Higher Secondary Course

COMPUTER APPLICATIONS
Commerce

CLASS - XI

Government of Kerala

DEPARTMENT OF EDUCATION

State Council of Educational Research and Training (SCERT); Kerala

THE NATIONAL ANTHEM

Jana-gana-mana adhinayaka, jaya he
Bharatha-bhagya-vidhata.
Punjab-Sindh-Gujarat-Maratha
Dravida-Utkala-Banga
Vindhya-Himachala-Yamuna-Ganga
Uchchala-Jaladhi-taranga
Tava subha name jage,
Tava subha asisa mage,
Gahe tava jaya gatha.
Jana-gana-mangala-dayaka jaya he
Bharatha-bhagya-vidhata.
Jaya he, jaya he, jaya he,
Jaya jaya jaya, jaya he!

PLEDGE

India is my country. All Indians are my brothers and sisters.

I love my country, and I am proud of its rich and varied heritage. I shall always strive to be worthy of it.

I shall give respect to my parents, teachers and all elders and treat everyone with courtesy.

I pledge my devotion to my country and my people. In their well-being and prosperity alone lies my happiness.

Foreword

Dear learners,

The syllabus of Computer Applications for the Commerce discipline has been revised in view of the fact that computer is used currently as a tool for various applications, especially in the field of e-Banking, e-Commerce, e-Governance, etc. Higher studies and placements, at present, greatly demand human resources with adequate knowledge in computer applications and information technology. The syllabus and the text book have been designed in order to provide options for students completing the course to pursue higher studies in Computer Applications and Commerce or make contributions in either of the two.

The text book, designed in accordance with the syllabus for the academic year 2014 - 15, begins with the concept of data processing and presents a variety of computer hardware and software for various applications of the times. Besides, the concerns and remedies for e-Waste management are also highlighted. A wide coverage is given for computer network and Internet technology, underlining the virtues and vices in their use. An exclusive section on applications of Information Technology and ICT enabled services are also incorporated considering their widespread usage and career opportunities.

The syllabus and the textbook also ensure a strong foundation to construct and enhance the problem solving and programming skills of the learner. The multi-paradigm programming language C++ is presented to develop programs which enable computers to manage different real life applications effectively.

I hope this book will meet all the requirements for stepping to levels of higher education and pave the way to the peak of success.

Dr P. A. Fathima
Director
SCERT, Kerala

Textbook Development Team

Joy John
HSST, St. Joseph's HSS,
Thiruvananthapuram.

Vinod V.
HSST, NSS HSS, Prakkulam, Kollam.

Roy John
HSST, St. Aloysius HSS,
Elthuruth, Thrissur,

Aboobacker P.
HSST, Govt. GHSS,
Chalappuram, Kozhikode.

Mustafa Shamsul Haq K. K.
HSST, GHSS Kuthuparamba,
Kannur.

Prasanth P. M.
HSST, St. Joseph's Boys' HSS,
Kozhikode.

Asees V.
HSST, GHSS Velliyode, Kozhikode.

Rajamohan C.
HSST, Nava Mukunda HSS
Thirunavaya, Malappuram.

Shajan Jos N.
HSST, St. Joseph's HSS, Pavaratty,
Thrissur.

Sai Prakash S.
HSST, St. Thomas HSS,
Poonthura, Thiruvananthapuram.

Experts

Dr Lajish V. L.
Assistant Professor, Dept. of Computer Science, University of Calicut.

Dr Madhu S. Nair
Assistant Professor, Dept. of Computer Science, University of Kerala.

Madhu V. T.
Director, Computer Centre, University of Calicut.

Dr Binu P. Chacko
Associate Professor, Dept. of Computer Science,
Prajyoti Niketan College, Pudukad.

Dr Sushil Kumar R
Associate Professor, Dept. of English, D.B College, Sasthankota.

Dr Vineeth K. Paleri
Professor, Dept. of Computer Science and Engineering, NIT, Kozhikode

Maheswaran Nair V.
Sub Divisional Engineer, Regional Telecom Training Centre, Thiruvananthapuram

Artists

Sudheer Y.
Vineeth V.

Academic Co-ordinator

Dr. Meena S.
Research Officer, SCERT.

Contents

Icons used in this textbook

Let us do

Check yourself

Information box

Lab activities

Learning outcomes

1

Key concepts

Fundamentals of Computer

Computers have now become an integral part of our daily life. People use computers for a variety of reasons and purposes. Be it education, business, entertainment, communication, government service or transportation, computers are inevitable today. As far as students are concerned, computers are used for learning different subjects effectively and for carrying out learning activities apart from their primary functions of computing. Try to recollect the situations where we used computers and identify the benefits you got from it. Therefore it is essential to know more about computers and its applications. This chapter presents the concepts of data processing and functional units of computer. Different data representation methods used in computers are also discussed in this chapter.

1.1 Data and Information

Many of us are familiar with the terms - data and information. We often use these terms interchangeably in our daily life. But there exists fundamental differences between these two. As part of our attempt to explore the field of computers, it is very essential to distinguish between these two terms.

Figure 1.1 shows a portion of the class diary of a teacher. Can you make out the words and numbers? Since it is a teacher's diary, these

can be the names of some students. What do the numbers mean? One cannot be sure. It can be the marks scored by students in tests, their attendance for some months, or something similar. We call these facts and figures data, because they do not give a complete idea. **Data** denotes raw facts and figures such as numbers, words, amount, quantity etc. that can be processed or manipulated.

Fig. 1.1 : Sample data

If these facts and figures were written as shown in Figure 1.2, there would be no confusion as to what they mean. It is clear that the figures show the scores obtained by

Fig. 1.2 : Sample information

students in Continuous Evaluation (CE) activities. We can see that when the data is arranged in a meaningful way, we get a clear-cut idea about these facts and figures. This is known as **information**. It is a meaningful and processed form of data.

Information may also act as data in other contexts. In our example of preparation of CE scores of students, the teacher converts these numbers into a consolidated score out of 10. Similarly, after the evaluation of answer scripts of the public examination, each of these students will be awarded a score out of 40. During the preparation of results of examination, the scores of all the subjects are collected and corresponding grades are granted. The personal details and grades are put in an appropriate format with suitable labels and it becomes the mark sheet of the student, which is again information.

Figure 1.3 shows the score sheet of a student issued after the public examination. It contains personal details of the student and the grades obtained in each subject. The personal details like Anitha Mohan, Female, 13/04/1997, etc. are printed against proper labels such as Name, Sex, Date of Birth, etc. Here the facts and figures that represent the personal details and grades are the data. When these data are specified with suitable labels, it becomes information about that student. Thus we can say that the score sheet contains the information about the performance of a student in an examination. This information adds to our knowledge about the level of achievement of the students in various subjects. It also helps them to take decisions about their higher studies or to plan their future.

No. N 389066

9HIH6G9GH9EEE9332583

GOVERNMENT OF KERALA
GENERAL EDUCATION DEPARTMENT
SECONDARY SCHOOL LEAVING CERTIFICATE

Register Number: 121367 Month & Year: MARCH 2013 No. of Chances: 1

This is to certify that the candidate herein has appeared for the SSLC Examination and secured the following grades

Subject	Grade	Grade in words
First Language Paper - I (MALAYALAM)	A+	A Plus
First Language Paper - II (MALAYALAM)	A+	A Plus
English	A+	A Plus
Hindi	A	A Only
Social Science	A	A Only
Physics	A+	A Plus
Chemistry	A	A Only
Biology	A	A Only
Mathematics	A	A Only
Information Technology	A	A Only

1. Name of Candidate : ANITHA MOHAN
2. Name in Regional Language : അനിത മോഹൻ
3. Name of School
   St. Joseph's H S S
4. Admission No. : 1377
5. Date of Birth (in figures) : 13/04/1997
           (in words) : THIRTEENTH APRIL NINETEEN NINETY SEVEN
6. Sex : FEMALE
7. Religion & Caste : HINDU, NAIR
8. Whether SC/ST/OEC/OBC/GL : GENERAL
9. Nationality : INDIAN
10. Name of Mother : JAYANTHI L
11. Name of Father : RADHAKRISHNAN K S
12. Name of Guardian : RADHAKRISHNAN K S
13. Home Address : ANITHA BHAVAN,
   MUTTATHARA
   VALLAKKADAVU P O,
   THIRUVANANTHAPURAM

RANGE OF GRADES

A+ 90% and above: Outstanding | B 60% - 69% : Good | D+ 30% - 39% : Marginal
A 80% - 89% : Excellent | C+ 50% - 59% : Above Average | D 20% - 29% : Need Improvement
B+ 70% - 79% : Very Good | C 40% - 49% : Average | E Below 20% : Need Improvement

Eligibility for higher studies - Minimum D+ grade for each paper

ELIGIBLE FOR HIGHER STUDIES

JOHNS V. JOHN
SECRETARY
Board of Public Examinations, Kerala

Date of Publication of Result : 24/04/2013

Name & Signature of the Head of School

Fig. 1.3 : SSLC score sheet

Information is always generated by performing some operations on data. In other words, data is like raw material to generate information. Now let us try to distinguish between these two terms. Table 1.1 summarises the comparison between data and information.

Data	Information
• Raw facts and figures	• Processed data
• Similar to raw material	• Similar to the finished product
• Cannot be directly used	• Adds to knowledge and helps in taking decisions
• Does not give precise and clear sense	• Clear and meaningful

Table 1.1 : Comparison between data and information

As we know, information always adds to knowledge. One can apply this knowledge to solve problems or in decision making. Generally, the ability to draw useful inferences from the acquired knowledge is known as intelligence. It depends on how we process knowledge and apply it in various situations. Recent advancement in Computer Science and technology have attempted to make computers do things, which at the moment people do better, incorporating knowledge and intelligence. This is referred to as artificial intelligence.

1.2 Data processing

In the case of preparation of the score sheet mentioned in the previous section, the scores given to each subject as part of Continuous Evaluation (CE) and Terminal Evaluation (TE) are added together, and grades are determined based on some predefined criteria. The activities or operations to generate information can collectively be termed as process. **Data processing** refers to the operations or activities performed on data to generate information. So we can say that information is the result of data processing.

As shown in Figure 1.4, data is supplied for processing and information is obtained after processing. In other words, data is the input to the process and information is the output from the process.
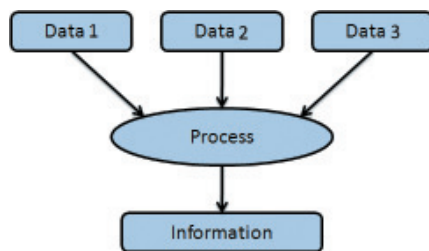

Fig. 1.4 : Data processing

Let us consider the case of the Single Window System (*Ekajaalakam*) - the admission procedure for higher secondary courses in Kerala. We can briefly list out its activities as follows:

1. The authority collects the data from applicants through application forms, in which score sheet of Class X examination will be referenced to furnish the required details. Note that in this context the facts and figures in the score sheet become the data.

2. The collected data is then fed to the computer.

3. The input data is stored and will later be retrieved for processing.

4. The data within the computer is used for performing operations such as calculations, comparisons, categorisation, sorting, filtering, etc.

5. The allotment slips for candidates and allotment lists for schools are generated. The slips and lists are printed and may be stored for later reference. It may be used as data to generate information in some other situation.

6. The slips are distributed to the applicants and lists are forwarded to schools.

Thus it is clear that data processing proceeds through six stages, as listed below:

(a)   Capturing data

(b)   Input of data

(c)   Storage of data

(d)   Processing / manipulating data

(e)   Output of information

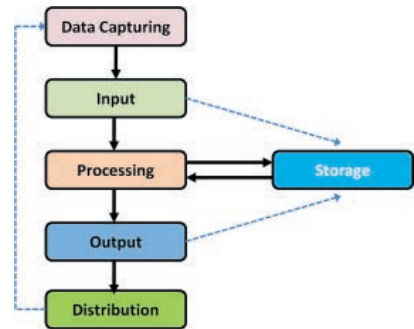(f)   Distribution of information

Fig. 1.5 : Stages of data processing

The thick arrow marks in Figure 1.5 indicate the flow of the activities in data processing and the dotted lines specify the flow of activities that are optional. Let us take a close look at these stages.

a. Capturing data

When we apply for admission to the higher secondary course, we usually provide details through a prescribed application form. The authority is actually

collecting the required data for the admission process through the proforma. This is the first stage in data processing. The proforma, also known as the source document, is so designed that all relevant data to be recorded in proper order and format. Thus, preparation of hard copy of source document and data collection are the activities that take place in this stage. Today, prescribed application forms are not used for collecting data. Instead, data are directly entered through on-line facility.

b. Input

In the case of seeking admission, we submit the filled up application form to the school.

