristics

Type	Numerical
Width	7.2
	4 digits for the rupee component
	1 digit for the decimal point
	2 digits for the paise component
Associated data stores	Payroll file
	Personnel file
Associated data processes	Payroll file
	PF accounting
	Personnel information systems
Comments	Gross salary is based on employees designation, and hence falls within a specified range.

DATA STORE

- Data type — Data store
- Data name — Payroll file
- Data aliases — Salary file
- Data description — Master file on employees for payroll accounting

Data characteristics

Composition	EMP-NAME
	Designation
	B-Salary
	Department
	:
	:
	G-Salary
	:
	:
Organisation	Sequential file
Volume	1000 records (approximately)
Size	350 K bytes (approximately) per record
Associated data	Payroll accounting
Processes	PF accounting
Inbound data flow	__________
Outbound data flow	__________
Comments	This file gets updated every month at the time of pay roll processing. On an average, about 5 records are deleted per month (retiring/leaving the organisation) and about 10 records are added per month (new appointments).

DATA FLOW

● Data type	Data flow
● Data name	DATA ON EMPLOYEES
● Data aliases	__________
● Data description	Data on employees required for payroll processing
Data characteristics	
Origin	Accounts office
Destination	Process 1 in payroll accounting
Contents	EMP-NAME
	Designation
	B-Salary
	Department
Associated data processes	Payroll accounting
Associated data stores	__________
Comments	__________

18.5 DECISION TABLES

A decision table displays and documents clearly the processing logic for each processing step identified in a DFD.

A decision table has the following structure.

	Combination of conditions	
Condition 1	Y	
Condition 2		• • • •
Condition 3	Y	
Action 1		
Action 2	Y	• • • •

This table is to be read from top to bottom, one column at a time. For example interpret column 1 as follows: If conditions 1 and 3 are satisfied, then take action 2.

For illustration, consider an 'accounts receivable' application in ABC Corporation. Customers are prioritised based on how much they owe to ABC Corporation and for what length of time. The following procedure is used to classify customers into A,B,C categories. A customer is a A type customer if the amount due from him is not for more than one month. A customer is C type customer if he owes at least Rs. 6,000 for a period exceeding 2 months or if he owes less than Rs. 6,000 for a period exceeding 3 months. All other customers are classified as B type customers.

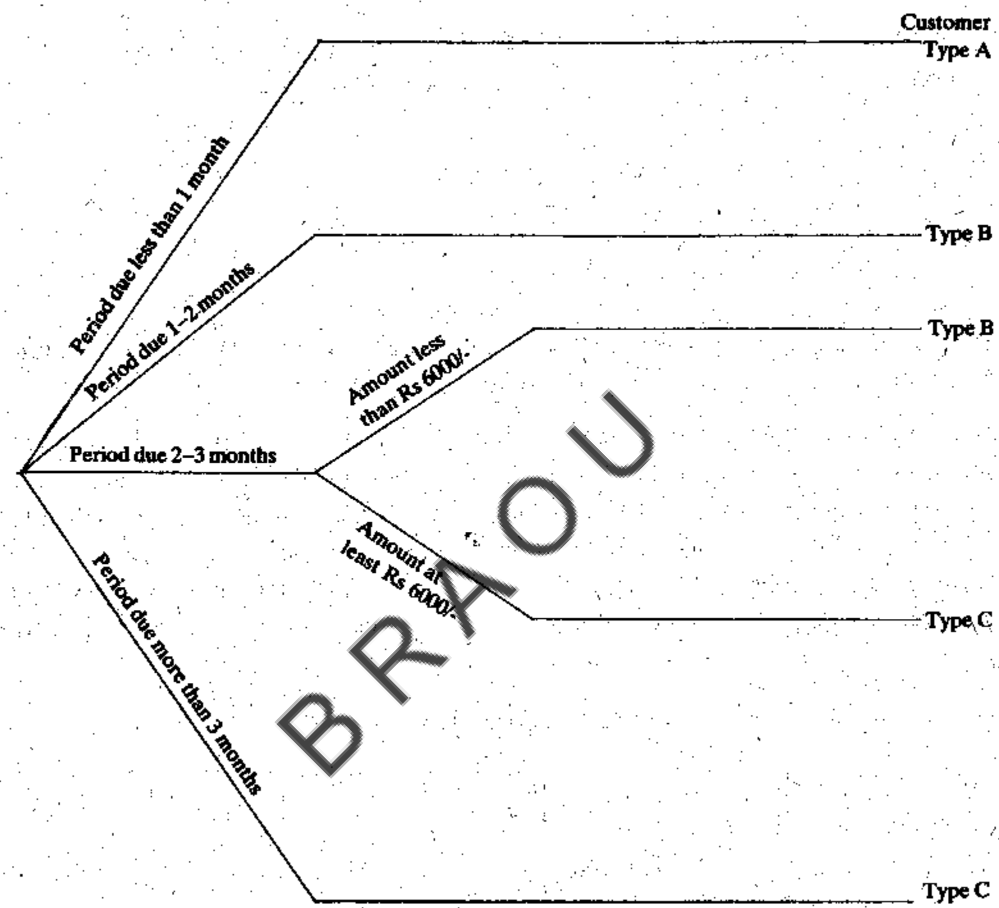
This logic can be represented in a decision table as shown below:

Conditions	Period due : less than 1 month	Y	Y						
	1-2 months			Y	Y				
	2-3 months					Y	Y		
	more than 3 months							Y	Y
	Amount due : less than Rs. 6000	Y		Y		Y		Y	
	At least Rs. 6000		Y		Y		Y		Y
Actions	Actions : Type A Customer	Y	Y						
	Type B Customer			Y	Y	Y			
	Type C Customer						Y	Y	Y

18.6 DECISION TREES

A decision tree is another way to document processing logic, specially when the number of alternatives is not too many.

Combinations of conditions are represented along the branches of a decision tree. The outcome of each of these combinations is given at the end of the final branch. For illustration, the processing logic followed by ABC Corporation for prioritising its customers (mentioned in section 4) can be explained in a decision tree as follows:



18.7 STRUCTURED ENGLISH

Processing logic can also be explained clearly in a structured English language. For illustration, the processing logic for ABC Corporation can be explained as follows:

IF Period-due is less than 1 month, then customer = A type.
ELSE
    IF period-due is less than 2 months, then customer = B type
    ELSE
        IF period-due is less than 3 months, then
            IF amount due is less then Rs. 6000 then the customer = B type.
            ELSE Customer = C Type
        ELSE Customer = C Type

Structured English statements should be properly indented and aligned for readability. It is also advisable to avoid long statements; if necessary break-up a long statement into many short statements.

Programming, the processing logic in a higher level language, is very easy if the logic already explained in structured English. Structured programming languages like PASCAL are based on structured English statements.

18.8 SUMMARY

In this unit we have explained in detail a few structured tools to aid the logical and physical designing, developing, implementataion and maintenance of a Computer Based Information System. Decision tables, trees and data dictionary have been discussed to help logically depict the flow of data, in a CBIS, through Data Flow Diagram. Available hardware, choice of appropriate programming tools are important considerations in the physical design of a CBIS, and are discussed in subsequent units.

18.9 SELF-ASSESSMENT EXERCISES

1 What is a Computer Based Information System (CBIS)?

2 What are the important stages in the life cycle of a CBIS?

3 What is a data flow diagram? Explain its use in system design.

4 What is the role played by data dictionary in a CBIS?

5 Explain the concept of Decision Table, Decision Tree and Structured English in describing the processing logic.

6 Suppose you are designing a system to convert numerical scores (0 to 100) in a test into letter grades (A, B, C, D, F) using the following procedure.

Numerical Score	**Letter Grade**
Less than 40	F
40 to 55	D
55 to 70	C
70 to 85	B
More than 85	A

a) Explain the processing logic through a Decision Table.

b) Explain the processing logic through a Decision Tree.

c) Explain the processing logic through Structured English.

7 What is the difference between a logical design and a physical design?

18.10 FURTHER READINGS

Terrace W. Pratt, 1983. *Programming Languages Design and Implementation,* Prentice-Hall of India: New Delhi.

Edler, John, 1984. *Construction of Data Processing Software,* Prentice-Hall International: Englewood-Cliffs.

Roger, Hunt & John, Shelley, *Computers and Commonsense* (3rd ed.), Prentice-Hall of India: New Delhi.

Sanders, D.H., *Computers Today,* McGraw Hill: New York.

UNIT 19 COMPUTER PROGRAMMING

Objectives

After going through this unit, you should be able to understand:

- the basic programming illustrations, fundamentals of programming languages and writing programme in higher languages.
- developing flow diagram before writing the programme in higher languages.

Structure

19.1 INTRODUCTION

In this unit we shall explore the subject of programming. We will find that the process of programme development is intuitive and requires commonsence rather than a strong mathematical background. Clarity in logical thinking would certainly be an asset, and we believe that it can be cultivated. We shall discover programming concepts in our informal thought process while coping with computational problems. We shall then elaborate on these concepts gradually leading to formalising our thought process.

We know that a computer is normally used to solve complex problems. But if we ask a question as to what computers can really do, the answer would be, nothing more than a school going child would do with paper and pencil, that is, add, subtract, multiply, divide, compare two numbers, and likewise do some simple operations on a string of characters. That means methods to solve complex problems must be expressed in terms of these elementary capabilities of computer. That is precisely what computer programming is all about.

19.2 A PROGRAMMING ILLUSTRATION

There are two ways to cope with complex computational problems. One is to state the method to solve it directly, if we can, which is normally very difficult for a complex problem. The other is to look at the problem as consisting of several somewhat less complex subproblems. This is called refinement of original problem into its subproblems. If we can cope with these subproblems, which we can hope to do because these are some what less complex, we would have coped with the original complex problem. Coping with each of the subproblems can be seen in the same manner as main problem, either state the method directly, if we can, or refine it in terms of further subproblems, and so on. The refinement process can go on until methods to solve all the subproblems on hand can be stated directly without the need of any further refinement.

We shall illustrate the process of stepwise refinement through an example. Consider the following problem:

Find squareroot of a given number

The above problem is far too complex at this stage to state its solution directly. Some of us may be perfectly justified in expecting computer languages to provide means to compute such mathematical expressions directly. For the time being we shall assume that no such facility exists to compute a squareroot, so that we have enough motivation to pursue the problem in all seriousness. Moreover, even if such a facility does exist, it will be in the form of a programme embedded in the computer language. We shall try to develop such a program.

Let us start attacking the problem. In the following paragraph we shall just capture our thought process:

> "First I must take down the 'given' number. Then may be I can guess a number close enough to the actual answer. Obviously it will be almost impossible to hit upon the correct answer on the first try. But if I am able to improve my guess in every successive try, then I should be able to get so close to the actual answer at some stage that I need not try to improve the accuracy any further. Also, since I am anyway improving may guess with every step, closeness of my first guess to actual answer may not matter much. How do I know that I am close to actual answer? If x is the given number and r is its squareroot, then,
>
> $$r \text{ X } r = x$$
>
> or alternatively,
>
> $$r = x/r$$
>
> The above relation gives me a very important clue to test closeness of my guess to actual answer. If I guess g to be the answer, x/g can also claim to be the answer. Unless g happens to be the actual answer, g and x/g will be at the opposite ends of the actual answer, and so their average is likely to be closer to the actual answer than either g or x/g. I can continue until the guess and the improved average guess are close enough.
>
> My thought process has assumed that for the 'given' number the squareroot exists. From the relation r = x/r it is clear that the given number x cannot be negative. If x is negative, then r can neither be positive, nor be negative, not even be equal to zero, meaning thereby that r does not exist."

Above verbose thought process leads us to refinement of original problem in terms of somewhat less complex subproblems in the boxes as follows:

```
Take the given number x

If x is positive
    then
        Find squareroot by successively
        improving initial guess

        Print the answer
    else
        Declare the given number
        to be invalid
```

In the above refinement the subproblem

```
Find squareroot by successively
improving initial guess
```

requires further refinement, all other subproblems being presumably capable of direct specification of solution.