There the data is extracted and fed into the computer. Sometimes, we may enter these details directly into the computer. Feeding data to the computer for processing is known as input. The input data is usually stored in computers before it is processed.
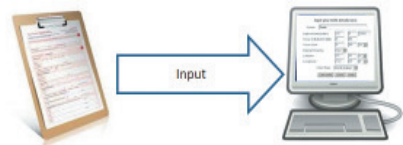
c. Storage

In many cases, the amount of data given to the computers will be large. Besides, the data entry may not be completed in a single session or a day. In the case of admissions, the data of lakhs of applicants is input to the computer. It usually takes a few weeks to complete the data entry. So the data input at different times should be stored then and there. The processing will start only after the entire data is stored. The information obtained as a result of processing is also stored in the computer. This stored data and information can be used in future for various purposes.

d. Process

The data stored in computers is retrieved for processing. Various operations like calculation, classification, comparison, sorting, filtering, summarising etc. are carried out as part of processing. In the case of admission to the higher secondary course, Weighted Grade Point Average (WGPA) of each applicant is calculated. Then the applicants are listed under various categories based on the descending order of WGPA. Here, school of choice, course, and performance in various co-curricular activities are considered. Finally, allotment lists for schools and allotment slips for applicants are prepared.

e. Output

The information obtained after processing will be available in this stage. Output stage should provide the information in such a form that the beneficiary should be able to take decision or solve the problem. In the case of admission to the higher secondary course, allotment slip for the applicant and allotment list for the school are generated in the desired format as outputs.

f. Distribution of information

The information obtained in the output stage is distributed to the beneficiaries. They take decisions or solve problems according to the information. For example in higher secondary admission, the allotment slips are distributed to applicants for joining the school allotted and allotment lists are issued to the schools for admitting the eligible applicants. The allotment slips may be used to prepare admission register or roll list of classes. The allotment lists may be used to prepare nominal roll for registering the students for public examination.

> **Let us do**
> - *Identify and write the data processing activities in (i) opening an account in a bank and (ii) applying for scholarships*
> - *Identify data processing cases in any other real life situations and write the activities performed in each stage.*

Check yourself

1. Raw facts and figures are known as _____ .

2. Processed data is known as _____.

3. Which of the following helps us to take decisions?

    (a) data   (b) information   (c) knowledge   (d) intelligence

4. Manipulation of data to get information is known as _____.

5. Arrange the following in proper order:

    Process, Output, Storage, Distribution, Data Capture, Input

6. Pick the odd one out and give reason:

    (a) Calculation   (b) Storage   (c) Comparison   (d) Categorization

7. Why do we store information?

8. Information may act as data. State True or False.

9. Which is the final stage in data processing?

10. What is a source document?

1.3 Functional units of a computer

Even though computers differ in size, shape, performance and cost over the years, the basic organisation of a computer is the same. It is based on a model proposed by John Von Neumann, a mathematician and a computer scientist. It consists of some functional units namely Input Unit, Central Processing Unit (CPU), Storage Unit and Output Unit. Each of these units is assigned to perform a particular task. Let us discuss the functions of these units. Figure 1.7 shows the basic functional units of a computer.

Fig. 1.6 : John Von Neumann (1903 - 1957)

1. Input unit

The collected data and the instructions for their processing are entered into the computer through the input unit. They are stored in the memory (storage unit). The data may be in different forms like number, text, image, audio, video, etc. A variety of devices are
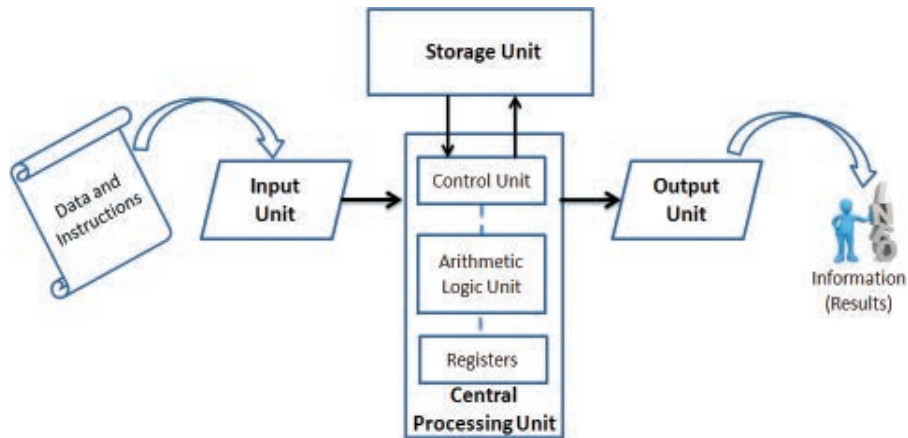
Fig. 1.7 : Functional units of a computer

available to input the data depending on its nature. Keyboard, mouse, scanner, mic, digital camera, etc. are some commonly used input devices. In short, the functions performed by input unit are as follows:

1. Accepts instructions and data from the outside world.

2. Converts these instructions and data to a form acceptable to the computer.

3. Supplies the converted instructions and data to the computer for processing.

2. Central Processing Unit (CPU)

The CPU is the brain of the computer. In a human body, all major decisions are taken by the brain and other parts of the body function as directed by the brain. Similarly, in a computer system, all major computations and comparisons are made inside the CPU. It is also responsible for activating and controlling the operations of other units of the computer. The functions of CPU are performed by three components - Arithmetic Logic Unit (ALU), Control Unit (CU) and registers.

a. Arithmetic Logic Unit (ALU)

The actual operations specified in the instructions are carried out in the Arithmetic Logic Unit (ALU). It performs calculations and logical operations such as comparisons and decision making. The data and instructions stored in the storage unit are transferred to the ALU and the processing takes place in it. Intermediate results produced by the ALU are temporarily transferred back to the storage and are retrieved later when needed for further processing. Thus there is a data flow between the storage and the ALU many times before the entire processing is completed.

b. Control Unit (CU)

Each of the functional units has its own function, but none of these will perform the function until it is asked to. This task is assigned to the control unit. It invokes the other units to take charge of the operation they are associated with. It is the central nervous system that manages and co-ordinates all other units of the computer. It obtains instructions from the program stored in the memory, interprets the operation and issues signals to the unit concerned in the system to execute them.

c. Registers

These are temporary storage elements that facilitate the functions of CPU. There are variety of registers; each designated to store unique items like data, instruction, memory address, results, etc.

3. Storage unit

The data and instructions entered in the computer through input unit are stored inside the computer before actual processing starts. Similarly, the information or results produced after processing are also stored inside the computer, before transferring to the output unit. Moreover, the intermediate results, if any, must also be stored for further processing. The storage unit of a computer serves all these purposes. In short, the specific functions of storage unit are to hold or store:

1.   data and instructions required for processing.

2.   intermediate results for ongoing processing.

3.   final results of processing, before releasing to the output unit.

The storage unit comprises of two types as detailed below:

Primary storage: It is also known as main memory. It is again divided into two - Random Access Memory (RAM) and Read Only Memory (ROM). RAM holds instructions, data and intermediate results of processing. It also holds the recently produced results of the job done by the computer. ROM contains instructions for the start up procedure of the computer. The Central Processing Unit can directly access the main memory at a very high speed. But it is costly and has limited storage capacity.

Secondary storage: It is also known as auxiliary storage and it takes care of the limitations of primary storage. It has a huge storage capacity and the storage is permanent. Usually we store data, programs and information in the secondary storage, but we have to give instruction explicitly for this. Hard disk, CDs, DVDs, memory sticks, etc. are some examples.

4. Output unit

The information obtained after data processing is supplied to the outside world through the output unit in a human-readable form. Monitor and printer are the commonly used output devices. The functions performed by output unit can be concluded as follows:

1.   Receives the results produced by the CPU in coded form.

2.   Converts these coded results to human-readable form.

3.   Supplies the results to the outside world.

Let us do

Fill up the following table by comparing human beings and the computer in the context of data processing. In the case of operations, the organs or components may be specified and for characteristics, performance may be indicated. You can also add more features.

Features	Human being	Computer
Operations		
Input	*Eyes, Ears*	*Keyboard, Mouse*
Output		
Calculation & Comparison		
Temporary Storage		
Permanent Storage		
Controlling		
Characteristics		
Speed		
Accuracy		
Reliability		
•		
•		

1.4 Computer - as a data processor

We have seen the activities involved in data processing and identified the different stages in data processing. Imagine the situation where humans are involved in these stages for performing the operations. It is sure that we will not get the information always in time and without any error all the time. We always need accurate, comprehensive, reliable and timely information in proper format and media so that it can be applied to the

context concerned to formulate knowledge. Only then problems can be solved and/or decisions be made appropriately. From the discussions we had so far, computers can be considered as the best data processing machine. In short, **computer** may be defined as an electronic machine designed to accept the data and instructions, performs arithmetic and logical operations on the data according to a set of instructions and output the results or information.

1.4.1 Characteristics of computers

We have already recognized some of the characteristics of computers by performing the learning activity of filling the comparison table given in the Let us do box. As we know, computers can execute millions of instructions in a second. The results produced after processing the data are very accurate, but computers do not have adequate knowledge or intelligence to interpret the results. They only carry out instructions like an obedient servant. The computer gives correct results only if the data and instructions given are correct. The term Garbage In Garbage Out (GIGO) is used to mean this feature. That is, if a wrong input is given to the computer, it will give a wrong output. Look at Table 1.2 and identify the advantages and limitations of computer.

Computers	
Advantages	**Limitations**
Speed: A computer can perform millions of operations in a second or in fraction of second. It can do in a minute, as much work as a man do taking months and years.	*Lack of IQ*: Many people think that computer has super human capabilities. However this is not true. A computer does not have natural intelligence as humans have.
Accuracy: A computer can perform arithmetic operations with a very high degree of accuracy. By accuracy, we mean fewer errors in the output and precision with which computations are performed.	
Diligence: Since computer is a machine, it can operate for long hours untiringly. Unlike human beings, it will not show any emotion or disobey you. Hence computers are best suited for routine jobs.	*Lack of decision making power*: Computer cannot decide on its own and it does not possess intuitive capabilities like human beings.
Versatility: Computer can be used to perform many different kinds of processing tasks. It is a general purpose data processing machine.	
Huge memory: Computer has enormous memory capacity. Huge volume of data can be stored in its memory for processing. The storage capacity can also be increased as per requirement.	

Table 1.2 : Advantages and limitations of computers

Check yourself

1. Who proposed the model of modern computers?
2. Name the components of CPU.
3. Which of the functional units of the computer is not directly involved in data processing?
4. What is meant by 'execution of an instruction'?
5. Which part of a computer can be compared to the human brain?

1.5 Number system

A number is a mathematical object used to count, label and measure. A number system is a systematic way to represent numbers. The number system we use in our day to day life is the decimal number system that uses ten symbols or digits. The number 289 is pronounced as two hundred and eighty nine and it consists of the symbols 2, 8 and 9. Similarly there are other number systems. Each has its own symbols and method for constructing a number. A number system has a unique base, which depends upon the number of symbols. The number of symbols used in a number system is called **base** or **radix** of a number system.

Let us discuss some of the number systems.

1.5.1 Decimal number system

The decimal number system involves ten symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 to form a number. Since there are 10 symbols in this number system, its base is 10. Therefore, the decimal number system is also known as base-10 number system.

Consider two decimal numbers 743 and 347

$743 \rightarrow$ seven hundred + four tens+ three ones ($7\times10^2 + 4\times10^1 + 3\times10^0$)

$347 \rightarrow$ three hundreds + four tens + seven ones ($3\times10^2 + 4 \times10^1 + 7\times10^0$)

Here, place value (weight) of 7 in first number 743 is $10^2 =100$. But weight of 7 in second number 347 is $10^0 =1$. The weight of a digit depends on its relative position. Such a number system is known as *positional number system.* All positional number systems have a base and the place value of a digit is some power of this base.

Place value of each decimal digit is power of 10 (10^0, 10^1, 10^2, ...). Consider a decimal number 5876.

This number can be written in expanded form as

Weight	10^3	10^2	10^1	10^0
Decimal Number	5	8	7	6

$= \quad 5 \times 10^3 + 8 \times 10^2 + 7 \times 10^1 + 6 \times 10^0$

$= \quad 5 \times 1000 + 8 \times 100 + 7 \times 10 + 6 \times 1$

$= \quad 5000 + 800 + 70 + 6$

$= \quad 5876$

In the above example, the digit 5 has the maximum place value, $10^3 = 1000$ and 6 has the minimum place value, $10^0 = 1$. The digit with most weight is called Most Significant Digit (**MSD**) and the digit with least weight is called Least Significant Digit (**LSD**). So in the above number MSD is 5 and LSD is 6.

Left most digit of a number is MSD and right most digit of a number is LSD

For fractional numbers weights are negative powers of 10 (10^{-1}, 10^{-2}, 10^3,.....) for the digits to the right of decimal point. Consider another example 249.367

Weight	10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}
Decimal Number	2	4	9	3	6	7
	MSD		(.)			LSD

$= \quad 2 \times 10^2 + 4 \times 10^1 + 9 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2} + 7 \times 10^{-3}$

$= \quad 2 \times 100 + 4 \times 10 + 9 \times 1 + 3 \times 0.1 + 6 \times 0.01 + 7 \times 0.001$

$= \quad 200 + 40 + 9 + 0.3 + 0.06 + 0.007$

$= \quad 249.367$

So far we have discussed a number system which uses 10 symbols. Now let us see the construction of other number systems with different bases.

1.5.2 Binary number system

A number system which uses only two symbols 0 and 1 to form a number is called binary number system. Bi means two. Base of this number system is 2. So it is also called base-2 number system. We use the subscript 2 to indicate that the number is in binary.

e.g. $(1101)_2$, $(101010)_2$, $(1101.11)_2$

Each digit of a binary number is called bit. A **bit** stands for **binary digit**. The binary number system is also a positional number system where place value of each binary

digit is power of 2. Consider an example $(1101)_2$. This binary number can be written in expanded form as shown below.

Weight	2^3	2^2	2^1	2^0
Binary Number	1	1	0	1

$$\qquad\qquad\text{MSB}\qquad\qquad\qquad\text{LSB}$$

$$= 1\times2^3 + 1\times 2^2 + 0\times2^1 + 1\times2^0$$
$$= 1\times8 + 1\times4 + 0\times2 + 1\times1$$
$$= 8 + 4 + 0 + 1$$
$$= 13$$

The right most bit in a binary number is called Least significant Bit (**LSB**). The leftmost bit in a binary number is called Most significant Bit (**MSB**).

The binary number 1101 is equivalent to the decimal number 13. The number 1101 also exists in the decimal number system. But it is interpreted as one thousand one hundred and one. To avoid this confusion, base must be specified in all number systems other than decimal number system. The general format is

$$\textbf{(Number)}_{\textbf{base}}$$

This notation helps to differentiate numbers of different bases. So a binary number must be represented with base 2 as $(1101)_2$ and it is read as "one one zero one to the base two".

If no base is given in a number, it will be considered as decimal. In other words, specifying the base is not compulsory in decimal number.

For fractional numbers, weights are negative powers of 2 ($2^{-1}, 2^{-2}, 2^{-3}, ...$) for the digits to the right of the binary point. Consider an example $(111.011)_2$

Weight	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Binary Number	1	1	1	0	1	1

$$\qquad\qquad\text{MSB}\qquad\qquad\qquad(.)\qquad\qquad\text{LSB}$$

$$= 1\times2^2 + 1\times2^1 + 1\times2^0 + 0\times2^{-1} + 1\times2^{-2} + 1\times2^{-3}$$

$$= 1\times4 + 1\times2 + 1\times1 + 0\times\frac{1}{2} + 1\times\quad + 1\times$$

$$= 4 + 2 + 1 + 0 + 0.25 + 0.125$$

$$= 7.375$$

Importance of binary numbers in computers

We have seen that binary number system is based on two digits 1 and 0. The electric state ON can be represented by 1 and the OFF state by 0 as shown in Figure 1.8 Because of this, computer uses binary number system as the basic number system for data representation.

$$1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0$$
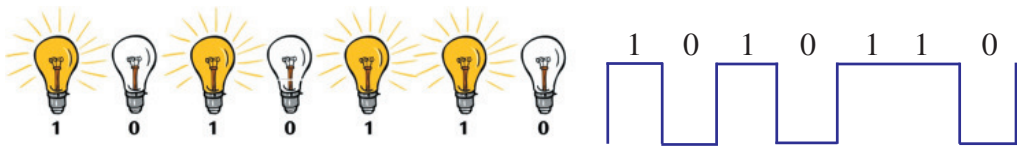
Fig. 1.8 : Digital representation of ON and OFF states

1.5.3 Octal number system

A number system which uses eight symbols 0, 1, 2, 3, 4, 5, 6 and 7 to form a number is called octal number system. Octa means eight, hence this number system is called octal. Base of this number system is 8 and hence it is also called base-8 number system. Consider an example $(236)_8$. Weight of each digit is power of 8 ($8^0, 8^1, 8^2, 8^3, \ldots$). The number $(236)_8$ can be written in expanded form as

Weight	8^2	8^1	8^0
Octal Number	2	3	6

$$= \quad 2 \times 8^2 + 3 \times 8^1 + 6 \times 8^0$$
$$= \quad 2 \times 64 + 3 \times 8 + 6 \times 1$$
$$= \quad 128 + 24 + 6$$
$$= \quad 158$$

For fractional numbers weights are negative powers of 8, i.e. ($8^{-1}, 8^{-2}, 8^{-3}, \ldots$) for the digits to the right of the octal point. Consider an example $(172.4)_8$

Weight	8^2	8^1	8^0	8^{-1}
Octal Number	1	7	2	4

$$= \quad 1 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 + 4 \times 8^{-1}$$
$$= \quad 64 + 56 + 2 + 4 \times$$
$$= \quad 122 + 0.5$$
$$= \quad 122.5$$

$\frac{1}{8}$

1.5.4 Hexadecimal number system

A number system which uses 16 symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F to form a number is called hexadecimal number system. Base of this number system is 16 as there are sixteen symbols in this number system. Hence this number system is also called base-16 number system.

In this system, the symbols A, B, C, D, E and F are used to represent the decimal numbers 10, 11, 12, 13, 14 and 15 respectively. The hexadecimal digit and their equivalent decimal numbers are shown below.

Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Consider a hexadecimal number $(12AF)_{16}$. Weights of each digit is power of 16 (16^0, 16^1, 16^2, ...).

This number can be written in expanded form as shown below.

Weight	16^3	16^2	16^1	16^0
Hexadecimal Number	1	2	A	F

$$= \quad 1 \times 16^3 + 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0$$
$$= \quad 1 \times 4096 + 2 \times 256 + 10 \times 16 + 15 \times 1$$
$$= \quad 4096 + 512 + 160 + 15$$
$$= \quad 4783$$

For fractional numbers, weights are some negative power of 16 (16^{-1}, 16^{-2}, 16^{-3}, ...) for the digits to the right of the hexadecimal point. Consider an example $(2D.4)_{16}$

Weight	16^1	16^0	16^{-1}
Hexadecimal	2	D	4

$$= \quad 2 \times 16^1 + 13 \times 16^0 + 4 \times$$
$$= \quad 32 + 13 + 0.25$$
$$= \quad 45.25$$

Table 1.3 shows the base and symbols used in different number systems:

Number System	Base	Symbols used
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Table 1.3 : Number systems with base and symbols

Importance of octal and hexadecimal number systems

As we have discussed, digital hardware uses the binary number system for its operations and data. Representing numbers and operations in binary form requires too many bits and needs lot of effort. With octal, the bits are grouped in threes (because $2^3 = 8$), and with hexadecimal, the binary bits are grouped in fours (because $2^4 = 16$) and these groups are replaced with the respective octal or hexadecimal symbol. This conversion processes of binary numbers to octal and hexadecimal number systems and vice versa are very easy. This short-hand notation is widely used in the design and operations of electronic circuits.

Check yourself

1. Number of symbols used in a number system is called _____.
2. Pick invalid numbers from the following
   i) $(10101)_8$ ii) $(123)_4$ iii) $(768)_8$ iv) $(ABC)_{16}$
3. Define the term 'bit'.
4. Find MSD in the decimal number 7854.25.
5. The base of hexadecimal number system is _____.

1.6 Number conversions

After having learnt the various number systems, let us now discuss how to convert the numbers of one base to the equivalent numbers in other bases. There are different types of number conversions like decimal to binary, binary to decimal, decimal to octal etc. This section discusses how to convert one number system to another.

1.6.1 Decimal to binary conversion

The method of converting decimal number to binary number is by repeated division. In this method the decimal number is successively divided by 2 and the remainders are recorded. The binary equivalent is obtained by grouping all the remainders, with the last remainder being the Most Significant Bit (MSB) and first remainder being the Least Significant Bit (LSB). In all these cases the remainders will be either 0 or 1 (binary digit).

Example:

Find binary equivalent of decimal number 25.

2	25	Remainders	
2	12	1	LSB
2	6	0	
2	3	0	
2	1	1	
	0	1	MSB

$$(25)_{10} = (11001)_2$$

Find binary equivalent of $(80)_{10}$.

2	80	Remainders	
2	40	0	LSB
2	20	0	
2	10	0	
2	5	0	
2	2	1	
2	1	0	
	0	1	MSB

$$(80)_{10} = (1010000)_2$$

Hint: Binary equivalent of an odd decimal number ends with 1 and binary of even decimal number ends with zero.

Converting decimal fraction to binary

To convert a fractional decimal number to binary, we use the method of repeated multiplication by 2. At first the decimal fraction is multiplied by 2. The integer part of the answer will be the MSB of binary fraction. Again the fractional part of the answer is multiplied by 2 to obtain the next significant bit of binary fraction. The procedure is continued till the fractional part of product is zero or a desired precision is obtained.

Example: Convert 0.75 to binary.

	$0.75 \times 2 = 1.50$
1	$.50 \times 2 = 1.00$
1	$.00$

$$(0.75)_{10} = (0.11)_2$$

Example: Convert 0.625 to binary.

	$0.625 \times 2 = 1.25$
1	$.25 \times 2 = 0.50$
0	$.50 \times 2 = 1.00$
1	$.00$

$$(0.625)_{10} = (0.101)_2$$

Example: Convert 15.25 to binary.

Convert 15 to binary.

2	15	Remainders
2	7	1
2	3	1
2	1	1
	0	1

Convert 0.25 to binary.

	$0.25 \times 2 = 0.50$
0	$.50 \times 2 = 1.00$
1	$.00$

$$(15.25)_{10} = (1111.01)_2$$

1.6.2 Decimal to octal conversion

The method of converting decimal number to octal number is also by repeated division. In this method the number is successively divided by 8 and the remainders are recorded. The octal equivalent is obtained by grouping all the remainders, with the last remainder being the MSD and first remainder being the LSD. Remainders will be either 0, 1, 2, 3, 4, 5, 6 or 7.

Example: Find octal equivalent of decimal number 125.

8	125	Remainders	
8	15	5	LSD
8	1	7	
	0	1	MSD

$$(125)_{10} = (175)_8$$

Example: Find octal equivalent of $(400)_{10}$.

```
8 |  400    Remainders
8 |   50        0        ↑
8 |    6        2        |
         0        6        |
```

$(400)_{10} = (620)_8$

1.6.3 Decimal to hexadecimal conversion

The method of converting decimal number to hexadecimal number is also by repeated division . In this method, the number is successively divided by 16 and the remainders are recorded. The hexadecimal equivalent is obtained by grouping all the remainders, with the last remainder being the Most Significant Digit (MSD) and first remainder being the Least Significant Digit(LSD). Remainders will be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E or F.

Example: Find hexadecimal equivalent of decimal number 155.

```
16 |  155      Remainders
16 |    9       11 (B)  ↑    ⟶  LSD
          0        9           ⟶  MSD
```

$(155)_{10} = (9B)_{16}$

Example: Find hexadecimal equivalent of 380.

```
16 |  380    Remainders
16 |   23       12 (C)  ↑
16 |    1        7
          0        1
```

$(380)_{10} = (17C)_{16}$

1.6.4 Binary to decimal conversion

A binary number can be converted into its decimal equivalent by summing up the product of each bit and its weight. Weights are some power of 2. $(2^0, 2^1, 2^2, 2^3, ...)$

Example: Convert $(11011)_2$ to decimal.

$(11011)_2$ = $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

Weight	2^4	2^3	2^2	2^1	2^0
Bit	1	1	0	1	1

= $16 + 8 + 2 + 1$

= 27

$(11011)_2 = (27)_{10}$

Example: Convert $(1100010)_2$ to decimal.

Weight	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Bit	1	1	0	0	0	1	0

$(1100010)_2 = $ $1\times2^6 + 1\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= $ $64 + 32 + 2$

$= $ 98 $(1100010)_2 = (98)_{10}$

Table 1.4 may help us to find powers of 2.

2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1024	512	256	128	64	32	16	8	4	2	1

Table 1.4 : Powers of 2

Converting binary fraction to decimal

A binary fraction number can be converted into its decimal equivalent by summing up the product of each bit and its weight. Weights of binary fractions are negative powers of 2 $(2^{-1}, 2^{-2}, 2^{-3}, ...)$ for the digits after the binary point.

Example: Convert $(0.101)_2$ to decimal.

Weight	2^{-1}	2^{-2}	2^{-3}
Bit	1	0	1

$(0.101)_2$ $= 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$

$= 0.5 + 0 + 0.125$

$= 0.625$ $(0.101)_2 = (0.625)_{10}$

Example: Convert $(1010.11)_2$ to decimal.

Weight	2^3	2^2	2^1	2^0
Bit	1	0	1	0

$(1010)_2$ $= 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0$

$= 8 + 0 + 2 + 0$

$= 10$ $(1010)_2 = (10)_{10}$

Weight	2^{-1}	2^{-2}
Bit	1	1

$(0.11)_2$ $= 1\times2^{-1} + 1\times2^{-2}$

$= 0.5 + 0.25$

$= 0.75$ $(0.11)_2 = (0.75)_{10}$

$(1010.11)_2 = (10.75)_{10}$

Table 1.5 shows negative powers of 2.

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
0.5	0.25	0.125	0.0625	0.03125

Table 1.5 : Negative powers of 2

1.6.5 Octal to decimal conversion

An octal number can be converted into its decimal equivalent by summing up the product of each octal digit and its weight. Weights are some powers of 8 (8^0, 8^1, 8^2, 8^3, ...).

Example: Convert $(157)_8$ to decimal.

Weight	8^2	8^1	8^0
Octal digit	1	5	7

$$(157)_8 \quad = 1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0$$
$$= 64 + 40 + 7$$
$$= 111$$

$$(157)_8 = (111)_{10}$$

Example: Convert $(1005)_8$ to decimal.