Refining the above further,

```
Make initial guess g

repeat
    Find alternate r = x/g
    Find average g = (r + g)/2
until difference between g and r is negligible
```


19.3 PROGRAMMING VOCABULARY

We are almost through with our stepwise refinement. In stating our refinement we have used some important words such as **if _ then _ else, repeat _ until** and so on. Meanings inherent in these words form fundamental concepts in programming. We shall identify few more of such words to make our programming vocabulary cover all essential concepts in programming, so that we are able to develop programs in English language. Of course, to see these programmes in action in computer we will have to translate the English language programs into one of the computer languages like **BASIC, FORTRAN, COBOL, PASCAL, C,** etc. But to our pleasant surprise we shall discover that most of these languages do retain most of our vocabulory. They would differ only in strict enforcement of syntax rules and offering some advanced level conveniences to advanced programmers. Our choice of English as language to conceive and develop programs will enable us to appeal to our intuitive problem-solving abilities, emphasise programming concepts, and retain option of choosing any computer language or their versions. Programming in a new language would then be simply a matter of learning its syntactic rules and special features.

```
input x                          : Take the given number
 if x > 0  then                  : If valid then try
      g = x/2                    :      any guess would do
 repeat                          :      repeatedly improve
      r = x/g                    :      alternate guess
      g = (r + g)/2              :      average of two guesses
           until abs (r-g) < 0.001 :        is it very close?
           print g               :          print the answer
      else                       : If number was invalid
           print "Invalid input" :          say so
      endif                      :
      end                        : end of program
```

You will be delighted to know that in the above simple looking program we have covered almost all the programming concepts! Words in bold, namely,

```
input
if _ then _ else _ endif
=
repeat _ until
print
end
```


form some of the **key words** in our programming vocabulary. We would like to maintain that the vocabulary corresponds to English language to be used by us to develop programs. Such programs would need conversion to computer language of our choice for implementation, though conversion effort would we negligible for a modern structured computer language. Would you be surprised if we tell you that the squareroot programe developed by us in structured English language would be acceptable to **Microsoft Quick BASIC** (Version 3.0) on IBM PC compatible machines if we just change repeat until to do _ loop until? the Microsoft Quick BASIC 3.0 will be our official implementation language, since programs developed in structured English will require minimal conversion effort to change them to that language. For those of you who do not have access to Microsoft Quick BASIC, or who prefer BASIC of any other pedigree or any other computer language for that matter, we shall give guidelines for conversion.

Apart from key words, we have used several other programming concepts. We have used numeric constants such as 2, 0.001, and string constant 'Invalid input'. We have also used variable names x, g and r to identify various quantities required by us. We have used expressions such as x > 0, x/2, (r + g)/2 and abs (r - g) < 0.001 to compute required quantities. We have even used a standard library function abs (   ) which is normally available in all computer languages. We shall now elaborate on these terms and concepts.

Constants

These have fixed value in the sense that 2 cannot be equal to 4. String constant is simply a sequence of characters such as 'Invalid input' which is a string of 13 characters. The numeric constant can be integer representing whole quantities or floating point number with a decimal point to represent numbers with fractional part. Constants would be probably the most familiar concept to us since we have used them in doing everything that has to do with numbers. Numeric constants can be added, subtracted, multiplied, divided, and also compared to say whether two of them are equal, not equal, less than or greater than each other.

As string constants are sequence of characters, a related strong constant may be obtained from a given one, by chopping off some characters from beginning or end or both, or by appending another string constant at beginning or end. For example, from 'Gone with', we can get 'one with', 'Gone wit', 'one with', 'Gone with wind', and so on. String constants also can be compared in a lexicographic (dictionary) sense to say whether two of them are equal, not equal, less than or greater than each other.

Variables or Identifiers

These are simply names given by us to identify the values that we deal with, e.g. the given number x, the guess g, the revised guess r, and so on. Every computer language would impose its own set of rules on identifiers restricting length of a name, an appendage character to signify type of value it represents such as numeric or string, and so on. Obviously all of them would for bid use of key words as identifiers. Some languages distinguish upper and lower cases (CAPITAL and small letters), in the sense that SURESH and Suresh are considered to be two different identifiers.

19.4 EXPRESSIONS

We know that we can express intended arithmetic operations using expressions such as x/2, (r + g)/2 and so on. Several simple expressions can even be nested together using parentheses to form complex expressions. Every computer language would give order in which various arithmetic operators are evaluated in a given expression. An expression may contain operators such as

parentheses	( )
exponentiation	^
negation	–
multiplication, division	x, /
addition, subtraction	+, -

The operators are evaluated in the order given above. For example, the expression

2 + 8 x (4 - 6/3)/ - 2^4

can be considered to be evaluated as follows:

2 + 8 x (4-6/3)/ - 2^4	sub expression (4-6/3) taken up first
2 + 8 x (4-2)/ - 2^4	division 6/3 within (4-6/3) has higher priority than 4-6
2 + 8x2/-2^4	subtraction (4-2) done next

	(4-6/3) is now complete
2 + 8x2/ - 16	exponentiation 2^4 next followed by negation - 16
2 + 16/ - 16	8 x 2 and 2/ - 16 have same priority but 8 x 2 is at left, so it scores
2 + - 1	16/- 16 has higher priority than 2 + 16
1	that is the final answer.

It is useful to remember order of priority of various operators. But it is safer to simplify expressions and enclose them in parentheses to avoid unpleasant surprises.

So far we have focused on arithmetic expressions. But expression is a very general concept. We mentioned earlier that apart from usual arithmetic we can compare numbers or strings. We do it by using relational operators in relational expressions. The relational operators such as:

=	equal to
<>	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

have same level of priority among themselves but lower priority than arithmetic operators mentioned earlier. The relational expressions result in one of the truth values, either TRUE or FALSE. We should not confuse truth values, TRUE or FALSE with string constants 'TRUE' and 'FALSE' respectively, just as we do not get mixed up between number 235 and string constant '235'. As a number, 235 means something (i.e. a value representing two hundred thirty five), but as a string constant, '235' is a meaningless sequence of three characters. Similarly, as truth values, TRUE and FALSE represent meaning of some assertion implied by result of some relational expression. Some computer languages explicitly provide logical constants. TRUE., FALSE., and/or. YES., No. to represent these truth values. Other languages use integer constants to implicity represent the truth values as for instance FALSE is usually represented by number 0, and any non-zero number is considered to represent TRUE. When a relational expression such as (3>5) is evaluated to be FALSE by such languages, a value 0 is assigned, whereas (5<7) would be evaluated as - 1. Now, since - 1 is a non-zero number, it is a valid representation of truth value TRUE, though some of us become uncomfortably emotional about using a negative number for the purpose. But those of us who are aware that numbers are internally represented in computer as a sequence of bits would appreciate virtues of - 1 as having all its bits at level 1. So at this stage we may grudgingly accept superiority of - 1 as a better representation of TRUE than any other non-zero number.

Note that relational expressions are capable of comparing only two values separated by appropriate relational operator. If we want to express idea such as whether number 7 happens to be within two other numbers 4 and 10, we may be tempted to write relational expression 4< = 7 < = 10. Such reasonable expectation from us may be a bit too complex for a computer language. Then we need to explain our idea in terms of simple relational expressions such as (4< = 7) AND (7 < = 10) would mean that 7 is between 4 and 10. Likewise (2<4) OR (10<2) would mean that 2 is outside 4 and 10. In the preceding examples we have joined two relational expressions using logical operators AND, OR to express our complex conditions to form what is known as logical expressions. Among other logical operators NOT simply negates a truth value in the sense that NOT TRUE is FALSE, and NOTE FALSE is TRUE. The logical operators have priority than relational operators. Among themselves they have the following order of priority during evaluation of logical expressions.

Operator	Meaning
NOT	Simply negates a truth value. Not (2> 5) is TRUE
AND	TRUE if both adjoining expressions are TRUE (4 < = 7) and (7 < = 10) is TRUE
OR	FALSE if both adjoing expressions are FALSE (2 < 4) OR (10<2) is TRUE as (2 < 4) is TRUE
XOR	TRUE only if one of the adjoining expressions is TRUE and other is FALSE. The XOR has same priority as OR. (4 < = 7) XOR (7 < = 10) is FALSE
EQV	This is opposite of XOR. FALSE if truth values of adjoining expressions are different (2 < 4) EQV (10 < 2) is FALSE (2 > 4) EQV (10 < 2) is TRUE
IMP	FALSE only if preceding expression is TRUE and succeeding is FALSE. (2 < 4) IMP (10 < 2) is FALSE (2 < 4) IMP (10 > 2) is TRUE (2 > 4) IMP (10 < 2) is TRUE (2 > 4) IMP (10 > 2) is TRUE

Of the six logical operators NOT, AND, and OR are sufficient to express any logical condition. Any expression involving XOR, EQV and IMP can be converted to equivalent expression involving appropriate combination of NOT, AND and OR. So there is no cause for concern if some of us are bit uneasy about meaning of XOR, EQV and IMP. Of these XOR and EQV may have some natural appeal. But many who are new to logical expressions would find IMP difficult to swallow. To get some intuitive feel for IMP operator, remember that we shall refuse to accept the assertion. 'Truth of A implies truth of B' if A is TRUE and B is FALSE, but gladly accept that both A and B are TRUE. However, can we refuse the assertion if A itself is FALSE? By the way, A IMP B is equivalent to NOT A OR B, which means that it is FALSE only if A is TRUE and B is FALSE, and in all other cases it is TRUE.

Although our discussion on expressions has become somewhat verbose, writing these expressions in our programs would presumably be quite natural as we are going to take English language route to develop programs. Crucial would be to grasp conditional expressions (i.e. relational and logical ones), also known as boolean expressions. They are to be naturally expected around terms such as if, till, until, and while in our thought process as they naturally imply some condition to be fulfilled.

Expressions would be our basic workhorses. Our programs would be full of them, and while executing the program, computer would be preoccupied most of the time evaluating them. But it would also be doing other things such as taking some values as input from an input device, assigning a value or result of an expression to a variable, send a value for display to a display unit, pondering to be or not to be on some conditional expression, repeating some tasks until we are satisfied, and calling some slave-programs to perform specific tasks. In the following discussion we shall elaborate on these actions.

Input statement

Syntax: input = list of variables

We used it to input 'given' number x in our squareroot program. That is what it is normally used for, to supply values to a list of variables through input device such as keyboard. On execution a question mark? would be displayed on the screen, in response to which we are expected to supply sequence of desired values matching appropriately the type (i.e. integer, floating-point or string type) and number of variables in the list.

Assignment statement

Syntax : identifier = some expression or identifier

This is the most obvious statement used for assigning value to an identifier (i.e. variable). In BASIC languages this is also known as LET statement though the key word LET is optional in most of them. Note the use of = as assignment operator. You would recall that it was also used as relational operator. Some purists justifiably object to ambiguous use of =. In Pascal, : is used for assignment instead of just = which is used as relational operator. In language C, = is used for assignment but == for relational operator. We, like in Microsoft Quickbasic, shall put up with the ambiguous use. Context of its use will hopefully resolve the ambiguity. For example, x=2=2 is expected to assign a value -1 to x because the expression to the right of x = is evaluated as a relational expression 2=2 being TRUE!

Print statement

Syntax: print = list of variables

This is again an obvious way to display values of specified list of variables on display screen. We shall make use of print to represent generic output statement independent of particualr output device such as display screen, printer and so on. Computer languages normally provide very rich facilities for making the display as attractive as possible. Display has a touch of glamour in it that consumes more than fair share of new programmer's time and energy. Modern languages provide exotic features like full-screen display, windowing and so on that is clearly beyond the scope of this unit. You are, therefore, requested to refer to respective language manuals for details.

If _ then _ else _ endif structure: (To Be or Not To Be!)