Weight	8^3	8^2	8^1	8^0
Octal digit	1	0	0	5

$$(1005)_8 \quad = 1 \times 8^3 + 0 \times 8^2 + 0 \times 8^1 + 5 \times 8^0$$
$$= 512 + 5$$
$$= 517$$

$$(1005)_8 = (517)_{10}$$

1.6.6 Hexadecimal to decimal conversion

An hexadecimal number can be converted into its decimal equivalent by summing up the product of each hexadecimal digit and its weight. Weights are powers of 16 (16^0, 16^1, 16^2, ...).

Example: Convert $(AB)_{16}$ to decimal.

Weight		16^1	16^0
Hexadecimal digit		A	B

$$(AB)_{16} \quad = 10 \times 16^1 + 11 \times 16^0$$
$$= 160 + 11$$
$$= 171$$

A = 10   B = 11

$$(AB)_{16} = (171)_{10}$$

Example: Convert $(2D5)_{16}$ to decimal.

Weight	16^2	16^1	16^0
Hexadecimal digit	2	D	5

$$(2D5)_{16} \quad = 2 \times 16^2 + 13 \times 16^1 + 5 \times 16^0$$
$$= 512 + 208 + 5$$
$$= 725$$

D = 13

$$(2D5)_{16} = (725)_{10}$$

1.6.7 Octal to binary conversion

An octal number can be converted into binary by converting each octal digit to its 3 bit binary equivalent. Eight possible octal digits and their binary equivalents are listed in Table 1.6.

Octal digit	0	1	2	3	4	5	6	7
Binary equivalent	000	001	010	011	100	101	110	111

Table 1.6 : Binary equivalent of octal digit

Example: Convert $(437)_8$ to binary.

3-bit binary equivalent of each octal digits are

4	3	7
↓	↓	↓
100	011	111

$$(437)_8 = (100011111)_2$$

Example: Convert $(7201)_8$ to binary.

3-bit binary equivalent of each octal digits are

7	2	0	1
↓	↓	↓	↓
111	010	000	001

$$(7201)_8 = (111010000001)_2$$

1.6.8 Hexadecimal to binary conversion

A hexadecimal number can be converted into binary by converting each hexadecimal digit to its 4 bit binary equivalent. Sixteen possible hexadecimal digits and their binary equivalents are listed in Table 1.7.

Example: Convert $(AB)_{16}$ to binary.

4-bit binary equivalent of each hexadecimal digits are

A	B
↓	↓
1010	1011

$$(AB)_{16} = (10101011)_2$$

Example: Convert $(2F15)_{16}$ to binary.

4-bit binary equivalent of each hexadecimal digits are

2	F	1	5
↓	↓	↓	↓
0010	1111	0001	0101

$$(2F15)_{16} = (10111100010101)_2$$

Hexa decimal	Binary equivalent
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Table 1.7 :
Binary equivalent of hexadecimal digits

1.6.9 Binary to octal conversion

A binary number can be converted into its octal equivalent by grouping binary digits to group of 3 bits and then each group is converted to its octal equivalent. Start grouping from right to left.

Example: Convert $(101100111)_2$ to octal.

We can group the given binary number 101100111 from right as shown below.

$$101 \qquad 100 \qquad 111$$
$$\downarrow \qquad \downarrow \qquad \downarrow$$
$$5 \qquad 4 \qquad 7$$

$$\boxed{(101100111)_2 \ = \ (547)_8}$$

Example: Convert $(10011000011)_2$ to octal.

We can group the given binary number 10011000011 from right as shown below.

After grouping, if the left most group does not have 3 bits, then add leading zeros to form 3 bit binary.

$$010 \qquad 011 \qquad 000 \qquad 011$$
$$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$$
$$2 \qquad 3 \qquad 0 \qquad 3$$

$$\boxed{(10011000011)_2 = (2303)_8}$$

1.6.10 Binary to hexadecimal conversion

A binary number can be converted into its hexadecimal equivalent by grouping binary digits to group of 4 bits and then each group is converted to its hexadecimal equivalent. Start grouping from right to left.

Example: Convert $(101100111010)_2$ to hexadecimal.

We can group the given binary number 101100111010 from right as shown below.

$$1011 \qquad 0011 \qquad 1010$$
$$\downarrow \qquad \downarrow \qquad \downarrow$$
$$B \qquad 3 \qquad A$$

$$\boxed{(101100111010)_2 = (B3A)_{16}}$$

Example: Convert $(110111100001100)_2$ to hexadecimal.

We can group the given binary number 110111100001100 from right as shown below.

After grouping, if the left most group does not have 4 bits, then add leading zeros to form 4 bit binary.

$$0110 \qquad 1111 \qquad 0000 \qquad 1100$$
$$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$$
$$6 \qquad F \qquad 0 \qquad C$$

$$\boxed{(110111100001100)_2 = (6F0C)_{16}}$$

1.6.11 Octal to hexadecimal conversion

Conversion of an octal number to hexadecimal number is a two step process. Octal number is first converted into binary. This binary equivalent is then converted into hexadecimal.

Example: Convert $(457)_8$ to hexadecimal equivalent.

First convert $(457)_8$ into binary.

$$(457)_8 = \quad 4 \qquad\qquad 5 \qquad\qquad 7$$
$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$
$$100 \qquad\qquad 101 \qquad\qquad 111$$
$$= (100101111)_2$$

Then convert $(100101111)_2$ into hexadecimal.

$$(100101111)_2 = \quad 0001 \qquad\qquad 0010 \qquad\qquad 1111$$
$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$
$$= \quad 1 \qquad\qquad 2 \qquad\qquad F$$
$$= (12F)_{16} \qquad\qquad \boxed{(457)_8 = (12F)_{16}}$$

1.6.12 Hexadecimal to octal conversion

Conversion of an hexadecimal to octal number is also a two step process. Hexadecimal number is first converted into binary. This binary equivalent is then converted into octal.

Example: Convert $(A2D)_{16}$ into octal equivalent.

First convert $(A2D)_{16}$ into binary.

$$(A2D)_{16} = \qquad A \qquad\qquad 2 \qquad\qquad D$$
$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$
$$1010 \qquad\qquad 0010 \qquad\qquad 1101$$
$$= (101000101101)_2$$

Then convert $(101000101101)_2$ into octal.

$$(1010\ 00101101)_2 = \quad 101 \qquad 000 \qquad 101 \qquad 101$$
$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$
$$5 \qquad\qquad 0 \qquad\qquad 5 \qquad\qquad 5$$
$$= (5055)_8$$
$$\boxed{(A2D)_{16} = (5055)_8}$$

Table 1.8 summarises the procedures for all types of number conversions that are discussed in this chapter.

Conversion	Procedure
Decimal to Binary	Repeated division by 2 and grouping the remainders
Decimal to Octal	Repeated division by 8 and grouping the remainders
Decimal to Hexadecimal	Repeated division by 16 and grouping the remainders
Binary to Decimal	Multiply binary digit by place value(power of 2) and find their sum
Octal to Decimal	Multiply octal digit by place value (power of 8) and find their sum
Hexadecimal to Decimal	Multiply hexadecimal digit by place value (power of 16) and find their sum
Octal to Binary	Converting each octal digit to its 3 bit binary equivalent
Hexadecimal to Binary	Converting each hexadecimal digit to its 4 bit binary equivalent
Binary to Octal	Grouping binary digits to group of 3 bits from right to left
Binary to Hexadecimal	Grouping binary digits to group of 4 bits from right to left
Octal to Hexadecimal	Convert octal to binary and then binary to hexadecimal
Hexadecimal to Octal	Convert hexadecimal to binary and then binary to octal

Table 1.8: Procedure for number conversions

1.7 Binary addition

As in the case of decimal number system, arithmetic operations are performed in binary number system. When we give instruction to add two decimal numbers, the computer actually adds their binary equivalents. Let us see how binary addition is carried out. The rules for adding two bits are given in Table 1.9

Note that a carry bit 1 is created only when two ones are added. If three ones are added (i.e. 1+1+1), then the sum bit is 1 with a carry bit 1.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 1.9: Binary addition rules

Example: Find sum of binary numbers 1011 and 1001.

$$
\begin{array}{r}
1\ 0\ 1\ 1\ + \\
\underline{1\ 0\ 0\ 1} \\
1\ 0\ 1\ 0\ 0
\end{array}
$$

Example: Find sum of binary numbers 110111 and 10011.

$$
\begin{array}{r}
1\ 1\ 0\ 1\ 1\ 1\ + \\
\underline{1\ 0\ 0\ 1\ 1\ 0} \\
1\ 0\ 1\ 1\ 1\ 0\ 1
\end{array}
$$

1.8 Data representation

Computers process different types of data such as numbers, characters, images, audios and video. We know that computer is an electronic device functioning on the basis of two states - ON and OFF. The two-state operation is called binary operation. Hence the data given to computer should also be in binary form. Data representation is the method used internally to represent data in a computer. Computer uses a fixed number of bits to represent a piece of data which could be a number, a character, image, audio, video etc. Let us see how various types of data can be represented in computer memory.

1.8.1 Representation of numbers

Numbers can be classified into integer numbers and floating point numbers. Integers are whole numbers or fixed point numbers without any fractional part. A floating point number or a real number is a number with fractional part. These two numbers are treated differently in computer memory. Let us see how integers are represented.

a. Representation of integers

There are three methods for representing an integer number in computer memory.

i)   Sign and magnitude representation
ii)   1's complement representation
iii)   2's complement representation

The following data representation methods are based on 8 bit word length.

A **word** is basically a fixed-sized group of bits that are handled as a unit by a processor. Number of bits in a word is called **word length**. The word length is the choice of computer designer and some popular word lengths are 8, 16, 32 and 64.
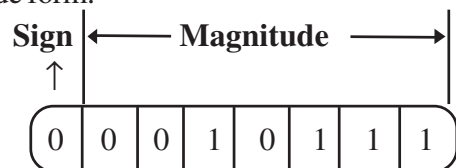
i) Sign and magnitude representation

In this method, first bit from left (MSB) is used for representing sign of integer and remaining 7-bits are used for representing magnitude of integer. For negative integers sign bit is 1 and for positive integers sign bit is 0. Magnitude is represented as 7 bit binary equivalent of the integer.

Example: Represent + 23 in sign and magnitude form.

Number is positive, so first bit (MSB) is 0.

7 bit binary equivalent of $23 = (0010111)_2$

So + 23 can be represented as $(00010111)_2$

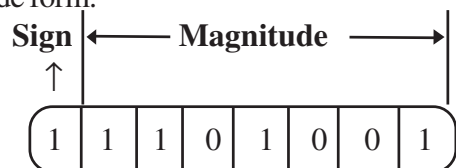| Sign	Magnitude						
0	0	0	1	0	1	1	1

Example: Represent -105 in sign and magnitude form.

Number is negative, so first bit(MSB) is 1.

7 bit binary equivalent of $105 = (1101001)_2$

So -105 can be represented as $(11101001)_2$

Sign	Magnitude						
1	1	1	0	1	0	0	1

Note: In this method an 8 bit word can represent $2^8-1 = 255$ numbers (i.e. -127 to +127). Similarly, a 16 bit word can represent $2^{16}-1 = 65535$ numbers (i.e. -32767 to +32767). So, an n-bit word can represent 2^n-1 numbers i.e., $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$. The integer 0 can be represented in two ways: $+0 = 00000000$ and $-0 = 10000000$.

ii) 1's complement representation

In this method, first find binary equivalent of absolute value of integer. If number of digits in binary equivalent is less than 8, provide zero(s) at the left to make it 8-bit form. 1's complement of a binary number is obtained by replacing every 0 with 1 and every 1 with 0. Some binary numbers and the corresponding 1's compliments are given below:

Binary Number	1's Complement
11001	00110
10101	01010

If the number is negative it is represented as 1's complement of 8-bit form binary. If the number is positive, the 8-bit form binary equivalent itself is the 1's complement representation.

Example: Represent -119 in 1's complement form.

Binary of 119 in 8-bit form $= (01110111)_2$

-119 in 1's complement form $= (10001000)_2$

Example: Represent +119 in 1's complement form.

Binary of 119 in 8-bit form $= (01110111)_2$

+119 in 1's complement form $= (01110111)_2$

(*No need to find 1's complement, since the number is positive*)

Note: In this representation if first bit (MSB) is 0 then number is positive and if MSB is 1 then number is negative. So 8 bit word can represent integers from -127 (represented as 10000000) to +127 (represented as 01111111). Here also integer 0 can be represented in two ways: $+0 = 00000000$ and $-0 = 11111111$. An n-bit word can represent 2^n-1 numbers i.e. $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$.

ii) 2's complement representation

In this method, first find binary equivalent of absolute value of integer and write it in 8-bit form. If the number is negative it is represented as 2's complement of 8-bit form binary. If the number is positive 8-bit form binary itself is the representation. 2's complement of a binary number is calculated by adding 1 to its 1's complement.

For example, let us find the 2's complement of $(10101)_2$.

1's complement of $(10101)_2$ $= (01010)_2$

So 2's complement of $(10101)_2$ $= 01010 +$

$\qquad\qquad 1$

$= (01011)_2$

Example: Represent -38 in 2's complement form.

Binary of 38 in 8-bit form $= (00100110)_2$

-38 in 2's complement form $=$ 11011001+

                                              1

                                          $= (11011010)_2$

Example: Represent +38 in 2's complement form.

Binary of 38 in 8-bit form $= (00100110)_2$

+38 in 2's complement form $= (00100110)_2$

(No need to find 2's complement)

Note: In this representation if first bit (MSB) is 0 then number is positive and if MSB is 1 then number is negative. Here integer 0 has only one way of representation and is 00000000. So an 8 bit word can represent integers from -128 (represented as 10000000) to +127(represented as 01111111). It is the most common integer representation. An *n*-bit word can represent 2^n numbers - (2^{n-1}) to $+(2^{n-1}-1)$.

Table 1.10 shows the various representation methods of integers in 8 bit word length.

Features	Sign & Magnitude	1's complement	2's complement	Remarks
Range	-127 to +127	-127 to +127	-128 to +127	Range is more in 2's comple-ment
Total Numbers	255	255	256	
Representation of integer 0	Two ways for representations	Two ways for representations	Only one way for representa-tion	In 2's comple-ment there is no ambiguity in 0 representation
Representation of positive integers	Binary equiva-lent of integer in 8 bit form	Binary equiva-lent of integer in 8 bit form	Binary equiva-lent of integer in 8 bit form	All three forms are same.
Representation of negative integers	Sign bit 1 and magnitude is represented in 7 bit binary form	Find 1's complement of 8 bit form binary	Find 2's complement of 8 bit form binary	For all negative numbers MSB is 1

Table 1.10 : Representation of integers in 8-bit word length

To compare the three types of representations let us consider the following table. For clarity and easy illustration, 4-bits are used to represent the numbers in this table .

Number	Sign & Magnitude	1's complement	2's complement
-8	Not possible	Not possible	1000
-7	1111	1000	1001
-6	1110	1001	1010
-5	1101	1010	1011
-4	1100	1011	1100
-3	1011	1100	1101
-2	1010	1101	1110
-1	1001	1110	1111
0	1000 or 0000	0000 or 1111	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111

From this table, it is clear that the MSB of a binary number indicates the sign of the corresponding decimal number irrespective of the representation. That is, if the MSB is 1, the number is negative and if it is 0, the number is positive. The table also shows that only 2's complement method can represent the maximum numbers for a given number of bits. This fact reveals that, a number below -7 and above +7 cannot be represented using 4-bits in sign & magnitude form and 1's complement form. So we go for 8-bit representation. Similarly in 2's complement method, if we want to handle numbers outside the range -8 to +7, eight bits are required.

In 8-bits implementation, the numbers from -128 to +127 can be represented in 2's complement method. The range will be -127 to +127 for the other two methods. For the numbers outside this range, we use 16 bits and so on for all the representations.

b. Representation of floating point numbers

A floating point number / real number consists of an integer part and a fractional part. A real number can be written in a special notation called the floating point notation**.** Any number in this notation contains two parts, *mantissa* and *exponent***.**

For example, 25.45 can be written as 0.2545×10^2, where 0.2545 is the mantissa and the power 2 is the exponent. (In normalised floating point notation mantissa is between 0.1 and 1). Similarly -0.0035 can be written as -0.35×10^{-2}, where -0.35 is mantissa and -2 is exponent.

Let us see how a real number is represented in 32 bit word length computer. Here 24 bits are used for storing mantissa (among these the first bit for sign) and 8 bits are used for storing exponent (first bit for sign) as in Figure 1.10. Assumed decimal point is to the right of the sign bit of mantissa. No separate space is reserved for storing decimal point.

Consider the real number 25.45, that can be written as 0.2545×10^2, where 0.2545 is the mantissa and 2 is the exponent. These numbers are converted into binary and stored in respective locations. Various standards are followed for representing mantissa and exponent. When word length changes, bits used for storing mantissa and exponents will change accordingly.
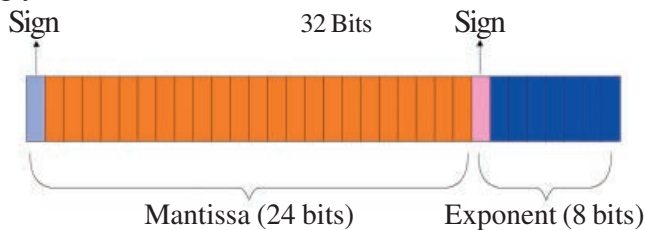

Fig 1.10: Representation of floating point numbers

> In real numbers, binary point keeps track of mantissa part and exponent part. Since the value of mantissa and exponent varies from number to number the binary point is not fixed. In other words it floats and hence such a representation is called floating point representation.

1.8.2 Representation of characters

We have discussed methods for representing numbers in computer memory. Similarly there are different methods to represent characters. Some of them are discussed below.

a. ASCII

The code called ASCII (pronounced "AS-key"), which stands for American Standard Code for Information Interchange, uses 7 bits to represent each character in computer memory. The ASCII representation has been adopted as a standard by the U.S. government and is widely accepted. A unique integer number is assigned to each character. This number called ASCII code of that character is converted into binary for storing in

memory. For example, ASCII code of A is 65, its binary equivalent in 7-bit is 1000001. Since there are exactly 128 unique combinations of 7 bits, this 7-bit code can represent only128 characters.

Another version is ASCII-8, also called extended ASCII, which uses 8 bits for each character, can represent 256 different characters. For example, the letter A is represented by 01000001, B by 01000010 and so on. ASCII code is enough to represent all of the standard keyboard characters.

b. EBCDIC

It stands for Extended Binary Coded Decimal Interchange Code. This is similar to ASCII and is an 8 bit code used in computers manufactured by International Business Machine (IBM). It is capable of encoding 256 characters. If ASCII coded data is to be used in a computer which uses EBCDIC representation, it is necessary to transform ASCII code to EBCDIC code. Similarly if EBCDIC coded data is to be used in a ASCII computer, EBCDIC code has to be transformed to ASCII.

c. ISCII

ISCII stands for Indian Standard Code for Information Interchange or Indian Script Code for Information Interchange. It is an encoding scheme for representing various writing systems of India. ISCII uses 8-bits for data representation. It was evolved by a standardisation committee under the Department of Electronics during 1986-88, and adopted by the Bureau of Indian Standards (BIS). Nowadays ISCII has been replaced by Unicode.

d. Unicode

Using 8-bit ASCII we can represent only 256 characters. This cannot represent all characters of written languages of the world and other symbols. Unicode is developed to resolve this problem. It aims to provide a standard character encoding scheme, which is universal and efficient. It provides a unique number for every character, no matter what the language and platform be.

Unicode originally used 16 bits which can represent up to 65,536 characters. It is maintained by a non-profit organisation called the Unicode Consortium. The Consortium first published the version 1.0.0 in 1991 and continues to develop standards based on that original work. Nowadays Unicode uses more than 16 bits and hence it can represent more characters. Unicode can represent data in almost all written languages of the world.

1.8.3 Representation of audio, image and video

In the previous sections we have discussed different data representation techniques and standards used for the computer representation of numbers and characters. While we attempt to solve real life problems with the aid of a digital computer, in most cases we may have to represent and process data other than numbers and characters. This

may include audio data, images and videos. We can see that like numbers and characters, the audio, image and video data also carry information. In this section we will see different file formats for storing sound, image and video.

Digital audio, image and video file formats

Multimedia data such as audio, image and video are stored in different types of files. The variety of file formats is due to the fact that there are quite a few approaches to compressing the data and a number of different ways of packaging the data. For example an image is most popularly stored in Joint Picture Experts Group (JPEG) file format. An image file consists of two parts - header information and image data. Information such as name of the file, size, modified data, file format, etc are stored in the header part. The intensity value of all pixels is stored in the data part of the file.

The data can be stored uncompressed or compressed to reduce the file size. Normally, the image data is stored in compressed form. Let us understand what compression is. Take a simple example of a pure black image of size 400×400 pixels. We can repeat the information black, black,…., black in all 16,0000(400×400) pixels. This is the uncompressed form, while in the compressed form black is stored only once and information to repeat it 1,60,000 times is also stored. Numerous such techniques are used to achieve compression. Depending on the application, images are stored in various file formats such as BMP (Bitmap file format), TIFF (Tagged Image File Format), GIF (Graphics Interchange Format), PNG - (Portable (Public) Network Graphic).

What we said about the header file information and compression is also applicable for audio and video files. Digital audio data can be stored in different file formats like WAV, MP3, MIDI, AIFF, etc. An audio file describes a format, sometimes referred to as the 'container format', for storing digital audio data. For example WAV file format typically contains uncompressed sound and MP3 files typically contain compressed audio data. The synthesized music data is stored in MIDI(Musical Instrument Digital Interface) files. Similarly video is also stored in different files such as AVI (Audio Video Interleave) - a file format designed to store both audio and video data in a standard package that allows synchronous audio with video simultaneous playback, MP3, JPEG-2, WMV, etc.

Check yourself

1. Which is the MSB of representation of -80 in sign and magnitude method?
2. Write 28.756 in mantissa exponent form.
3. ASCII stands for_____.
4. Represent -60 in 1's complement form.
5. Define Unicode.
6. List any two image file formats.

Let us sum up

Data processing is a series of activities by which data is converted into information. The limitations of manual data processing are overcome by electronic data processing and the computer is the best electronic data processor. A computer has five functional units such as input unit, storage unit, arithmetic logic unit, control unit and output unit. Though the data supplied to computers is of different forms, internally these are represented using bits. Different number systems are associated with computer and any quantity on one system has an equivalent form in another system. Different types of coding systems are available to represent characters in computer. Data in the form of audio, image and video are also stored in binary form and different file formats are used to represent them.

Learning outcomes

After the completion of this chapter the learner will be able to

- distinguish between data and information.
- identify various stages in data processing.
- list the functional units of a computer and explian the functions of each.
- explain why the computer is the best electronic data processing machine.
- infer the concept of data representation inside computers.
- convert a number from one system to another.
- list the features of various coding systems to represent characters.

Sample questions

Very short answer type

1. What is data?
2. Processed data is known as _____.
3. _____ is the place value of 9 in 29610.
4. Hexadecimal number system uses _____symbols and octal number system uses _____ symbols.
5. Find octal equivalent of the decimal number 55.
6. EBCDIC stands for _____.
7. Name the coding system that can represent almost all characters used in the human languages in the world.
8. In sign and magnitude form, if the number is negative, sign bit is _____ and if number is positive, sign bit is ____.

Short answer type

1. Distinguish between data and information.
2. The application form for Plus One admission contains your personal details and your choice of groups and schools.
   a. Identify the data and information in the admission process.
   b. Explain how the information helps the applicants and the school authorities.
   c. Write down the activities involved in the processing of the data.
3. How is a computer superior to human in data processing?
4. Explain the role of storage in data processing activities.
5. List down the functions of input unit of a computer.
6. Is secondary storage essential for a computer? Justify your answer.
7. Write down the role of control unit in a computer?
8. How does the memory unit help CPU to perform its function?
9. "Computers are slaves, humans are masters". Do you agree with this? Give reasons.
10. List down the characteristics of computers.
11. Computer is a versatile machine. How?
12. What is meant by the term diligence, the characteristic of a computer?
13. Define the term data representation.
14. What do you mean by a number system? List any four number systems.
15. Find missing terms in the following series
   a. 101_2 , 1010_2, 1111_2, _____, _____.
   b. 15_8 , 16_8, 17_8, _____, _____.
   c. 18_{16}, $1A_{16}$, $1C_{16}$, _____, _____.
16. Binary number system is used for data representation in computer memory. Why?
17. Convert the following decimal numbers into binary equivalent:
   a. 25   b. 128   c. 255   d. 19.875   e. 89.25
18. Convert the following binary numbers into decimal equivalent:
   a. 1011_2 b. 111001_2 c. 1000001_2 d. 1001.11_2 e. 1111.111_2
19. Convert the following decimal numbers into octal and hexadecimal numbers
   a. 17   b. 75   c. 100   d. 199   e. 256
20. Convert the following binary numbers into octal and hexadecimal
   a. 1011_2 b. 101001_2 c. 11100011_2 d. 110001110_2 e. 10000010001_2
21. Convert the following octal numbers into decimal.
   a. 57_8 b. 101_8 c. 77_8 d. 245_8 e. 1205_8

22. Convert the following hexadecimal numbers into decimal.

    a. $2D_{16}$ b. 101_{16} c. AB_{16} d. $1F8_{16}$ e. ABC_{16}

23. Convert the following octal numbers into binary and hexadecimal.

    a. 67_8 b. 123_8 c. 167_8 d. 745_8 e. 1054_8

24. Convert the following hexadecimal numbers into binary and octal.

    a. $7F_{16}$ b. 207_{16} c. AB_{16} d. $9F8_{16}$ e. ABC_{16}

25. If $(X)_2 = (Y)_8 = (Z)_{16} = (28)_{10}$, then find X, Y and Z.

26. Arrange the following numbers in descending order

    a. $(101)_{16}$ b. 110_{10} c. 111000_2 d. 251_8

27. What are the methods of representing integers in computer memory?

28. Represent the following numbers in sign and magnitude method

    a. -19   b. +49   c. -97   d. -127

29. Represent the following numbers in 1's complement method

    a. -24   b. +69   c. -100   d. -127

30. Represent the following numbers in 2's complement method

    a. -33   b. +71   c. -111   d. -127

31. Represent - 83 in all the three number representation forms.

32. Find out the decimal number which is represented as $(10011001)_2$ in sign and magnitude method.

33. Explain the method of representing a floating point number in a 32 bit computer.

34. What are the methods of representing characters in computer memory?

35. Write short notes on ASCII and ISCII.

36. Briefly explain the significance of Unicode in character representation.

Long answer type

1. Taking the example of a real life situation like banking, briefly describe the activities involved in each stage of data processing.