Syntax: if (condition)

```
        then
          | statements for Task A |
        else
          | statements for Task B |
        endif
```

This is one of the most important program-structures that enables a modern programmer to develop well structured programs. Note that we are calling it as structure rather than a statement. As structure it accommodates statements corresponding to alternate tasks to be performed depending upon truth value of the condition. The structure has three components, a conditional expression the truth value of which determines the action, a then _ clause denoting task to be performed if the condition is TRUE, and an optional else _ clause with task to be performed if the condition is FALSE. It is highly recommended that the structured programs be appropriately indented to properly delineate boundaries of various clauses and structures. Our stepwise refinement will gradually and naturally unfold the program structures such as if _ then _ else _ endif and others. Each subtask taken up for further refinement has its own indentation. Recalling our squareroot program, note the clear indenting that makes the program highly readable and easy to comprehend.

```
input x
if x > 0
   then
   |    g = x/2
   |    repeat
   |    |    r = x/g
   |    |    g = (r + g)/2
   |    until abs (r - g) < 0.005
   |    print  g
   else
        print "Invalid input"
endif
end
```

Note three levels of indentations. First denotes boundary of if _ else _ endif structure. Second is then _ clause and else _ clause. Third is repeat _ until loop structure that we shall be discussing shortly. Proper indenting enables us to focus our attention on particular task that needs to be attended to.

Returning to our if _ then _ else _ endif structure after a useful dose on virtues of indenting, we note that the structure does not impose any restrictuion on contents of then or else clauses. They may have any number of and any type of program structures snugly nested within them. See for example, repeat _ until structure nested within then _ clause of if _ then _ else _ endif structure. Note that we have presented most general and useful if _ then _ else _ endif structure with multiple line then and else clauses. Other variations are single line clauses and without explicit else clause. Some languages, particularly BASIC of older vintage are hopelessly unstructured. The translated version of our structured squareroot program in one such BASIC would look like,

```
10 INPUT X
20 IF NOT (X > 0) THEN 100
30 LET G = X/2
40 LET R = X/G
50 LET G = (R + G)/2
60 IF NOT (ABS (G - R)< = 0.001) THEN 40
```

```
70 PRINT G
80 GOTO 32767
100 PRINT "INVALID INPUT"
32767 END
```

Note IF NOT (..) THEN .. used to simulate if _ then _ else _ endif as well as repeat until loop structures. This ambiguous use of a statement makes the program difficult to comprehend. Moreover, emphasis seems to be on navigation through program using GOTOs making then highly unreadable. More about it later. Let us close our discussion on if_then_else_endif structure by relating it to our Microsoft Quick BASIC. The structure is fully provided by the language. For some strange reason, syntaxwise it insists that the then key word must follow if (condition) in the same line, and also insists on splitting endif as end if. It extends the structure to allow use of elseif _ clause to accommodate multiple conditions. Following example would clarify the use of elseif_ clause to handle a quantity discount situation:

```
if (x > = 10000) then
        price = 10
    elseif (x > = 8000) then
        price = 10.5
    elseif (x > = 6000) then
        price = 11.0
    elseif (x > =4000) then
        price = 11.5
    elseif
        price = 12.0
end if
```

Note that elseif _ clause is part of if _ then _ else _ endif structure, making it a if_then_elseif_.. _else _endif structure. Also, note that only one of the clauses would be executed. A very close cousin of this extended structure is case _ endcase structure which we may discuss later. Other variations of this structure are single line versions using line numbers or statement labels to GOTO the desired part of the program, like the ones we have used in the example BASIC program. The use of GOTOs is inevitable in unstructured languages. But they can be and should be always avoided while using structured languages like Microsoft Quick BASIC, Pascal, C and so on, except in some unavoidable situations that we shall present later.

19.5 LOOP STRUCTURES

Obviously, purpose of a loop structure is to repeat certain task until no more repetition is desired. We shall be presenting several variations of loop structure appropriate to handle different situations.

repeat _ until structure:

Syntax : repeat

Statements for task being repeated

until (condition)

Our choice of repeat _ until was quite natural in sequareroot problem where we started with some initial guess and iteratively improved it until we were satisfied with level of accuracy. A loop structure is characterised by the body of statements to be repeated, and condition for termination of loop. In repeat _ until version the condition comes after the body. The condition is tested after the body is executed, if it is evaluated as FALSE then body is executed again. The execution of the body is repeated until condition is evaluated to be TRUE. That reminds me of a joke about a doctor who prescribed for the patient to

```
repeat
        The Dose
until (the patient dies)! That's a cruel one?
```

Note that in repeat _ until version the body would be executed at least once since decision to terminate the loop is taken only after executing the body. Most structured languages do provide repeat _ until loop structure, perhaps using different key words. The Microsoft Quick BASIC calls it do _ loop until. In unstructured languages this may be simulated by placing a statement

IF NOT (condition) THEN label

after the body where label refers to the first statement of the body as we have shown in our translation of sequareroot program to unstructured BASIC.

while _ endwhile structure:

```
Syntax: while (condition) do
    | Body of statements |
endwhile
```


In this version, the condition is tested before every execution of the body. The decision made is whether to execute the body. The body is executed if the condition is evaluated to be TRUE, after which the condition is tested again. The loop is active as long as the condition evaluates to TRUE, and terminates when the condition is evaluated to be FALSE. Unlike the repeat _ until version, the while _ endwhile does not guarantee the execution of the body at least once. In case the condition is found to be FALSE for the first time, the body will be altogether skipped. To illustrate while _ endwhile structure we could have developed our squareroot program as:

```
input x
if x > 0
   then
      g = x/2
            while (abs (g - x/g) > 0.001) do
                  g = (g + x/g)/2
      endwhile
      print g
   else
      print "Invalid input"
endif
end
```

Note that in this version if our initial guess itself is close enough then we do not need to execute the body. The while _ endwhile loop structure does a sort of

"Look before you loop" syndrome!

Just as repeat _ until, most structured languages do offer the while _ endwhile structure, albeit with different key words. The Microsoft Quick BASIC gives two syntax options, the do while _ loop and while _ wend with identical meaning. For the benefit of unstructured languages the while _ endwhile loop structure may be simulated by placing

IF NOT (condition) THEN label

statement before the body, where label corresponds to a statement just after the last statement of the body.

repeat _ endrepeat structure:

Snytax: repeat

Body of statements including provision for exiting the loop

endrepeat

Two previous versions of the loop structure were characterised by the condition being either after or before the body. The repeat _ endrepeat caters to the third possibility where termination condition can be known only inside the body. The structure does not provide for any termination condition, in the sense that it is truly an infinite loop structure. It is the responsibility of the body to force an exit out of loop. For example, suppose that we want our squareroot program to compute as many squareroots as user cares to compute before he chooses to terminate the program by inputing a zero or a negative number. We can put our previous program in a repeat _ endrepeat loop structure with a provision to exit if x is less than or equal to zero.


```
repeat
  input x
  if x < = 0
    then
      exit
    else
      g = x/2
      while abs(g - x/g) > 0.001 do
            g = (g + x/g)/2
      endwhile
        print x, g
  endif
endrepeat
```

Note the use of exit statement inside the body of repeat _ endrepeat loop structure. In general exit statement is meant to terminate innermost loop enclosing it.

Usually, repeat _ endrepeat is not explicitly provided in structured languages as it can be very easily simulated with a forever TRUE condition in repeat _ until or while _ endwhile structures. The Microsoft Quick BASIC does provide the repeat _ endrepeat structure as do _ loop without any until or while key words in its syntax. It enables termination of such loop with exit do statement. For unstructured BASICs the repeat _ endrepeat may be simulated using

GOTO label

statement after the last statement of the body, where label corresponds to the first statement of the body. Obviously somewhere in the body there must exist a statement such as

if (exit condition) THEN label 2

where label 2 corresponds to next statement after the GOTO label.

for _ endfor structure :

Syntax: for var = start to finish step incr do

Body of statements

endfor

This loop structure is distinctly different from the ones we have discussed so far. Unlike them, the for _ endfor structure counts the number of times the body is to be executed on the basis of number of times a control variable is incremented from start to finish in specified steps. Managing the control variable is the responsibility of the for _ endfor structure, so we need not and should not worry about assigning any value to it in the body of statements. Any attempt to do so may land us in trouble. However, the value of the control variable is available for our use as we shall see in the case of array data structure to be discussed later. To illustrate the for _ endfor structure let us overstretch our squareroot example further. Suppose for some strange reason we would like to go through fixed number of iterations of improving the guess. The reasons may not be as strange as you may like to believe. Imagine finding squareroot of very small numbers, even smaller than the limit of accuracy, 0.001 specified in our program. So we have the following version of our squareroot program.


```
input x
if x > 0
  then
    g = x/2
    for i = 1 to 10 do
        g = (g + x/g)/2
    endfor
    print x, g
  else
    print "Invalid input"
endif
```

Did you notice that step is missing in the for _ endfor structure? Yes, for obvious reasons. Most often the for _ endfor structure is used for normal counting from start to finish with increment of 1. So as a matter of convenience we do not require to specify any step if it is a normal step of 1.

This loop structure is so famous that it is available in all languages, structured or not. As usual, key words provided may be different. The Microsoft Quick BASIC calls it for _ next, and optionally provides for specification of control variable after next. Many other BASICs insist on mention of the control variable after next.

We have discussed program structures quite elaborately. Let us turn our attention to data structures that hold our values. So far we have been dealing with simple variables that hold only one value of desired type. So we have integer variable, floating-point variable, and string variable holding only one value of corresponding type at any moment. Can we have data structures that are capable of holding more than one value? Programming languages do offer complex data structures required for advanced programming. We shall discuss one of them.

Array data structure

Syntax: <u>var</u> (subscripts)

We can declare that a variable is required to hold more values of which each can be referred to by using corresponding subscript. Such a data structure is called an array and variable representing array is known as subscripted variable. They are normally declared in a dimension statement. In the simplest case array has only one dimension representing a sequence of values. Any of the elements in the sequence may be identified by an index. For example, consider an array of 10 values. The array will be declared by using a dimension statement.

i	A (i)
0	5.15
1	3.75
2	6.95
3	4.55
4	2.75
5	1.85
6	9.95
7	4.35
8	1.50
9	7.45

```
dimension    A (9)
```

where A is the variable name and 9 refers to maximum value of index we intend to use. This array could be easily filled using statements like

```
for i = 0 to 9 do
    input A (i)
endfor
```


In the example element A (3) has a value 4.55, A (7) is 4.35 and so on. The subscripted elements may be manipulated just like simple variables. They especially go hand-in-hand with control variable of for _ endfor structure as can be seen in above example, where one statement, input A (i) forming body of a for _ endfor loop is capable of inputting values for all the elements.

Arrays can be more complex than a sequence. We can visualise a two-dimensional arrangement where elements are arranged in rows and columns, where individual element is identified by two subscripts, one referring to row and the other to the column. The situation is presented in the following example.

j	0	1	2	3
i				
0	17	23	9	13
1	33	51	18	19
2	41	35	49	11
3	73	29	14	47
4	95	63	77	29

Relevant program segment would be

```
dimension A (4,3)
for i = 0 to 4 do
  for j = 0 to 3 do
     input A (i,j)
  endfor
endfor
```

In the example, A (3,2) is 14. We could add a third dimension with one more subscript and so on. The Microsoft Quick BASIC would allow you to go up to sixty three dimensions! Normally, you would not have occasions to go for more than two dimensions. On each dimension, the corresponding subscript may range from 0 to 32767. The maximum value each subscript will take is required to be declared in what all BASICs call a DIM statement. Although most of them do condone certain amount of laziness in explicit declaration (e.g arrays with subscript value less than 10), you are strongly advised to resist such temptation of non-declaration. In fact, most modern languages do insist on explicit declaration of even simple variables to avoid unpleasant surprises later while running the programme. We close our discussion on arrays with a reminder that arrays can be of any type, integer, floating-point, or string in the sense that each element can hold the value of that type. The Microsoft Quick BASIC identifies integer variables with special declaration character % as the last character in the identifier name [e.g. stock %, age % (60) denotes a simple integer variable and an integer array of 61 elements respectively]. It identifies string variables with $ as last character [e.g. z$, name$ (60)].

After elaborating on programming concepts like constants, variables, expressions, input, print and assignment statements, if _ then _ else _ endif and loop structures, and array data structure, let us develop some programs using them.

Let us develop a program to sort a sequence of numbers in ascending order. Too simple? Do not underestimate this problem. It has deserved a fat 800 pages book by eminent computer sceintist (See **Searchin and Sorting** by Donald Knuth). Also, more than 60 per cent of world computing power at any time is engaged in this activity! So we have,

Given a sequence of numbers in any random order, print them in ascending order.

Remember arranging cards during a card game? Pick them up one by one simultaneously arranging the picked cards in required order. Can we express it in terms of subproblems?

declare array to hold sequence of number
How many numbers?
Take one by one arranging the ones taken so far in ascending order
Print the ordered sequence

Refining it further,

```
dimension A (100)
input n
for i = 1 to n do
      input A(i)
```

Put the just _ picked in its proper position so that ones taken so far are in ascending order

```
endfor
for i = 1 to n do
  print A(i)
endfor
end
```

Let us refine the one in the box further. Ascending order means smallest number at the top and the highest at the bottom. For every number, the number above it cannot be bigger than that number. If that is the case, our number can pull down the bigger number to move-up and continue its ascent till number above it is not bigger than it, or it has itself reached the top. Moral of the story is, everybody rises to one's level of incompetence! Expressing it step by step and polishing it up,

Informally...	Polished one
where am I? at the bottom	k = i
while I am not on top and a bigger one is above	while k > 1 AND A(k) < A(k−1) do
pull it down, I take that position	x = A(k) A(k) = A(k−1) A(k−1) = x
and so on	k = k−1
	endwhile

The boxed statements above achieve famous swapping of two elements A(k) and A(k-1). Try doing it without using an intermediate variable x. You cannot do it without losing one of the values in the process! The swapping is so famous that the Microsoft QuickBASIC provides a swap statement to express it!

Putting it all together,

```
dimension A(100)
input n
if n < = 100
  then
     for i = 1 to n do
       input A(i)
       k = i
       while k >1 AND A(k) < A(k−1) do
         x = A(k)
         A(k) = A(k−1)
         A(k−1) = x
         k = k−1
       endwhile
   endfor
   for i = 1 to n do
     print A(i)
   endfor
 else
   print "Too many numbers"
endif
end
```

Note little things like making sure that we do not cross self-imposed limits on number of elements in dimension statement. During the process of stepwise refinement we do not get digressed by such matters of details. After we are through with the development of main body of logic we can always insert it in some conditional structure. This also underscores importance of viewing programs as collection of bodies of statements snugly fitting in appropriate program structures, rather than sequence of statements to be navigated through GOTOs. This collection of bodies may be developed some what independent of each other, even by different individuals, and even as independent subprograms to be called from different parts of the program. That is the subject matter we are going to discuss next.

19.6 SUBPROGRAMS

Considering the two programs that we have already developed, would you agree that they are quite general purpose in nature? No wonder that squareroot program is generally available as standard library function in almost all languages. Many operating systems do provide 'sort' as an utility. There are many such problems that are required to be solved as subproblems by different programs, or even by a given program at several places within the program. For example, it is quite possible that we need to handle two or more arrays and we need all of them sorted in ascending order. We may obviously reproduce the relevant body of statements suitably modified to suit the local conditions wherever it is required. But can this reinventing of wheel be avoided? Yes, the answer is subprograms which are essentially slave-programs that can be invoked or called upon to do specific tasks by any part of main program or other subprograms. There are two kinds of subprograms, one whose main purpose in life is to have one value apart from any other side effects one may entrust to such subprogram, and the other kind are those whose preoccupation is the side effects only. Implication is that former may be used as operands in expressions, because they are equivalent to a simple variable or element of an array as far as having a value of some type is concerned. The subprograms with responsibility of having a value so that they can be used as operands in expressions are called functions. Other subprograms meant only for side effects are called procedures or subroutines. Note that something more is expected from functions than procedures, though our focus on one value of function in contrast with procedures, may give us a different feeling. There is some amount of redundancy in concept of procedures, there is no reason why we cannot ignore value of a function and enjoy only the side effects. The language C, in fact does just that. In language C, everything including the main program itself is a function! We shall however respect the time honoured convention of treating functions and procedures as distinct.

Function structure:

Syntax: function <u>name</u> (<u>argument list</u>)

Body of statements including those assigning value to function

endfunction

The function _ endfunction structure has three components. The name identifies the function and is normally governed by syntax rules of variable names or identifiers. The function is invoked by using the name just as variables in expressions. The argument list consist of parameters that are passed on to the function to participate in the computation. Body of statement is just like such a thing in any other structure with some subtle differences. Somehow we should be able to assign that one value which the function is expected to return to the expression invoking it. We shall do it as in Microsoft Quick BASIC using assignment statement. Some languages do it by enclosing the value or expression in a return statement. Other issue is of sharing of data beyond what is communicated through the argument list, between main program and subprograms, both the functions and the procedures. Some languages allow explicit sharing by declaring some variable to be common or external, all other variables being considered as local to the subprograms, distinctly different from identifiers with same name in the main or other subprograms. Other languages like Microsoft QuickBASIC, consider all identifiers as global unless explicitly declared as local by using static kind of a statement. We shall close the discussion on function after giving BASIC syntax for it. In BASICs like Microsoft Quick BASIC, where multi-line function definitions are allowed, the syntax would be,

```
DEF FNname (argument list)
STATIC list of local variables
    Body of statements including
    FNname = (expression)
    and, possibly
    EXIT DEF
END DEF
```

Single line function definition in such languages would be a straight forward

```
DEF FNname (argument list) = (expression)
```

Procedure Structure:

```
Syntax: procedure name (argument list)
            Body of statements
        endprocedure
```

There is nothing more that we can discuss about procedures as we have discussed the related issues like sharing of data among programs, local and globally shared variables in the context of functions. We shall only discuss their form in languages like BASICs. Unstructured BASICs do not normally provide independent procedures in their full form. Parts of the program may be visited using GOSUBs to be RETURNed on encountering such a statement. There is no argument list, list, local variables and so on. the Microsoft QuickBASIC offers full-fledged procedure capability with the following syntax.

```
SUB name (argument list) STATIC
    STATIC list of local variables
    SHARED list of main program
               varibales to be shared
    Body of statements possibly
    including EXIT SUB
END SUB
```

Note the STATIC in SUB line as well as for declaration of local variables. Former says the procedure cannot be recursively called by itself. The latter meaning as local variables is the same as in function declaration. In addition, we have SHARED list of variables that were declared in main program but also to be shared in the procedure. The Microsoft Quick BASIC has some what different scheme for sharing of data between main program and the procedure compared to that with function. We can declare the COMMON list of variables in main program and procedures, matching in type possibly with different names. We can declare some as COMMON SHARED list of variables in the main program so that they can be referred by procedures by name without having to declare as COMMON or SHARED in the procedure. Third option is to declare them as SHARED list of variables as mentioned before. Note that variables used in procedures are independent of those used in the main program or other procedures unless they figure in COMMON SHARED list in main program or SHARED list in procedure. In case we want to use variables declared as COMMON SHARED in main program as local variables in procedures we have to declare them as STATIC in the procedures. While developing programs in structured English, we shall follow the same conventions as the Microsoft Quick BASIC for handling global and local variables in subprograms.

We shall illustrate use of subprogram by developing a program for advancing calendar by a day. The solution procedure will have to take care of limits on number of days in a month, leap years and so on. This will also give us an opportunity to explore multiple choice case _ endcase structure. We shall follow similar narrative style as in sorting program presenting informal and polished program side by side capturing process of program development.

Informally...	Polished program
Take given date	input d,m,y
if valid	if valid (d,m,y)
then	then
advance it by a day	call advday (d.m.y)
print the new date	print d.m.y
else	esle
print "Invalid date"	print "Invalid date"
	endif
	end

We have to develop two more subprograms, a function valid (d,m,y) and a procedure advday (d,m,y) for advancing the day. Please resist the temptation of writing a nonolithic program where obvious subprogram possibilities are clearly visible.

Taking up advday further...	procedure advday (dd, mm, yy)
simple, just add 1 to day!	dd = dd + 1
oops! may be a mistake.	if NOT (valid (dd, mm, yy))
what if it was 31 August?	then
the next day would be 1 not 32	dd = 1
but you should advance month too	call advmonth (dd, mm, yy)
	endif
	endprocedure

```
Taking up advmonth further...        procedure advmonth (d,m,y)
as before, advance month by 1                m = m + 1
think of worst, 31 Dec.                      if NOT (valid (d,m,y))
next month would be 1 rather                     then
than 12, the year would advance              m = 1
too                                          y = y + 1
                                         endif
                                     endprocedure
```

We have refined everything except the function for ascertaining validity of a date. Note that valid (d,m,y) has not only been used in the main program but also has been invoked in every procedure to check whether advancing day or month has made the date invalid. To check validity we consider various cases of day, month or year values being valid or not. We naturally use a case _ endcase structure which is much simpler than its big brother if _ then _ elseif _ .. _ else _ endif when all the conditional expressions consist of only one variable participating in equality testing.

```
Taking up valid ..                   function valid (xd, xm, xy)
but is it obviously                    if (xd < 1 OR xy < 0)
invalid?                                     then
                                                 valid = 0
how about day & month                        else
check case by case                               case xm of
any of the 31-day months?                            1,3,5,7,8,10,12
Note the use of relational                               : valid = (xd < = 31)
expression in assignment.
any of the 30-day months?                            4,6,9,11
                                                         : valid = (xd < = 30)
is it February?                              2 : valid = (xd < = 28 + leap (xy)
Remember to write leap function
invalid month                                    : valid = 0
                                             endcase
                                       endif
                                     endfunction
```

How do we find out whether year is leap? If the year is divisible by 4 but not by 100 except by 400 then the year is considered to be leap. Thus 1988 is leap, but 1900 is not, however 2000 is leap. The Microsoft Quick BASIC gives a operator called MOD to compute remainder of an integer division. MOD has a priority higher than addition or subtraction.

```
Lastly....
To compute leap factor               function leap (yy)
divisible by 4?                        if (yy MOD 4)
                                       then
No                                           leap = 0
divisible by 100?                      elseif yy MOD 100 then
4 but not 100                                leap = 1
divisible by 400?                      elseif yy MOD 400 then
```

```
100 but not 400                           leap = 0
                                    else
by 400                                    leap = 1
                                    endif
                                  endfunction
```

We end the discussion on calendar advancing program by making some observations on Microsoft Quick BASIC implementation. Note that all the function definitions must precede their use. Also note that case _ endcase structure as used by us would be translated as

```
SELECT CASE xm
   CASE 1, 3, 5, 7, 8, 10, 12
      fnvalid = (xd < = 31)
   CASE 4,6,9,11
      fnvalid = (xd < = 30)
   CASE 2
      fnvalid = (xd < = 28 + fnleap (xy))
   CASE ELSE
      fnvalid = 0
   END SELECT
```

Note that the function names have been presumed to have been changed to begin with fn as required by the language.

19.7 MISCELLANEOUS TOPICS

String Handling:

So far we have been considering mainly the numeric problems. But the concepts covered so far are equally applicable to textual problems involving string type of values. The body of statements in various program structures would consist of string expressions performing concatenation (simply joining using + operator), and comparison of strings. Languages provide rich set of functions for string handling. Prominent among the functions available in the Microsoft Quick BASIC are

LEN (x$)	Returning number of characters in value of x$
LEFT $ (x$,n)	REturning a string consisting of leftmost n characters in value of x$
RIGHT $ (x$, n)	Close cousin of above, returning a string of rightmost n characters in value of x$
MID $(x$,n,m)	Middle cousin, returning a string of m characters from value of x$ beginning with nth character
INSTR(n,x$, y$)	Detective function to search whether value of y$ is occurring as substring in x$ beginning with nth character. If found, returns the position indicating start of y$, returns O otherwise.

To understand the above functions, the string value of x$ of length 1 = LEN(x$), may be visualised to be consisting of three parts, namely, LEFTS (x$,n) of first n characters, MID$(x$, n + 1,m) being the next m charactaters, and RIGHT$ (x$, 1-n-m) of remaining tail-enders of length LEN (x$)-n-m. Any of these parts may be the substring of our interest. and INSTR (n,x$,y$) locates that for us.