2. With the help of a block diagram, explain the functional units of a computer.

3. Write down the following numbers in expanded notation

    a. $(1011.101)_2$ b. $(65356)_{10}$ c. $(A5F)_{16}$ d. $(67.4)_8$ e. $(763.452)_{10}$

4. Briefly explain different methods for representing numbers in computer memory.

5. Briefly explain different methods for representing characters in computer memory.

2

Key concepts

- **Hardware**
  - o Processor
  - o Motherboard
  - o Peripherals and ports
  - o Memory (Primary memory, Secondary memory)
  - o Input / output devices
- **e-Waste**
- **Green computing**
- **Software**
  - o System software (Operating system, Language processors, Utility software)
  - o Application software (General purpose, Specific purpose)
  - o Free and open source software concepts
  - o Freeware and Shareware
- **Humanware / Liveware**

Components of the Computer System

We are familiar with computers and their uses today. Computer can be defined as a fast electronic device that accepts data, processes it as per stored instructions and produces information as output. This chapter presents an overview of the basic design of a computer system: how the different parts of a computer system are organised and various operations are performed to do a specific task. We know that a computer has two major components - hardware and software. Hardware refers to all physical components associated with a computer system while software is a set of instructions for the hardware to perform a specific task. When we use computers to solve any problem in real life situations, the tasks are usually set up to process data to generate information. This chapter presents the various hardware components followed by electronic waste, its disposal methods and the concept of green-computing. Different classifications of software are also dealt with. We also discuss the concepts of free and open source, freeware, shareware and proprietary software.

2.1 Hardware

We know that a computer system consists of hardware and software. The term hardware represents the tangible and visible parts of a

computer, which consists of some electromechanical components. These hardware components are associated with the functional units of a computer. Let us discuss some of these components.

2.1.1 Processors

In high school classes we learned that the Central Processing Unit (CPU) / processor is responsible for all computing and decision making operations and coordinates the

Fig. 2.1 : Central Processing Units

working of a computer. The performance of a CPU determines the overall performance of the computer (refer Figure 2.1). Since CPU is an Integrated Circuit (IC) package which contains millions of transistors and other components fabricated into a single silicon chip, it is also referred as microprocessor. A CPU is usually plugged into a large socket on the main circuit board (the motherboard) of the computer. Since heat is generated when the CPU works, a proper cooling system is provided with a heat sink and fan. Intel core i3, core i5, core i7, AMD Quadcore, etc. are some examples of processors.

Every computer contains an internal clock that regulates the rate at which instructions are executed. The CPU requires a fixed number of clock ticks (or clock cycles) to execute each instruction. The faster the clock, the more instructions the CPU can execute per second. Another factor is the architecture of the chip. The number of bits a processor can process at one time is called word size. Processors with many different word sizes exist: 8-bit, 16-bit, 32-bit, 64-bit, etc.

Registers are storage locations inside CPU, whose contents can be accessed more quickly by the CPU than other memory. They are temporary storage areas for instructions or data. They are not a part of memory; rather they are special additional storage locations that offer computers the advantage of speed. Registers work under the direction of the control unit to accept, hold, and transfer instructions or data and perform arithmetic or logical operations at high speed. It speeds up the execution of programs.

Important registers inside a CPU are:

a. **Accumulator:** The accumulator is a part of the Arithmetic Logic Unit (ALU). This register is used to store data to perform arithmetic and logical operation. The result of an operation is stored in the accumulator.

b. **Memory Address Register (MAR):** It stores the address of a memory location to which data is either to be read or written by the processor.

c. **Memory Buffer Register (MBR):** It holds the data, either to be written to or read from the memory by the processor.

d. **Instruction Register (IR):** The instructions to be executed by the processor are stored in the instruction register.

e. **Program Counter (PC):** It holds the address of the next instruction to be executed by the processor.

2.1.2 Motherboard

A motherboard is a large Printed Circuit Board (PCB) to which all the major components including the processor are integrated. Figure 2.2 shows the motherboard and its components. It also provides expansion slots for adding additional circuit boards like memory, graphics card, sound card, etc. The motherboard must be compatible with the processor chosen.
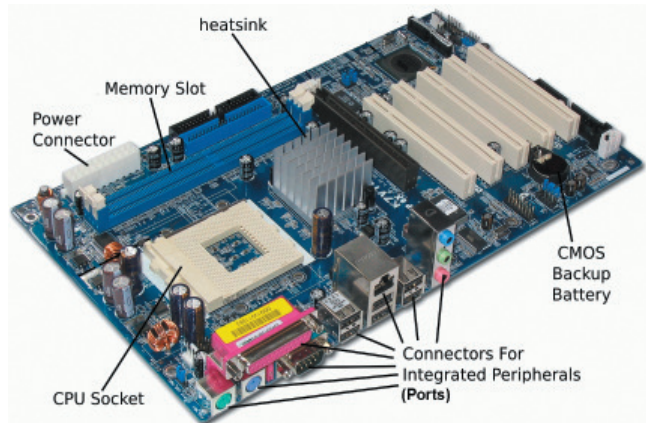

Fig. 2.2 : Motherboard

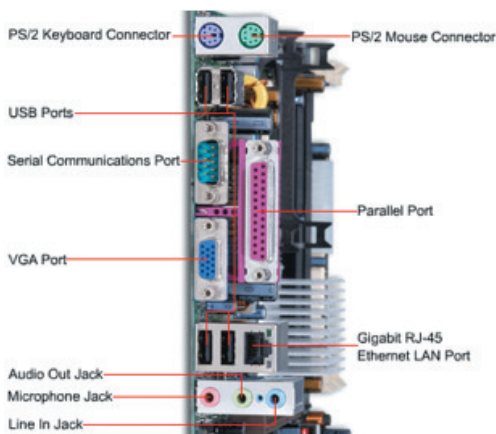### 2.1.3 Peripherals and ports

Fig. 2.3 : Ports

Peripherals are devices that are attached to a computer system to enhance its capabilities. Ports on the motherboard are used to connect external devices. Figure 2.3 shows the various ports in a computer system. Peripherals include input devices, output devices, external storage and communication devices. Peripheral devices communicate with the motherboard, through the ports available on the motherboard like Video Graphics Array (VGA), IBM Personal System/2 (PS/2), Universal Serial Bus (USB), Ethernet, High Definition Multimedia Interface (HDMI), etc. Let us look at some kinds of ports used on personal computers.

a. Serial port

A serial port / serial communication port transmits data one bit at a time. In older computers devices such as modem, mouse or keyboard were connected through serial ports (refer Figure 2.4).
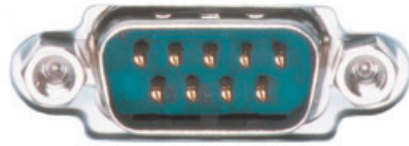

Fig 2.4: Serial port

Serial cables are cheaper to make and easier to shield from interference. Since its speed is too low, they are being replaced by faster ports like PS/2, USB, etc.

b. Parallel port

Parallel ports can transmit several bits of data simultaneously. It is faster than serial port and is used to connect a printer or scanner to the computer. Figure 2.5 shows a parallel port in a computer system.

Fig. 2.5 : Parallel port

c. USB port

USB (Universal Serial Bus) is a connection that provides high speed data communication between devices. Due to its high bandwidth data transfer is faster. It is used for short-distance communication. USB port is used for connecting devices like keyboard, mouse, printer, scanner, flash drive, external hard disk, etc. Figure 2.6 shows the various types of USB ports. The main advantage of USB ports are:
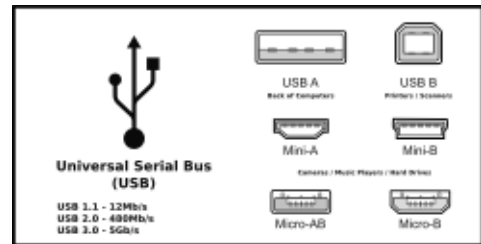

Fig. 2.6: USB ports

- USB ports are capable of supplying electric power to external devices. This technology led to the development of devices like external hard disk, flash drive, dongle, etc. which draw power from the USB port. It can also be used for power devices like mobile phones, tablets, laptops, etc.

- USB devices can be connected and disconnected even when the power is on .

d. LAN port

Alternatively referred to as an Ethernet port, network connection, and network port, the Local Area Network (LAN) port is a port connection that allows a computer to connect to a network using a wired connection. The registered jack RJ45 is a standard type of connector used for connecting cables through LAN ports (refer Figure 2.7).

Fig. 2.7 : LAN port and RJ45 connector

e. PS/2 port

Personal System/2 (PS/2) ports are special ports invented by IBM (International Business Machines) for connecting the keyboard and mouse (refer Figure 2.8). These types of ports are too slow and replaced by faster ports like USB nowadays.
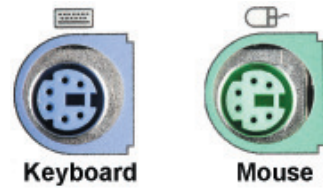

Fig. 2.8: PS/2 port

f. Audio ports

Audio ports are used to connect audio devices like speakers, microphone, etc. The three small connectors shown in Figure 2.9 are used for connecting:

a. Line in - Sound input (from any audio device like mobile line out, ipod, etc.)

b. Line out - Sound out (to connect your PC's sound output to external speakers)

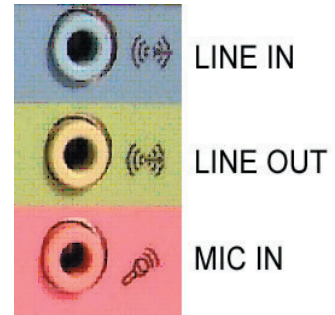c. Mic in - Microphone input port.

Fig. 2.9: Audio port

g. Video Graphics Array (VGA) port

VGA is a popular display standard developed by IBM. The VGA port is used to connect a monitor or a projector to a computer. The VGA connector has 15 pins displayed in three rows as indicated in Figure 2.10. Super VGA (SVGA) was developed to support 800×600 resolution and was later replaced by Extended Graphics Array (XGA) that supports 1024 by 768 resolution. Regardless of what resolution a monitor is able to support, they are commonly referred to as VGA.

Fig. 2.10 : VGA port

h. High Definition Multimedia Interface (HDMI)

HDMI is a type of digital connection capable of transmitting high-definition video and multi channel audio over a single cable (refer Figure 2.11). To do the same thing with analog cables, we need to connect several video and audio cables.

Fig. 2.11 : HDMI port

2.1.4 Memory

Memory is a place where we can store data, instructions and results temporarily or permanently. Memory can be classified into two: primary memory and secondary memory. Primary memory holds data, intermediate results and results of ongoing jobs temporarily. Secondary memory on the other hand holds data and information permanently. Before learning more about memory, let us discuss the different memory measuring units.

Binary Digit	= 1 Bit	1 MB (Mega Byte)	= 1024 KB
1 Nibble	= 4 Bits	1 GB (Giga Byte)	= 1024 MB
1 Byte	= 8 Bits	1 TB (Tera Byte)	= 1024 GB
1 KB (Kilo Byte)	= 1024 Bytes	1 PB (Peta Byte)	= 1024 TB

a. Primary storage

Primary memory is a semiconductor memory that is accessed directly by the CPU. It is capable of sending and receiving data at high speed. This includes mainly three types of memory such as RAM, ROM and Cache memory.

i. Random Access Memory (RAM)

RAM refers to the main memory that microprocessor can read from and write to. Data can be stored and retrieved at random from anywhere within the RAM, no matter where the data is. Data or instructions to be processed by the CPU must be placed in the RAM (refer Figure 2.12). The contents of RAM are lost when power is switched off. Therefore, RAM is a volatile memory. The storage capacity of RAM is 2 GB and above
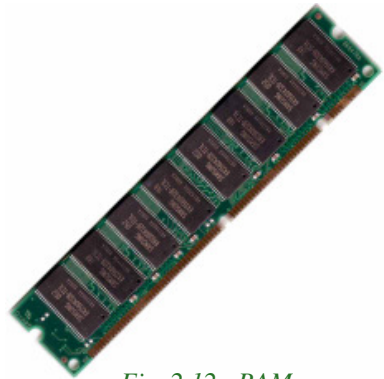
Fig. 2.12 : RAM

The speed of a RAM refers to how fast the data in memory is accessed. It is measured in Mega Hertz (MHz). When a computer is in use, its RAM contains the following:

1.   The operating system software.

2.   The application software currently being used.

3.   Any data that is being processed.

ii. Read Only Memory (ROM)

ROM is a permanent memory that can perform only read operations and its contents cannot be easily altered. ROM is non-volatile; the contents are retained even after the power is switched off. ROM is used in most computers to hold a small, special piece of 'boot up' program known as Basic Input Output System (BIOS). This software runs when the computer is switched on or 'boots up'. It checks the computer's hardware and then loads the operating system. It is slower than RAM. Figure 2.13 shows a typical ROM chip.
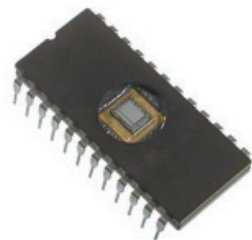
Fig. 2.13 : ROM chip

There are some modified types of ROM that include:

1. **PROM** - Programmable ROM which can be programmed only once. PROMs are programmed at the time of manufacture.