Let us see string handling in action through couple of programs. First we shall develop a program to check whether a given word is a palindrome that reads the same backward as forward (e.g. madam, rotor, malayalam).

```
Informally..                    polishing it up...

The given word                  input w$
Start looking it backwards      r$ = " "
taking each character           for j = LEN (w$) to 1 step - 1 do
build reverse word                 r$ = r$ + MID$ (w$. j. 1)
completely                      endfor
Comparing the two               if w$ = r$
Print appropriate                       then
message                                       print w$ "is a palindrome"
                                        else
                                              print w$, "is not a palindrome"
                                endif
```

Alternatively, the for _ endfor part could also be equivalently written as

```
looking from front              for j = 1 to LEN (s$) do
and appending every new            r$ = MID$ (w$, j, 1) + r$
character in front              endfor
```

Next let us write a general letter to x x x so that we can later personalise it to any particular name where all the occurrences of x x x will be replaced by the particular name. So,

```
Take down the letter            L$ = " "
Initially it is blank           repeat
Take it line by line               line input p$
unless current line is             if LEFT$ (p$,3) = "ZZZ" then exit
a nonsense "ZZZZ.."                L$ = L$ + p$
keep appending it               endrepeat
who is the person?              input X$
Replace by x x x him/her        repeat
x x x still there?                 n = INSTR (O, L$, "x x x")
unless not there                if n = 0 then exit
                                   t = LEN (L$)
replace                         L$ = LEFT$ (L$, n-1) + x$ + RIGHT$
                                (L$, t-n-2)
                                endrepeat
```

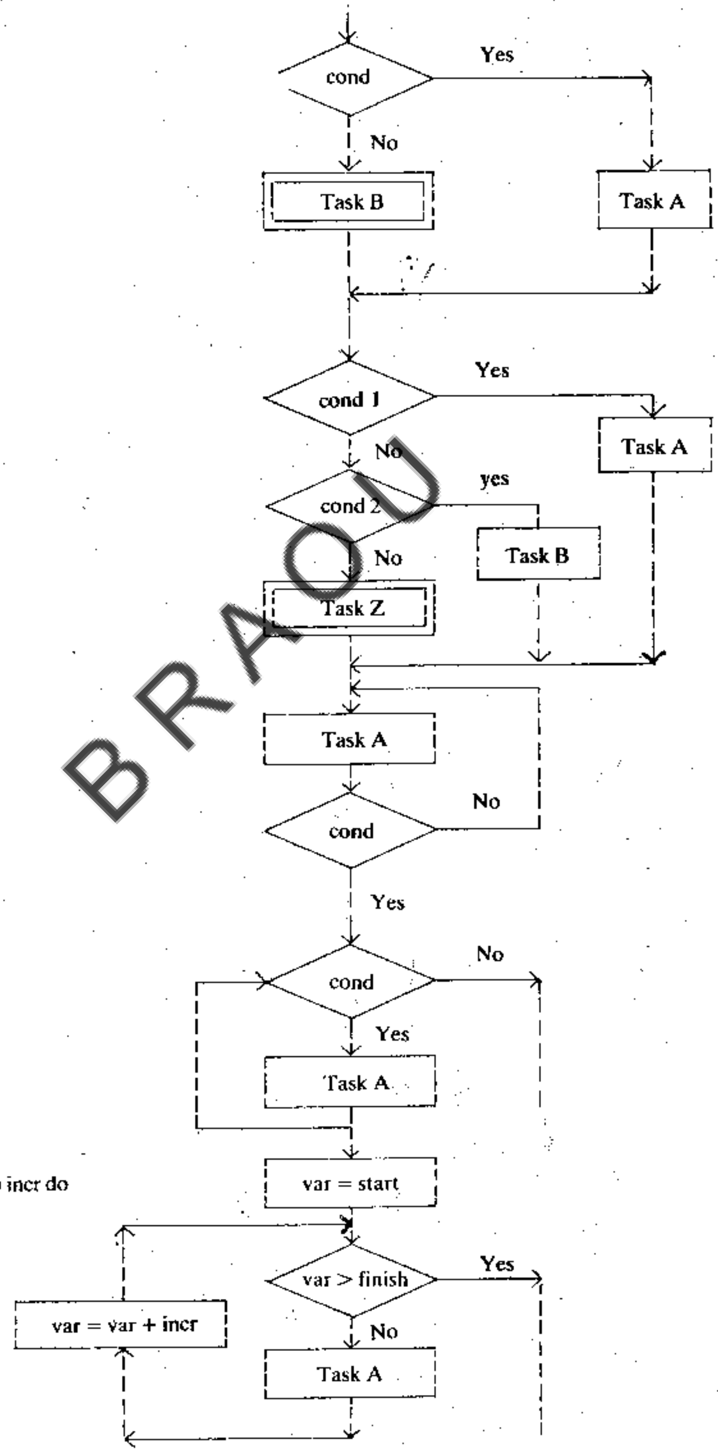
Note the use of line input for inputting a line. This is different from plain input in the sense that entire line with all its characters (spaces, commas, quotes, even return/line feed) are considered as part of string. Languages do provide such statements. The Microsoft Quick BASIC calls it line input whereas other BASICs call it inputline or even LINPUT. Also, note a smart trick of "ZZZZ..." to indicate that we are through with our letter.

Note also tne cut and paste kind of operation in L$ = LEFT$ (L$, n - 1) + x$ + RIGHT$ (L$, t-n-2), where unwanted "x x x" has been surgically removed by isolating LEFT$ (L$, n-1) and RIGHT (L$, t-n-2) around it and pasting x$ in its place.

Program Structure

```
if (cond.)
  then
     Task A
  else
     Task B
endif

if cond 1
  then
     Task A
  elseif cond 2 then
        Task B
  else
        Task Z
endif

repeat
        Task A
until (cond)

while (cond) do
     Task A
endwhile

for var = start to finish step incr do
     Task A
endfor
```

Flowchart



Why not Flowcharts?

With this we conclude our discussion on programming concepts. Some of you might be wondering why we have not discussed flowcharts. Our dislike for flowcharts is mainly due to the fact that they emphasise navigational view of program development just as GOTOs in programs. Moreover, flowcharts of even moderately complex programs tend to become very cluttered. They merely turn out to be documentation aids rather than program development tools. On the other hand, the structured approach followed by us not only helps in development through stepwise refinement but also makes the program self-documenting through proper indenting. Such programs may be viewed as textual flowcharts. However, if you still insist on knowing about them, we have given above the flowcharts for various program structures so that you can build the flowchart for your program if you like.

Note that flowchart for the for _ endfor loop structure is for positive incr value. In case the incr is negative value, then condition being tested would be var < finish.

Apart from being cluttered and inherently navigational, the flowcharts also make ambiguous use of decision symbol for single and multiple conditional branching as well as different loop structures. Therefore we strongly suggest that you do not use them for program development or documentation purposes. We also suggest that you develop program in structured English and then translate them into the implementation language of your choice. If the implementation language is very close to our structured English, say, differing only in choice of key words as in case of the Microsoft QuickBASIC then you may conveniently use its key words for program structures directly, but still following the stepwise refinement process of program development. If your implementation language is relatively unstructured, then we suggest that you develop a standard translation table for translation of each program structure that can be used to translate programs developed in structured English into your implementation language.

19.8 SUMMARY

In this unit we have explained the basics of computer programming. It follows the gradual exposure to writing programmes in higher language and logically matched it with the structured English. In this unit we have also exposed you to a number of programmes used commonly and simultaneously explained their processing logic. Further towards the end we have referred briefly to development of flowcharts.

UNIT 20 PROGRAMMING LANGUAGES: COBOL AND ITS APPLICATIONS

Objectives

After going through this unit you should able to:

- understand the basic features of COBOL
- understand and develop familiarity with the different steps involved in writing programmes in COBOL
- start writing simple (elementary) programmes in COBOL

Structure

20.1 INTRODUCTION

COBOL is a computer language designed for use in business data processing. COBOL stands for Common Business Oriented Language. Historically, the language was first conceived at a Pentagon meeting in 1959. At that meeting representatives from government, business users and computer manufacturers decided that it was feasible to proceed with development of a higher level language that would answer the needs of business data processing. However, it was not until August 1968 that a standard version of the language was approved by what is now called the American National Standards Institute (ANSI). This standard version, ANSI COBOL, was again revised in 1974 and has now been implemented by all major manufacturers. In this note we present the features of the commonly used elements of ANSI COBOL and some applications of this language.

20.2 BUSINESS DATA PROCESSING

Business data processing involves recording, manipulation and analysis of data and preparation of useful outputs and reports. In general business data processing problems are characterised by relatively simple algorithms coupled with high-volume input-output. Financial accounting, Sales invoicing, Inventory control are some examples of data processing.

In order to accomplish these tasks through a computer system we need facilities in the programming language to describe the structure of data files, have language statements to create, manipulate and analyse the data in the files, and to present the processed information in the form of reports.

In addition to these, documentation of programs is an important requirement of business applications which are dynamic. Therefore a business data processing language should have facilities to write the programs in a manner that can be understood and modified if necessary. In the subsequent sections we see how COBOL supports these features.

20.3 BRIEF OVERVIEW OF COBOL

Since input-output is a prime concern in business application programs, the COBOL design emphasises features for the specification of properties and structure of input-output files. COBOL supports hierarchical data structures building from elementary data items to group data items and records. This implies that the programmer can define convenient data structures and perform operations on those structures using the COBOL statements. These features enable the programmer to create data on computer, manipulate and analyse the data and also prepare reports in the desired form with little effort.

Another important language characteristic is the English-like syntax of the language statements. The computer programs written in this language are substantially self-documenting. Self-documentation is a feature of a language that allows the reader of a program to understand its function and follow its processing steps. COBOL instructions are English-like, and the programmer with little effort can develop self-documenting programs.

20.4 STRUCTURE OF COBOL PROGRAMS

COBOL programs have to be organised into four divisions. This organisation is largely a result of two design goals:

i) Separate machine-dependent from machine-independent program elements.

ii) Separate data descriptions from algorithm descriptions so that each might be modifiable without affecting the other.

The result is the following program structure:

1. **Identification Division** — Giving program identification details.
2. **Environment Division** — Giving the details of computer environment in which the program will be executed.
3. **Data Division** — In which the structural details of data files and variables used in the program are described.
4. **Procedure Division** — In which the processing details are described through COBOL statements.

COBOL programs should be written according to a special structure which is organised into a hierarchy of parts as described below:

A character is the lowest form in the program structure.

A word is made of one or more characters.

A clause consists of characters and words and is used to specify an attribute of an entry.

A statement is syntactically valid combination of words and characters written in the PROCEDURE DIVISION and beginning with a verb.

A sentence is a sequence of one or more statements, the last of which is terminated by a period followed by a space.

A paragraph consists of one or more sentences.

A section consists of one or more pragraphs.

A division consists of one or more praragraphs or sections.

Every program consists of four divisions in the following order: IDENTIFICATION DIVISION, ENVIRONMENT DIVISION, DATA DIVISION, and PROCEDURE DIVISION.

We illustrate these features with few examples from data processing environment.

20.5 COBOL IMPLEMENTATION OF A SIMPLE DATA PROCESSING PROBLEM

Consider a data processing problem in which we are required to prepare a statement giving the value of items in stock by processing records having data on stock quantities and unit prices in addition to the code and description of items in stock.

The layout of data record in the item-stock file is given below:

Item code			Description	Stock Quantity	Price
MG	SG	SN			
2	2	2	24	4	6: 4+2
Numeric Integer			Alphanumeric	Numeric Integer	Numeric non-integer

The Item code is a group data field consisting of main group (MG), sub group (SG) and serial number (SN): each being a two digit numeric integer. Description is 24 characters long alphanumeric data field. Stock quantity is four digit numeric integer data field. Finally the price per unit is a six digit long numeric non-integer field with four integer digits and two decimal digits.

The organisation of COBOL program for this problem would be as follows:

Identification Details

The identification details such as program identification, author, installation, date of writing, and date of compilation will have to be presented in the IDENTIFICATION SECTION of the program as presented in the following segment of the program:

```
IDENTIFICATION DIVISION.
PROGRAM--ID.        STKVLN.
AUTHOR.             IGNOU.
INSTALLATION.       IGNOUCC, DELHI.
DATE--WRITTEN.      01/06/87.
DATE--COMPILED.     TODAY.
```

Environment Details

Detials about the computing environment such as configuration, Input and Output devices assignment of files to devices will have to be described in the ENVIRONMENT DIVISION of the program in CONFIGURATION and INPUT--OUTPUT sections respectively. The ENVIRONMENT DIVISION for our stock valuation program would have the description of computing environment and the statements assigning the stock master file and report file to the devices supported by the system as shown below:

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-- COMPUTER. IBMPC.
OBJECT--COMPUTER. IBMPC.
INPUT--OUTPUT SECTION.
FILE--CONTROL.
        SELECT STK--MASTER ASSIGN TO DISK
            ORGANIZATION IS LINE SEQUENTIAL.
        SELECT STK--VAL--RPT ASSIGN TO DISK
            ORGANIZATION IS LINE SEQUENTIAL.
```


These statements convey that the program will be compiled on IBMPC computer, to obtain object code that can be used on IBMPC for execution. The files used will be STK--MASTER and STK--VAL--RPT, both are sequential and assigned the device disk implying that both these are disk files. Notice that such assignment is done in the FILE--CONTROL paragraph of the section.

The sections and paragraphs used so far are standard and have specific semantics as illustrated above. In the forthcoming segments of code also we will be encountering such standard paragraph and section names.

Data Description

Structures of the files and other data variables used in the program will have to be described in the DATA DIVISION. There will have to be described separately in two standard sections namely FILE SECTION and WORKING--STORAGE SECTION.The FILE SECTION should be used to describe structural details of all the files referenced in the program. WORKING--STORAGE SECTION will have to be used for describing the details of temporary variables and record structures.

In our example the FILE SECTION will have the descriptions of STK--MASTER and STK--VAL--RPT files.

The data elements used to hold value of stock, total value, names of files, titles etc. will be described in the WORKING--STORAGE SECTION. Similarly record structures to compose title lines, report lines and footer lines etc., will also be described in this section.

DATA DIVISION for our stock valuation program is given below:

```
DATA DIVISION.
FILE SECTION
FD STK--MASTER
      LABEL RECORDS ARE STANDARD
      VALUE OF FILE--ID IS STKMST--NAME
      DATA RECORD IS STOCK-REC.
01 STOCK--REC.
      02 ITEM--CODE.
      05 MAIN--GR                 PIC 99.
      05 SUB--GR                  PIC 99.
      05 SR--NO                   PIC 99.
      02 DESCRIPTION              PIC X(24).
      02 STK--QTY                 PIC 9(4).
      PRICE--PU                   PIC 9(4) V99.
FD STK--VAL--RPT
      LABEL RECORDS ARE STANDARD
      VALUE OF FILE--ID IS SV--RPT--NAME
      DATA RECORD IS REPORT--REC.
01 REPORT--REC                    PIC X(70)
WORKING--STORAGE SECTION.
01 REPORT--LINE.
      02 FILLER             PIC X(3) VALUE SPACES.
      02 R--ITEM--CODE      PIC 9(6).
      02 FILLER             PIC X(3) VALUE SPACES.
      02 R--DESCRIPTION     PIC X(24).
      02 FILLER             PIC X(3) VALUE SPACES.
      02 R--STK--QTY        PIC Z, ZZ9.
      02 FILLER             PIC X(3) VALUE SPACES.
      02 R--PRICE--PU       PIC ZZ9.99.
      02 FILLER             PIC X(3) VALUE SPACES.
      02 R--VALUE           PIC ZZ,ZZ,ZZ9.99.
01 HEADER--REC.
      02 FILLER             PIC X VALUE SPACES.
      02 FILLER             PIC X(9) VALUE "ITEM CODE".
      02 FILLER             PIC XX VALUE SPACES.
      02 FILLER             PIC X(16) VALUE "ITEM DESCRIPTION".
      02 FILLER             PIC X(6) VALUE SPACES.
      02 FILLER             PIC X(8) VALUE "QUANTITY".
      02 FILLER             PIC XX VALUE SPACES.
      02 FILLER             PIC X(7) VALUE "PRICE".
      02 FILLER             PIC X(10) VALUE SPACES.
      02 FILLER             PIC X(5) VALUE "VALUE".
```

```
01 STKMST--NAME          PIC X(15).
01 SV--RPT--NAME         PIC X(15).
01 END--OF--DATA         PIC X(3).
01 DASH--LINE            PIC X(70) VALUE ALL "--".
01 TOTAL-VALUE           PIC 9(7) V99 VALUE ZERO.
01 ITEM--VALUE           PIC 9(7) V99.
01 R--TOTAL--REC.
   02 FILLER             PIC X(11) VALUE "TOTAL VALUE".
   02 FILLER             PIC X(41) VALUE SPACES.
   02 R--TOTAL-VALUE     PIC Z,ZZ,ZZ,ZZ9.99.
```

This segment of the program which presents the structural details of files and data variables requires explanations of several conventions used.

File Description: Starting with the FILE SECTION, the first entry used is FD, which stands for File Description. Through this entry programmer conveys whether the file is on auxiliary storage devices like magnetic tapes or disks with standard label records or it is on terminal devices like card reader or line printer which do not have label records. In addition to this he also specifies the name of the variable which holds the file name through FILE-ID clause. The name assigned to the data record is also specified in this entry through DATA RECORD clause. For example the STK--MASTER file is a disk file with standard label records; its data record is assigned the name STOCK--REC. Although the file is referenced with the name STK--MASTER in the procedure development, the variable STKMST--NAME holds the actual name of the file containing the stock data to be processed.

Record description: Just below the FD entry the programme should present the structural details of the data record as a hierarchical data structure using the level numbers and names of data fields. The characteristics of the data fields should also be specified for the elementary data elements. For example the STK--RECORD is composed of ITEM--CODE, DESCRIPTION, STK--QTY and PRICE--PU in the order described. These are the names which we have chosen to call the data fields presented in the record layout. It may be noticed that a level number 01 is assigned to STOCK-REC and the other data fields described above are assigned the higher number 02 since they are components of (hierarchically lower to) STOCK--REC. The same logic is applied in assigning the level numbers to sub fields of any of these data fields. For example the sub fields of ITEM--CODE, i.e. MAIN--GR, SUB--GR, and SR--NO are assigned the level numbers 05. It may also be noticed that the data elements like STOCK--REC and ITEM--CODE did not contain the description of their type characteristics since they are not elementary data items.

Description of Elementary Data Items: The elementary data items i.e., items which do not have further sub components should have description of their type and size etc. through PICTURE clause. The picture clause includes the letters PICTURE or PIC followed by the description of the field.

The field description includes the information as whether the field is numeric, alphabetic, or alphanumeric; whether it is computational or display; whether it contains editing characters; whether it contains a decimal point; and whether a numeric field can contain a negative value etc. Some examples of field descriptions are presented below:

Picture character	Purpose	Example
9	represent the presence of numerical digit	99 9(6) or 999999
V	indicate the position of assumed decimal point	99 V99 9(4) V99
P	indicate the position of decimal point in cases in which decimal point is not within the number	99PPPV VPP9999
S	to designate a numeric field which is assigned.	S999V99 S9(5)V99
X	to denote the positions of alphanumeric characters in the field.	XXXX or X(4)
A	to indicate the presence of alphabatic characters or blanks in the field.	AAAA or A(4)

Editing Symbols: Edit symbols are used in the picture description to present the data values in a convenient form in the output. The data-names whose picture description includes edit symbols cannot be used in arithmetic operations since such fields contain alphanumeric characters rather than simple numeric characters. Following are some picture characters used as edit symbols:

Picture character	Purpose	Example
$	to indicate where $ sign should appear in the output. When floated (repeated more than once) indicates its insertion left to the most significant digit of field.	$999 $$$$ $$$$.$$ SSS.99
	indicate the position where " " should be inserted in the data field	9,999.99 $99,999,99
	indicates the position of the decimal point in the data field.	999.99 $$,$$$.$$
Z	replace leading zeroes with balnks	Z99 ZZ,ZZZ.ZZ
CR	insert chars 'CR' to the right of number if it is negative	$$,$$$.99CR 999.99CR
DB	insert chars 'DB' to the right of number, if it is negative	ZZ,ZZZ.ZZDB 9999.99DB
*	replace leading zeroes by'*' ('*' in this format is known as cheque protection symbol.)	$ x x x x.99

+	to have the sign of the number included either to left or right can be used as floating symbol	+ 999.99 999.99+ + + + + . + +
-	to have the sign included only if the number is negative else blank can be used as floating symbol	-999.99 999.99- - - - - . - -
B	blank insertion	ABABX(8)
0	zero insertion	9990000
/	stroke insertion	99/99/99

Depending on the storage and output requirements of the application being programmed, one can build the desired picture strings for various data fields using the above described symbols.

Naming the Data Elements: Throughout the abvove discussion, we have identified the data elements with some data-names. The data-names can be coined at the discretion of the programmer, except that there are certain rules that must be followed:

A data-name can be up to 30 characters and can include alphabetic characters, numeric characters, and hyphens. At least one character must be alphabetic. A hyphen if used, must be embedded; that is it cannot be the first or last character of the data-name. Blanks cannot be included in the data-names. Within the above rules the programmer may use any data-name, with the exception of the COBOL reserved words.

Certaindata fields which are not required to be referenced by any specific name may be assigned the COBOL standard name FILLER. Any number of FILLERs can be used in the DATA DIVISION.

Assigning Initial Values: The description of data element may also include the VALUE clause to initial values to these data elements.

Examples:

```
01   DASH--LINE              PIC X(70) VALUE ALL "--".
01   TOTAL--VALUE            PIC 9(7) V99 VALUE ZERO.
01   HEADER--REC.
     02   FILLER             PIC X VALUE SPACES.
     02   FILLER             PIC X(8) VALUE "ITEMCODE
```

Processing Details

In a COBOL program, the processing instructions have to be presented in its PROCEDURE DIVISION. In the example of stock valuation, the processing steps would be:

i) Prompting the user for the names of stock data file and report file, accepting the user supplied names and opening these files to perform input and output operations respectively.

ii) Writing the title lines on the output report.

iii) Reading a record from stock data file, computing the value of the stock and writing a record on the output file giving out the item details, its stock and computed value.

iv) Performing the step (iii), repeatedly, till we reach the end of file on the stock data file.

v) Writing out the total values and closing lines on the output file and closing the files.

These steps expressed in COBOL would be as follows:

```
PROCEDURE DIVISION.
START-PARA.
      DISPLAY "ENTER THE NAME OF STOCK DATA FILE".
      ACCEPT STKMST-NAME.
      DISPLAY "ENTER THE NAME OF OUTPUT FILE".
      ACCEPT SV-RPT-NAME.
      OPEN INPUT STK-MASTER.
        OUTPUT STK-VAL-RPT.
      MOVE "NO" TO END-OF-DATA.

PROCESS-PARA
      WRITE REPORT-REC FROM DASH-LINE.
      WRITE REPORT-REC FROM HEADER-REC.
      WRITE REPORT-REC FROM DASH-LINE.
      PERFORM READ-STOCK-REC.
      PERFORM PROCESS-RECORD UNTIL END-OF-DATA = "YES".
      WRITE REPORT-REC FROM DASH-LINE.
      MOVE TOTAL-VALUE TO R-TOTAL-VALUE
      WRITE REPORT-REC FROM R-TOTAL-REC.
      WRITE REPORT-REC FROM DASH-LINE.
      CLOSE STK-MASTER STK-VAL-RPT.
      STOP RUN.