2. **EPROM** - Erasable Programmable ROM that can be erased using ultra violet radiation and can be programmed using special electronic circuits.

3. **EEPROM** - Electrically Erasable Programmable ROM which can be erased and rewritten electrically.

Table 2.1 shows the comparison between RAM and ROM.

RAM	ROM
• It is faster than ROM	• It is a slower memory
• It stores the operating system, application programs and data when the computer is functioning.	• It stores the program required to boot the computer initially
• It allows reading and writing.	• Usually allows reading only.
• It is volatile, i.e. its contents are lost when the device is powered off.	• It is non-volatile, i.e. its contents are retained even when the device is powered off.

Table 2.1 : Comparison of RAM and ROM

iii. Cache memory

Cache memory is a small and fast memory between the processor and RAM (main memory). Frequently accessed data, instructions, intermediate results, etc. are stored in cache memory for quick access. When the processor needs to read from or write to a location in RAM, it first checks whether a copy of that data is in the cache. If so, the processor immediately reads the cache, which is much faster than reading from RAM. Cache is more expensive than RAM, but it is worth using it in order to maximise system performance. Commonly used cache memories and their capacities are Level 1 Cache (128 KB), Level 2 Cache (1 MB), Level 3 Cache (8 MB) and Level 4 Cache (128 MB).

b. Secondary or Auxiliary memory

Secondary memory is of permanent nature. Unlike the contents of RAM, the data stored in these devices does not vanish when power is turned off. Secondary memory is much larger in size than RAM, but is slower. It stores programs and data but the processor cannot access them directly. Secondary memory is also used for transferring data or programs from one computer to another. It also acts as a backup. The major categories of storage devices are magnetic, optical and semiconductor memory.

i. Magnetic storage devices

Magnetic storage devices use plastic tape or metal/plastic disks coated with magnetic materials. Data is recorded magnetically in these devices. Read/write heads are used to access data from these devices. Some of the popular magnetic storage devices are magnetic tapes, floppy disks, hard disks, etc.

Magnetic Tape

Magnetic tape is a backup device that can store huge volume of data. The cost per unit volume of a magnetic tape is less compared to other magnetic storage devices.

It is a recording medium consisting of a thin tape with a coating of fine magnetic material. It can be used for recording analog or digital data. Data is stored in frames across the width of the tape with a read/write head. The frames are grouped into blocks or records which are separated from other blocks by gaps as shown in Figure 2.14.
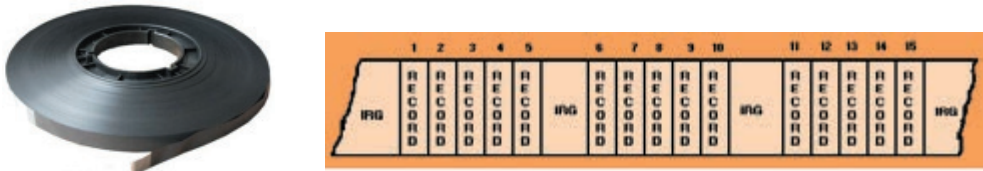

Fig. 2.14 : Magnetic Tape and Blocks of data on tape

It takes a lot of time to locate data in a magnetic tape, as it is a sequential access medium, similar to an audio cassette. For example, if we need to access the 100th record on the tape, we must first move through the previous 99 records. The capacity of tape media is referred in Terabytes.

Hard disk

The hard disk consists of metal disks coated with magnetic material concealed in dust free containers. Hard disks have very high storage capacity, high data transfer rates and low access time. It is more durable and less error prone. It is the most common secondary storage device used in computers.

A hard disk may contain one or more platters. Each platter requires two read/write heads, one for each surface. A tiny electromagnetic read/write head attached to an access arm magnetises tiny spots on the disk to store data as shown in Figure 2.15. The same electromagnetic head can later sense the magnetic fields of the spots, allowing the data to be read back from the disk.
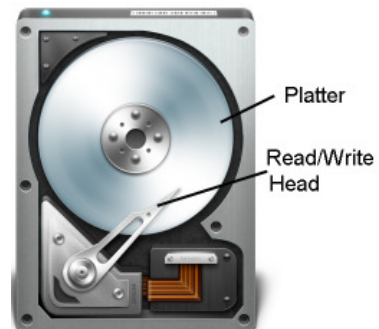

Platter

Read/Write Head

Fig. 2.15 : Hard disk

Data is stored on the surface of a platter in sectors and tracks. Tracks are concentric circles on a platter and a sector is a pie-sliced part of a disk platter as shown in Figure 2.16. Formatting is the activity of creating sectors and tracks on the disk. Only then can read/write operation be performed on the disk. If a disk having data is formatted, then all data will be lost.

Large disks can be partitioned logically into distinct sections or volumes. These volumes are independent of each other and can be formatted independently.
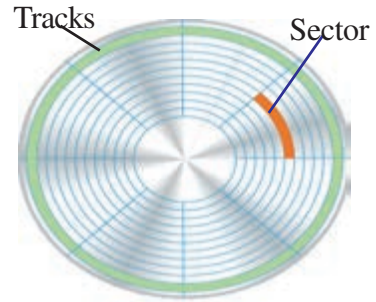

Fig. 2.16 : A platter

Rating of Hard Disk Drive (HDD) depends on capacity, speed and access time

Storage capacity	:	500GB, 1TB or more
Speed	:	How fast the disk spins, rotations per minute (rpm) usually 5400 rpm/7200 rpm
Access time	:	Time to retrieve bits of data from disk (in milliseconds)

ii. Optical storage devices

Optical disk is a data storage medium which uses low-powered laser beam to read from and write data into it. It consists of an aluminum foil sandwiched between two circular plastic disks. Data is written on a single continuous spiral in the form of pits and lands as shown in Figure 2.17. The laser beam reads this pits and lands as 0s and 1s. It is very cheap to manufacture optical disks in large quantities and are a popular secondary storage medium. The main types of optical disks are CD, DVD and Blu-Ray.
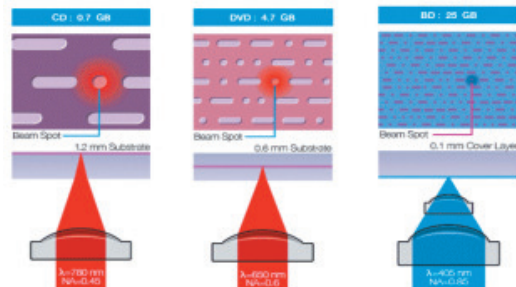

Fig. 2.17 : CD, DVD and Blu-ray disk - pits & lands

Compact Disk (CD)

Compact Disc is an optical storage medium capable of storing upto 700 MB of data. A CD drive uses red laser beams for reading from and writing data into CD. There are two types of CDs, CD-R and CD-RW. In CD-R (CD-Recordable) data can be written

Fig. 2.18 : CD

once and read many times where in CD-RW (CD-Rewritable) disks can be erased and rewritten at any time.

Digital Versatile Disk (DVD)

Digital Versatile Disk is an optical storage media similar to CD-ROM, but with a higher storage capacity. This is achieved by using smaller spots to record data. Recording and reading of data is done using DVD drive. Here also red laser beam is used for doing these operations. The capacity of a DVD varies from 4.37 GB to 15.9 GB. There are three major types of DVDs: DVD-ROM, DVD-RW and DVD-RAM.

DVD-ROM   :   DVD-Read Only Memory functions in the same way as CD-ROM

DVD-RW   :   DVD-Rewritable disks can be erased and rewritten at any time.

DVD-RAM   :   DVD Random Access Memory disks can be recorded and erased repeatedly. These are compatible only with devices manufactured by companies that support the DVD-RAM format. It is comparable to DVD-Rewritable disk, but have higher lifetime and can be erased more often than a DVD-RW.

A normal DVD known as DVD-5 stores 4.37 GB data where as a dual layer double side DVD known as DVD-18 can store 15.9 GB.

Blu-ray DVD

Blu-ray is an optical disk format developed to enable recording, rewriting and playback of High Definition (HD) video as well as storing huge amounts of data. CD and DVD technologies use red laser to read and write data while blu-ray format uses a blue-violet laser. Hence it has the name Blu-ray. The benefit of using a blue-violet laser is that it has a shorter wavelength than a red laser, which makes it possible to focus the laser spot with greater precision. This allows data to be packed more tightly. Therefore it is possible to store more data on the disk even though it is of the same size of a CD/DVD. The format offers more than five times the storage capacity of traditional DVDs and can hold up to 25 GB on a single-layer disc and 50 GB on a dual-layer disc.

iii. Semiconductor storage (Flash memory)

Flash drives use EEPROM chips for data storage. They do not contain any moving parts and hence they are shockproof. Flash memory is faster and durable when compared to other types of secondary memory. The different variants of flash memories are available today.

USB flash drive

A flash drive is a small external storage device, which consists of flash memory typically the size of a human thumb. USB flash drives are portable and rewritable. The storage capacity of a USB drive currently varies from 2 GB to 32 GB.

Fig. 2.19 : Flash drive

Flash memory cards

Flash memory card is another type of flash memory. They are flat and have a size of about 1 inch × 0.75 inch with a thickness of about 2 mm. Memory cards currently have storage capacities in the range of 1 GB - 32 GB. Flash memory cards also have a smaller version which is used within cell phones, tablets, etc (refer Figure 2.20). These smaller cards are about 6 mm × 3 mm in size and are less than 1 mm thick.
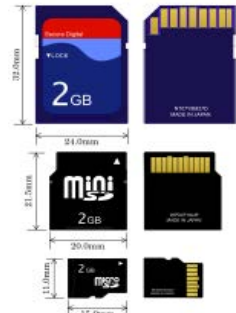

Fig. 2.20 : Flash memory cards

Role of different types of memories in computer

Let us discuss an example of data processing. Consider the case of a payroll program to calculate the salary of an employee. The data for all the employees is available in the hard disk. All the data about a particular employee is taken to the RAM and from there data related to salary calculation (bonus, deductions, etc.) is taken to the cache. The data representing the hours worked and the wages is moved to their respective registers. Using data on the hours worked and the wage, ALU makes calculations based on instructions from control unit. For further calculations, it moves the overtime hours, bonuses, etc. from cache to registers. As the CPU finishes calculations about one employee, the data about the next employee is brought from secondary storage into RAM, then cache and eventually into the registers.
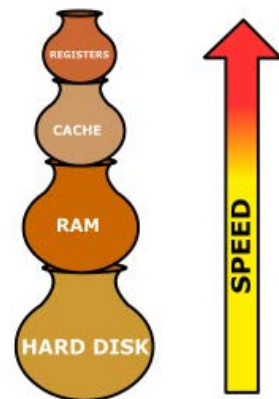

Fig. 2.21 : Memory hierarchy

Table 2.2 summarises the characteristics of the various kinds of data storage in the storage hierarchy. Modern computers are designed with this hierarchy due to the characteristics listed in the table.

To see how registers, primary memory and second storage work together, let us use the analogy of making a salad. In our kitchen we have:

- A refrigerator where we store vegetables for the salad.
- A counter where we place all vegetables before putting them on the cutting board for chopping.
- A cutting board on the counter where we chop vegetables.
- A recipe that details what vegetables to chop.
- The corners of the cutting board are kept free for partially chopped piles of vegetables that we intend to chop more or to mix with other partially chopped vegetables.
- A bowl on the counter where we mix and store the salad.
- Space in the refrigerator to put the mixed salad after it is made.

The process of making the salad is then: bring the vegetables from the fridge to the counter top; place some vegetables on the chopping board according to the recipe; chop the vegetables, possibly storing some partially chopped vegetables temporarily on the corners of the cutting board; place all the vegetables in the bowl and keep it back in the fridge if not served on the dinner table.

In this context the refrigerator serves as secondary (hard disk) storage. It can store high volumes of vegetables for long periods of time. The counter top functions like the computer's motherboard - everything is done on the counter (inside the computer). The cutting board is the ALU - the work gets done there. The recipe is the control unit - it tells you what to do on the cutting board (ALU). Space on the counter top is the equivalent of RAM - all required vegetables must be brought from the fridge and placed on the counter top for fast access. Note that the counter top (RAM) is faster to access than the fridge (disk), but cannot hold as much, and cannot hold it for long periods of time. The corners of the cutting board where we temporarily store partially chopped vegetables are equivalent to the registers. The corners of the cutting board are very fast to access for chopping, but cannot hold much. The salad bowl is like a cache memory, it is for storing chopped vegetables to be temporarily removed from the corners of the cutting board (as there is too much) or the salad waiting to be taken back to the fridge (putting data back on a disk) or to the dinner table (outputting the data to an output device).

Storage	Speed	Capacity	Relative Cost	Volatile
Registers	Fastest	Lowest	Highest	Yes
Cache	More Fast	Low	Very High	Yes
RAM/ROM	Very Fast	Low/Moderate	High	Yes
Hard Disk	Moderate	Very High	Very Low	No

Table 2.2 : Comparison of different types of memory

Check yourself

1. The fastest memory in a computer is _____.
2. The storage capacity of a single layer DVD is _____.
3. What is cache memory?
4. What is the use of program counter register?
5. What is HDMI?