READ-STOCK-REC.
      READ STK-MASTER AT END MOVE "YES" TO END-OF-DATA.
PROCESS-RECORD.
      MOVE ITEM-CODE TO R-ITEM-CODE.
      MOVE DESCRIPTION TO R-DESCRIPTION.
      MOVE STK QTY TO R-STK-QTY.
      MOVE PRICE-PU TO R-PRICE-PU.
      MULTIPLY STK-QTY BY PRICE-PU GIVING ITEM-VALUE.
      MOVE ITEM-VALUE TO R-VALUE.
      ADD ITEM-VALUE TO TOTAL-VALUE.
      WRITE REPORT-REC FROM REPORT-LINE.
      PERFORM READ-STOCK-REC.
```

In this segment of the program which consisted of PROCEDURE DIVISION, we have used a number of COBOL statements and sentences involving appropriate verbs to convey the operation to be performed on the data elements or records of the files referenced. The COBOL verbs are more English like and the operation to be performed can be derived from the literary meaning of the verbs used in the sentences.

Following are some COBOL verbs alongwith their meanings:

COBOL verb	Meaning
1. Input-ouput verbs	
OPEN	Open the file for the specified operation (INPUT or OUTPUT).
CLOSE	Close the specified file. After this operation the READ or WRITE operations cannot be performed on the referenced file.
READ	Read a record from the referenced file.
WRITE	Write out the referenced record on to its file.
ACCEPT	Accept (read) data from user terminal.
DISPLAY	Display (write) the data values on the user terminal.
2. Data Movement verb:	
MOVE	Move the data from one data element to another. The result of move operation depends on the types and sizes of sending and receiving fields. The group move or move into a alphabetic or alphanumeric data field, aligns the values left justified. Moving a numeric integer to another numeric integer field results in right justification. Movement of numeric non-integer is done with decimal alignment. Movement of numbers to edited fields is guided by the edit symbols used in the picture description of the receiving fields. If the receiving field is short, the result will be truncated appropriately. Example: MOVE ITEM-VALUE TO R-VALUE moves the value 300000 of ITEM-VALUE as 30,000-00 into R-VALUE because of the numeric edited picture ZZ,ZZ,ZZ99.99 of R-VALUE and picture 9(7) V99 of ITEM-VALUE.
3. Procedure control verb:	
PERFORM	Branch to the indicated paragraph, execute the instruction(s) in the paragraph, and then return to the statement immediately following the PERFORM instruction.

4. Arithmetic verbs:

ADD SUBTRACT MULTIPLY DIVIDE	Execute the specified arithmetic operation.

In our stock valuation program in the START-PARA we have statements which first prompt for the names of the files using the DISPLAY verb. User responses to these prompts are accepted into appropriate variables using ACCEPT verb. The stock master file is opened for input thereby permitting only reading operations on the STK-MASTER file. The stock valuation report, STK-VAL-RPT, is opened for output so that a new file is created and records are appended to it as we execute statements involving WRITE verb on this file. The variable END-OF-DATA is initialised with a value "NO", in this paragraph.

The next paragraph of the procedure writes the title lines on the report file. It then executes a paragraph which reads a record from STK-MASTER file. The paragraph PROCESS-RECORD, moves the data fields of the current stock record into corresponding edited fields in the report record, computes item value by multiplying stock quantity with price per unit, builds up the total stock value, writes report record to the output file and reads in the next stock record. This paragraph is executed repeatedly until end-of-data condition is encountered on the stock master file. Such condition is setup in the READ-STOCK-REC paragraph, by moving "YES" to the variable END-OF-DATA on encountering the end of file status on STK-MASTER file.

After performing (executing) the PROCESS-RECORD para until an end of data condition is encountered, the footer lines including the one giving the total value of items in the stock are moved to the output file. The files are then closed using CLOSE verb and the program execution is terminated using STOP RUN statement.

It may be noticed that the PROCEDURE DIVISION statements are almost English-like. With meaningful names chosen for paragraphs and data elements (records and fields) a procedure division can be programmed to be self explanatory.

Sample Input and Output Files

We present below a sample STK-MASTER file and the corresponding STK-VAL-RPT file generated by executing the COBOL program illustrated above:

LISTING OF STK-MASTER FILE

```
123456TUBE LIGHT 40W                          10003000
023451FILAMENT BULB 60W                       02000500
```

LISTING OF STK-VAL-RPT FILE CREATED BY PROGRAM

ITEM CODE	DESCRIPTION	QUANTITY	PRICE	VALUE
123456	TUBE LIGHT 40W	1,000	30-00	30,000-00
023451	FILAMENT BULB 60W	200	5-00	1,000-00
TOTAL VALUE				31,000-00

20.6 OTHER FEATURES OF COBOL

Through the example of stock valuation we have introduced a number of features of COBOL. The features not covered so far include table handling, string manipulation, advanced sequence controls, advanced file structures, report writer and screen handling.

Table Handling

COBOL permits the arrays to be used as elements of records. The language offers powerful table handling features which are described in the subsequent sections.

Description of Table Data Structures

The OCCURS clause can be used to declare a vector, individual elements of which may be an elementary data item or another record or array. For example to set up a table of sales values in say four zones, we would have the following data description:

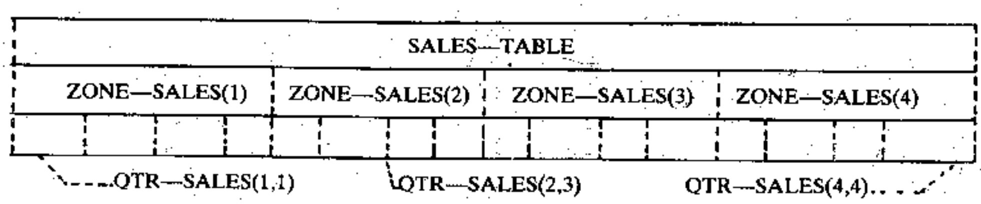
```
01    SALES-TABLE.
          05 SALES-VALUE      OCCURS 4 TIMES      PIC 9(6) V99.
```

Through this description we define a record, SALES-TABLE which consists of the vector of four numeric non-integer data elements with six integer digits and two decimal digits in size. Reference to these data elements can be made by using an appropriate subscript with the data-name SALES-VALUE. For example you want to refer to the sales value of third zone, you should make the reference as: SALES-VALUE (3).

We can also define a two dimensional table to store the sales values zone-wise and quarter-wise as given below:

```
01    SALES-TABLE.
      02    ZONE-SALES             OCCURS 4 TIMES.
            05    QTR-SALES        OCCURS 4 TIMES        PIC 9(6) V99.
```


The layout of SALES-TABLE would be as given below:



COBOL permits up to three levels of nesting of OCCURS, thereby allowing the use of up to three dimensional arrays.

COBOL also permits the use of mixed structures containing both records and arrays as presented in the example below.

```
01    EMPLOYEE-DATA.
          02    EMP-CODE              PIC 9(6).
          02    EMP-NAME              PIC X(30).
          02    EXPERIENCE OCCURS 10    TIMES.
```

```
03   ORGN-NAME    PIC X(20).
03   YEARS-SRVD   PIC 99V9.
03   LST-DSGN     PIC X(20).
03   LST-GR-SAL   PIC 9(5).
```

In this example we have the EMPLOYEE-DATA record consisting of the data fields EMP-CODE and EMP-NAME and an array EXPERIENCE of 10 records each consisting of the fields ORGN-NAME, YEARS-SRVD, LST-DSGN and LST-GR-SAL.

In this data structure, if we want to refer to LST-GR-SAL of the third experience record of an employee, we should address it as LST-GR-SAL(3), References to other data fields of EXPERIENCE will have to be made in a similar manner. The reference EXPERIENCE(5) addresses the fifth record of the EXPERIENCE table. This implies that operations such as MOVE treat all the four sub fields of the referenced record as one unit.

Operations on Table Data Structure

COBOL provides two powerful operations on tables. These are SEARCH and extended form of PERFORM.

Through SEARCH feature one can programme the searching operation on a data table with a single sentence. The following example illustrates SEARCH feature:

Consider a data processing task in which we read the data of about 300 items into a table. We are then required to find the description and price of item, given its code. Assume that the table is sorted on the item code order. Since the table will be searched by reference to the item code, the record description entry can be written in such a way as to indicate the fact that the table is sorted on item code in ascending order. Further record description be so written that reference to table entries can be made by use of a variant of subscript concept, called an index as follows.

```
01   ITEM-TABLE.
     02 ITEM-DATA OCCURS 300 TIMES
                          ASCENDING KEY IS ITEM-CODE
                          INDEXED BY IT-IX
            03   ITEM-CODE     PIC X(8).
            03   DESCRIPTION   PIC X(30).
            03   PRICE-PU      KPIC 9(4) V99.
```

Suppose RQ-ITEM is a field that contains a code for which we want to find the description and price-pu. The following PROCEDURE DIVISION statements perform the required task.

```
SEARCH ALL ITEM-DATA AT END PERFORM NOT-FOUND
                    WHEN ITEM-CODE (IT-IX) = RQ-ITEM
                    MOVE DESCRIPTION (IT-IX) TO RQ-DESCR
                    MOVE PRICE-PU (IT-IX) TO RQ-PRICE.
```

Other form of SEARCH enables searching an unsorted table. The same example with search into unsorted ITEM-TABLE would be as follows:

DATA DIVISION entry:

```
01 ITEM-TABLE.
      02    ITEM-DATA OCCURS 300 TIMES INDEXED BY IT-IX.
               03    ITEM-CODE       PIC X(8).
               03    DESCRIPTION     PIC X(30).
               03    PRICE-PU        PIC 9(4) V99.
```

PROCEDURE DIVISION entry:

```
    SET IT-IX TO 1.
    SEARCH ITEM-DATA AT END PERFORM NOT-FOUND
               WHEN ITEM-CODE (IT-IX) = RQ-ITEM
               MOVE DESCRIPTION (IT-IX) TO RQ-DESCR
               MOVE PRICE-PU (IT-IX) TO RQ-PRICE.
```

The value of IT-IX is initialised to 1 by statement SET IT-IX TO 1. Since the table has not been sorted, a linear search will be performed beginning with the first entry of the table.

Sequence Control

In the PROCEDURE DIVISION each paragraph and section begins with a label which may be used as object of GOTO or PERFORM control transfer. Execution follows the physical statement sequence without regard to sentence, paragraph, or section boundaries unless a GOTO, PERFORM, or IF statement is used to explicitly transfer control. Paragraphs or sections may be used as simple subprogrammes, but control is also allowed to flow into the same paragraphs or sections in the normal sequence of execution.

A simple GOTO may transfer control to any paragraph or section label. A multiple branch using a list of labels and a computed subscript is also provided in COBOL.

An if_ then_ else is provided in COBOL using the syntax: If < condition> <statement sequence> ELSE < statement-sequence> The < condition > specified may be relational or boolean expression.

The PERFORM statement in COBOL serves both as an iteration statement and as simple parameterless subprogram call. In its simplest form which we have seen,

PERFORM L1.   or.   PERFORM L1 THRU Ln.

it causes execution of designated paragraph (L1) or sequence of paragraphs (L1 thru Ln) as a simple parameterless subprogram.

The other forms of PERFORM:

PERFORM L1 THRU Ln   k TIMES

serves to call the subprogram repeatedly k times. The statement:

PERFORM L1 THRU Ln UNTIL < condition >.

iterates execution of subprogram until the condition evaluates to true.

The statement:

```
    PERFORM L1 THRU Ln
          VARYING I FROM I1 BY S1 UNTIL < condition-1 >
          AFTER J FROM I 2 BY S2 UNTIL < condition-2 >
          AFTER K FROM I 3 BY S3 UNTIL < condition-3 >.
```

allows repeated execution of the subprogram with from one to three indices moving through integer ranges. In the above statement, index K cycles completely for each step in the value of J and so on.

In addition to the basic statement sequence-control structures mentioned above, many COBOL statements provide for execution of one or more statements when a special condition arises during execution of the base statement. For example each arithmetic statement may contain a suffix designating actions to be taken in case of a size error, an error caused by the result of the arithmetic operation being too large to fit in the designated result variable location. Thus the statement:

```
ADD A TO B ON SIZE ERROR PERFORM ERR-PARA-1.
```

would cause execution of subprogram ERR-PARA-1 should the sum of the values of A and B exceed the space allocated for B.

Other conditional checks include the end-of-file checking on READ statements and end-of-page checking on WRITE statements.

String Manipulation

COBOL offers string manipulation features through STRING, UNSTRING and INSPECT verbs.

STRING and UNSTRING verbs are designed to facilitate transfer of data from several sources into one destination and from one destination to many destinations, respectively.

Example 1

Assume that we want to print a report which lists a customer name in column 5-25, a city name starting with column 31, one blank space and then the PIN code. The source of data is CUSTOMER-RECORD with the following description.

```
01 CUSTOMER-RECORD.
    02    CUSTOMER-NAME    PIC X(20).
    02    CUST-ADDRESS     PIC X(100).
    02    CITY-STATE       PIC X(20).
    02    PIN              PIN 9(6).
```

The data CITY-STATE is recorded such that the city name is followed by a comma a space, and then the state name. e.g., AHMEDABAD, GUJARAT. The output record is described as:

```
01    OUT-REC              PIC X(132).
```

We use STRING verb as follows:

```
MOVE SPACES TO OUT-REC
MOVE 5 TO STARTING-PLACE.
STRING CUSTOMER-NAME DELIMITED BY SIZE,
    CITY-STATE DELIMITED BY ','
    SPACE
    PIN DELIMITED BY SIZE
INTO OUT-REC WITH POINTER AT STARTING-PLACE
```

Example 2

Suppose the data is recorded in free from (without predefined fields) as follows:

VIJAY D SHAH, 3,12,2.75

ANAND K TIWARI, 4,15,3.25

Notice that the name fields are separated by one or more blank spaces, the commas separate the remaining three fields. We would like to move these data fields to the following fixed format record:

```
01   STUDENT-DATA.
     02   FIRST-NAME        PIC X(20).
     02   MIDDLE-NAME       PIC X(15).
     02   LAST-NAME         PIC X(20).
     02   CATEGORY          PIC 9.
     02   UNITS-REGD        PIC 99.
     02   GPA               PIC X(4).
```

Assuming that the source data is in

```
01   FREE-FORM-RECORD       PIC X(132).
```

We can write:

```
UNSTRING FREE-FORM RECORD
         DELIMITED BY ALL SPACES OR ',
INTO     FIRST-NAME
         MIDDLE-NAME
         LAST-NAME
         CATEGORY
         UNITS-REGD
         GPA.
```


At times we need to access and manipulate individual characters in a field. COBOL provides INSPECT verb to accomplish such character manipulations. The following examples illustrate the use of INSPECT verb:

1. Suppose we want to replace all leading blanks by zeroes in field called AMOUNT.

   ```
   INSPECT AMOUNT REPLACING LEADING ' ' BY '0'
   ```

2. Count the number of dollar signs in TEST and replace all dollar signs after the first by asterisks:

   ```
   INSPECT AMOUNT TALLYING COUNT-A FOR ALL '$'
                  REPLACING ALL '$' BY '*' AFTER INITIAL '$'.
   ```

Advanced File Structures

COBOL supports Indexed and relative files in addition to the normal sequential files. Sequential file organisation indicates that the records in the file are positioned in a sequential order according to one of the data fields in the record. Indexed file organisation is the one in which the records are filed sequentially, but a table is available which identifies the location of the groups of records, thereby reducing the access time. Relative file organisation is such that the logical order and physical order do not necessarily correspond with one another. For such file a technique or rule is required to locate the record.

Sequential Files: In COBOL, the sequential files can be opened for input, output, input & output and extend operations. When a file is opened for input & output operation, COBOL allows replacement of an existing record with the help of REWRITE verb. EXTEND mode can be used to add records at the end of a sequential file.

Indexed Files: In order to use indexed files the SELECT statement in the ENVIRONMENT DIVISION has to include following details:

```
SELECT < file-name > ASSIGN TO DISK
       ORGANISATION IS INDEXED
                         | SEQUENTIAL |
       ACCESS MODE IS    | RANDOM     |
                         | DYNAMIC    |
       RECORD KEY IS < data-name >
[ALTERNATE RECORD KEY IS < data-name-2 > [WITH DUPLICATE]]
```

In this format, ACCESS MODE clause specifies the way in which records in the file will be accessed. ACCESS MODE IS SEQUENTIAL specifies that records will be accessed in ascending order of the record key. RANDOM option specifies that the order in which records are accessed will be controlled by the programmer. This control is accomplished by moving the value of the key of the desired record into the RECORD KEY field and issuing an input-output command (READ, WRITE, REWRITE, DELETE). The DYNAMIC option allows the programmer to change at will from sequential access to random access using appropriate forms of input-output statements.

RECORD KEY rferences a data-name which must be a field within the record description of the file. When the ALTERNATE RECORD option is used, we access the records either on the basis of the prime key specified in the RECORD KEY clause or on the basis another ALTERNATE RECORD KEY. The DUPLICATES phrase specifies that the value of the associated alternate record key may be duplicated with any of the records in the file.

An indexed file can be opened as INPUT, OUTPUT or I-O. Reading records from an indexed file is accomplished using one of the two formats:

```
READ < file-name > [NEXT] RECORD [INTO < indentifier >]
      [AT END < imperative statement >].
READ < file-name > RECORD [INTO < identifier >]
      KEY IS < data-name >
      [INVALID KEY < imperative statement >]
```

Relative Files: The relevant ENVIRONMENT DIVISION format for relative files is as follows:

```
SELECT < file-name > ASSIGN TO DISK
     ORGANISATION IS RELATIVE
                      | | SEQUENTIAL [RELATIVE KEY IS < data-name >]
     ACCESS MODE IS   | | RANDOM   |
                      | | DYNAMIC  |     RELATIVE KEY IS < data-name >
```

Records are read from a relative file using one of the two formats:

```
READ < file-name > RECORD [INTO < identifier >]
        [AT END < imperative-statement >]
READ < file-name > RECORD [INTO < identifier >]
        [INVALID KEY < imperative statement >]
```

Data Base Management Systems: In order to provide interface with Data Base Management Systems (DBMS), manufacturers offer COBOL Data Manipulation Language (DML) processors which include enhanced instruction set to carry out data manipulation operations through COBOL programmes. COBOL DML processors provide natural interface to both network and relational data base management systems.

Report Writer Feature

Report writer feature of COBOL facilitates production of reports by specifying the physical appearance of a report rather than requiring specification of detailed procedures necessary to produce the report. The report writer is part of the COBOL language and can be incorporated in any programme.

To use report writer feature, the ENVIRONMENT DIVISION and DATA DIVISIONS of the programs should include the description of the desired report file and its format. For example the description of a typical report file to generate invoice summary report may have the following description.

```
ENVIRONMENT DIVISION entry:
FD SUMMARY-FILE
        LABEL RECORDS ARE STANDARD
        REPORT IS INVOICE-SUMMARY.
```

This description is same as that for any file except that REPORT IS clause is used in the place of DATA RECORD IS clause.

The DATA DIVISION entries describing the report should be described in a new section REPORT SECTION, immediately following the WORKING-STORAGE SECTION. The Report Description (RD) entry similar to FD must be used to describe the report details as shown below:

```
REPORT SECTION.
RD INVOICE -SUMMARY
            PAGE LIMIT IS 50 LINES
            HEADING 3
            FIRST DETAIL 5
            LAST DETAIL 45.
01 TYPE IS REPORT HEADING
            NEXT GROUP NEXT PAGE.
    02      LINE NUMBER IS 8 COLUMN NUMBER IS 25
            PICTURE IS A(24)
            VALUE IS 'ABC TEXTILE COMPANY LTD'
    02      LINE NUMBER IS PLUS 2
            COLUMN NUMBER IS 30
            PICTURE IS X(20)
            VALUE IS 'AHMEDABAD-380015'
```

```
01    TYPE IS PAGE HEADING

01    INVOICE-DATA TYPE IS DETAIL

01    TYPE IS PAGE FOOTING

01    TYPE IS REPORT FOOTING
```

It may be noted that each 01 level entry in the above example introduces a report group in a manner similar to that of record descriptions in an ordinary file.

In the PROCEDURE DIVISION, the verbs INITIATE, GENERATE and TERMINATE can be used to obtain the report as illustrated below:

```
    PROCEDURE DIVISION.
        OPEN INPUT IN-FILE
             OUTPUT SUMMARY-FILE.
        MOVE 'NO' TO END-OF-DATA.
        READ IN-FILE RECORD AT END MOVE 'YES' TO END-OF-DATA.
        INITIATE SUMMARY-REPORT.
        PERFORM PREPARE-REPORT UNTIL END-OF-DATA = 'YES.
        TERMINATE SUMMARY-REPORT.
        CLOSE IN-FILE SUMMARY-FILE.
        STOP RUN.
PREPARE-REPORT.
        GENERATE SUMMARY-REPORT.
        READ IN-FILE RECORD AT END MOVE 'YES' TO END-OF-DATA.
```

This program reads an input file, IN-FILE consisting of invoices and generates the report file, SUMMARY-FILE, as per the description given in the REPORT SECTION. The verbs INITIATE and TERMINATE are used to initiate and terminate the report generation process respectively.

Screen Handling

Recent versions of COBOL processors screen handling as one of the attractive features which facilitates development of on-line interactive systems with convenient screen formats. Normally such feature is offered through new SCREEN SECTION in DATA DIVISION following the WORKING-STORAGE SECTION. In this section the format of desired screen layout alongwith positions where input data is to be accepted or output data to be displayed is described. For example the following SCREEN SECTION entry TITLE-SCR would display the titles at the specified line numbers and column positions of the screen. DISPLAY TITLE-SCR statement of PROCEDURE DIVISION would not only display these details but also displys the values of the data items referenced.

```
SCREEN SECTION.
01    TITLE-SCR
      02    LINE 5   COLUMN 5 REVERSE VIDEO VALUE 'ITEM CODE'.
      02    LINE 5   COLUMN 15 PIC 9(6) FROM ITEM-CODE.
      02    LINE 8   COLUMN 5 REVERSE VIDEO VALUE 'DESCRIPTION'.
      02    LINE 8   COLUMN 20 PIC X(24) FROM DESCRIPTION.
      02    LINE 10 COLUMN 5 REVERSE VIDEO VALUE 'QUANTITY'.
      02    LINE 10 COLUMN 15 PIC 9(4) FORM STK-QTY.
01    ENQ-SCR.
      02    BLANK SCREEN.
      02    LINE 5 COLUMN 5 BLINK VALUE 'ITEM CODE?'
      02    LINE 5 COLUMN 15 PIC 9(6) TO ITEM-CODE.
```

In this example, when DISPLAY TITLE-SCR is executed, the titles ITEM-CODE, DESCRIPTION and QUANTITY will appear in reverse video at line numbers 5,8 and 10 respectively all starting from column number 5. Values of the data elements ITEM-CODE, DESCRIPTION, and STK-QTY are also displayed on the corresponding lines at column numbers 15,20 and 15 respectively as per the picture description of these items i.e., 9(6), X(24) and 9(4). Similarly when ACCEPT ENQ-SCR statement is executed, the screen is first cleared and the message 'ITEM CODE?' appears on line number five, starting from column 5 in blink mode. The user response is accepted from column 15 of line 5 and is assigned to the data element, ITEM-CODE.

This briefly explain screen handling in COBOL. Extended features and variations do exist from compiler to compiler.

20.7 SUMMARY

In this unit we have discussed programming in COBOL. After introducing briefly COBOL in the beginning, the actual programming steps starting from Identification Division to Procedure Division, were discussed. We have also developed some simple programmes as examples which would help you for writing elementary programmes. Towards the end, we have also discussed screen handling, which is an advanced and attractive feature of COBOL.

20.8 SELF-ASSESSMENT EXERCISES

1. What kind of information do you need for feeding into 'Environment Division' details? Explain with suitable examples.
2. What are the COBOL verbs, explain their place of use in the programming giving appropriate examples.
3. Write a simple programme in COBOL for preparing payroll of employees of your organisation.

20.9 FURTHER READINGS

American National Standard Programming Language COBOL-X3. 23-1974, published by ANSI.

Micro soft COBOL Compiler User's Guide, Micro Soft, Redmond, USA.

Micro Soft COBOL Reference Manual, Micro Soft, Redmond, USA.

Terrace W. Pratt, 1983. *Programming Languages Design and Implementation,* Prentice Hall of India: New Delhi.

Yourdon, Gane, Sarson and Lister, 1979. *Learning to Program in Structured COBOL, Parts 1 & 2,* Prentice Hall of India: New Delhi.

Philippakis and Kazmier, 1978. *Information Systems through COBOL,* 2nd Edition., McGraw Hill Kogakusha Ltd.: Tokyo.

Elder, John 1984. *Construction of Data Processing Software,* Prentice Hall International Englewood-Cliffs.