2.1.5 Input/Output devices

The computer will be of no use unless it is able to communicate with the outside world. Input/output devices are required for users to communicate with the computer. In simple terms, input devices feed data and instructions into the computer and output devices present information from a computer system. These input/output devices are connected to the CPU through various ports or with the help of wireless technologies. Since they reside outside the CPU, they are called peripherals.

a. Input devices

An input device is used to feed data into a computer. It is also defined as a device that provides communication between the user and the computer. We will now discuss some input devices in detail.

i. Keyboard

Keyboard is the most common input device. It allows the user to input alphabets, numbers and other characters. Keyboard detects the key pressed and generates the corresponding ASCII code which can be recognised by the computer. The standard US keyboard introduced in 1986 has 101 keys. It has a keyboard layout called the

QWERTY design. QWERTY gets its name from the first six letters across in the upper-left-hand corner of the keyboard as shown in Figure 2.22.
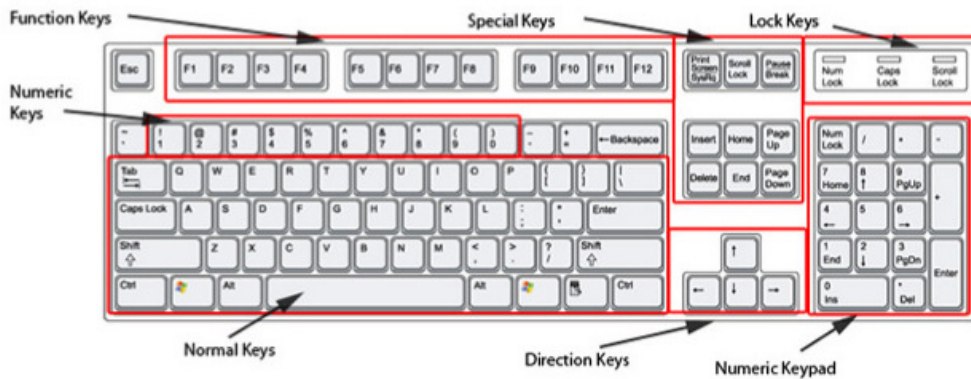

Fig. 2.22 : Keyboard

Keyboards can be classified as wired and wireless. Wired keyboards are connected to the CPU through a serial, PS/2 port or a USB port. Wireless keyboards are connected to the computer through infrared (IR), radio frequency (RF) or bluetooth connections. Portable flexible keyboards are also available now. New generation keyboards like laser keyboards that project the keyboard layout to any surface are being developed.

ii. Mouse

A mouse is a small hand-held device used to indicate the position of a cursor or its movement on a computer display screen by rolling it over a mouse pad / flat surface. A mouse has one or more buttons and possibly a scroll wheel. Scroll wheel is used for scrolling the screen vertically or horizontally. The different types of mouse are ball, optical and laser mouse. Ball mouse works on the principle of the movement of the ball, whereas optical mouse uses LED and laser mouses use laser beams for sensing the movement. Laser mouse has more precise movements when compared to other types of mouse. Wired mouse uses serial, PS/2 and USB ports to communicate, whereas a wireless mouse communicates with the computer via radio waves.

Fig. 2.23 : Mouse

iii. Light pen

A light pen is a pointing device shaped like a pen. The tip of the light pen contains a light-sensitive element which when placed against the screen, detects the light from the

screen enabling the computer to identify the location of the pen on the screen. Light pens have the advantage of 'drawing' directly onto the screen. They are used by engineers, artists, fashion designers for Computer Aided Designing (CAD) and drawing purposes.

Fig. 2.24 : Light Pen

iv. Touch screen

It is an input device that allows the user to operate by simply touching on the display screen. Some computers, tablets, smart phones, etc. have touch-sensitive display screens. It can also be operated using a stylus which gives more precision. Information kiosks at railway stations and bank ATMs also use touch screens as input device.

Fig. 2.25 : Touch screen

v. Graphic tablet

A graphics tablet consists of an electronic writing area and a special "pen" that works with it. Graphic tablet allows artists to create graphical images with motions and actions similar to traditional drawing tools. The pen of the graphics tablet is pressure sensitive. Hard or soft pressure on the tablet using the pen can result in brush strokes of different width in an appropriate graphics program.
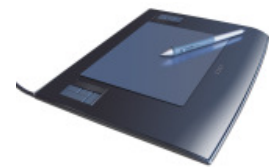
Fig. 2.26 : Graphic tablet

vi. Touchpad

A touchpad is a pointing device found on the portable computers and some external keyboards. It allows us to move the mouse pointer without the need of an external mouse. Touchpad is operated by using finger and dragging it across the flat surface; as the finger moves on the surface, the mouse cursor will move in that same direction. The touchpad also has two buttons below the touch surface that enables to click.

Fig. 2.27 : Touchpad

vii. Joystick

Joystick is an input device used for playing video games, controlling training simulators and robots. Joysticks and other game controllers can also be used as pointing device. The joystick has a vertical stick which can move in any direction. It can be used to control

Fig. 2.28 : Joystick

objects in a video game or to make menu selections by the movement of a cursor displayed on the screen. It has a button on the top that is used to select the option pointed by the cursor.

viii. Microphone

A microphone can be attached to a computer to input sound. It accepts sound which is analogue in nature as input and converts it to digital format. The digitised sound can be stored in the computer for processing or playback. A computer loaded with speech recognition software like the one preinstalled in Windows 7, can convert what a person has said into text, which can be saved for word processing. A voice recognition program can process the input and convert it into machine-recognisable commands.

Fig. 2.29 : Microphone

ix. Scanner

Scanners can capture information, like pictures or text, and convert it into a digital format that can be edited using a computer. The quality of the image depends on the resolution of the scanner. The resolution of the image scanned is expressed in Dots Per Inch (DPI). The higher the DPI, the better the resolution.

The different variants of scanners are flat bed, sheet feed and hand held scanner. A sheet feed scanner can scan a single sheet, whereas flat bed can scan even from a book, but they are not portable. A hand held scanner is portable but the scanning action is not smooth as the scanner is moved manually.

Fig. 2.30 : Scanner

Optical Character Recognition (OCR) software is used to recognise the printed text in an image scanned and convert it into proper text format, which can be edited by a text editor. Advanced OCR system can read printed text in a large variety of fonts but has difficulty with hand written text. Accurate OCR SDK, Hindi OCR software, Akshara Malayalam OCR, etc. are examples of OCR softwares.

x. Optical Mark Reader (OMR)

OMR technology scans a printed form and reads predefined positions, and records the marks on the form. This technology is useful for applications in which large number of hand-filled forms need to be processed quickly with great accuracy, such as objective type tests and questionnaires.

Fig. 2.31 : Optical Mark Reader

OMR sheets are normaly use to evaluate multiple choice questions in competitive exams. It consists of bubble shaped options to mark answers. Candidates are required to darken the correct bubble option using a pen or pencil. OMR readers can recognise these marks and the appropriate software uses this input for evaluation (refer Figure 2.31). For accuracy of results, good quality paper and accurate alignment of printing is essential.
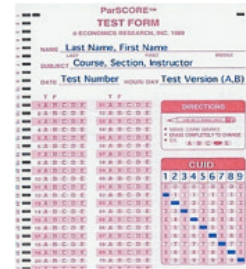

Fig. 2.32 : OMR sheet

xi. Barcode/Quick Response (QR) code reader

A barcode is a set of vertical lines of different thickness and spacing that represent a number. Barcode readers are devices that are used to input data from such set of barcodes (refer Figure 2.33). Hand-held scanners are commonly seen in shops to scan codes and price information for each of the items so that billing will be easier. Mobile phones with camera and special software can also be used as a barcode reader.

Fig. 2.33 : Barcode reader

A QR code is similar to barcodes. Barcodes are single dimensional whereas QR codes are two dimensional as shown in Figure 2.34. The two dimensional way of storing data allows QR code to store more data than a standard barcode. This code can store website URLs, plain text, phone numbers, email addresses and any other alphanumeric data. The QR code can be read using a barcode reader or a mobile phone with a camera and a special software installed.

Fig. 2.34 : QR Code

xii. Magnetic Ink Character Recognition (MICR) Reader

MICR readers are used in banks for faster electronic clearing of cheques. The lower portion of a cheque contains cheque number, branch code, bank code, etc. printed in a special font using an ink containing iron oxide particles as shown in Figure 2.35. Iron oxide has magnetic properties. MICR reader shown in Figure 2.36, can easily recognise these characters by magnetically charging them while scanning. This MICR data along with the image of the cheque is send to the cheque drawer's (the person who issues the cheque) branch to transfer the amount. This reduces errors in data entry and speeds up money transfer.
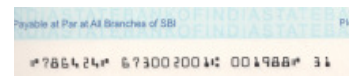

Fig. 2.35 : MIC Code

Fig. 2.36 : MICR Reader

xiii. Biometric sensor

A biometric sensor is a device that identifies unique human physical features with high accuracy. It is an essential component of a biometric system which uses physical features

like fingerprints, retina, iris patterns, etc., to identify, verify and authenticate the identity of the user. The three major types of biometric sensors are semiconductor sensor, optical sensor and ultrasound sensor. Figure 2.37 shows a finger print sensor.

Fig. 2.37: Biometric sensor

xiv. Smart card reader

A smart card is a plastic card that stores and transacts data. The data card may contain a memory or a microprocessor. Memory cards simply store data, while a microprocessor card, on the other hand, can add, delete and manipulate information in its memory. The smart card is used in most banking, healthcare, telephone calling, electronic cash payments and other applications.

Fig. 2.38: Smart card reader

Smart card readers are used to access data in a smart card. It can be contact type or contactless. A contact type of reader requires physical contact with the cards, which is made by inserting the card into the reader. A contactless type of reader works with a radio frequency that communicates when the card comes close to the reader. Many contactless readers are designed specifically for toll gate payment in transportation applications and person identity applications.

xv. Digital camera

A digital camera can take pictures and videos and convert them into digital format. Pictures or videos taken using a digital camera are stored inside its memory and can be transferred to a computer by connecting the camera to it.

Fig. 2.39 : Digital Camera

The quality of the lens, the density of Charge Couple Device (CCD), resolution (measured in megapixel), optical zoom and the software used in the camera determines the quality of the picture. Each picture is made up of thousands of tiny pixels (picture elements) and the camera stores the data about the color of each dot. The quality of the picture is determined by the number of pixels in each picture. Digital cameras have resolutions ranging from 2 mega pixel to 24 megapixels and optical zoom ranging from 3x to 60x.

Fig. 2.40 : Web Camera

Web camera is a compact and less expensive version of a digital camera. It is used in computers for video calling, video chatting, etc. It does not have an internal memory. Applications like Skype, Yahoo Messenger, etc. use webcam to capture images. Now, laptops also come with an inbuilt web camera.

b. Output devices

Output devices are devices that print/display output from a computer. Outputs generated by the output devices may be hardcopy output or softcopy output. Hardcopy outputs are permanent outputs which can be used at a later date or when required. They produce a permanent record on paper. The common output devices that produce hardcopy outputs are printers and plotters. Softcopy outputs are electronic and are available on the screen in a digital form. They do not produce a permanent record. A common softcopy output device is the Visual Display Unit (VDU).

i. Visual Display Unit (VDU)

A Visual Display Unit (VDU) is an output device that visually conveys text, graphics and video information. Information shown on a display device is called softcopy because the information exists electronically and is displayed for a temporary period of time. Display devices include Cathode Ray Tube (CRT) monitors, Liquid Crystal Display (LCD) monitors, Thin Film Transistor (TFT) monitors, Light Emitting Diode (LED) monitors, gas plasma monitors.

Some of the characteristics of a VDU are size, resolution, pixel-pitch and response time. VDUs are available in different sizes. The size of a monitor is measured diagonally across the screen, in inches. The resolution of the monitor is the maximum number of pixels it can display horizontally and vertically (such as 800×600 or 1024×768 or 1600×1200). The pixel spacing on the screen is called the dot pitch. A screen with smaller dot pitch produces sharper images. Response time refers to the time taken for a pixel to turn from a state of brightness to a state of darkness and then back again. Monitors, with lesser response time provide better movie viewing experience.

Cathode Ray Tube (CRT) monitor

The Cathode Ray Tube (CRT) monitor resembles television sets of the past. Two types of CRT monitors are available, monochrome and color. A monochrome monitor displays characters and images in a single colour on a dark background. Another variation of monochrome monitor capable of displaying different shades of grey is called a grey scale monitor. A colour monitor uses three different basic colours such as red, blue and green to display 16 to 1 million different colours. These monitors are preferred by some graphic artists for their accurate colour rendering and by some gamers for faster response to rapidly changing graphics.

Fig. 2.41: CRT monitor

Flat panel monitor

Flat panel displays are thinner, lighter in weight, consume less power and emit less heat compared to CRT monitors. Flat panel monitors are most commonly used in computers, especially in laptops. Different types of flat panel monitors are LCD Monitors, LED Monitors, Plasma Monitors and OLED Monitors.

Fig. 2.42: LCD monitor

Liquid Crystal Display (LCD) Monitors: LCD displays consists of liquid crystals sandwiched between two plastic plates. These crystals rearrange to form an image when an electric current is passed through them. A light source at the back of this plate makes the picture visible. This light source can be fluorescent lamp or LED.

Light Emitting Diode (LED) Monitors: LED monitors use LED directly behind the liquid crystal display (LCD) in order to light up the screen. This technique is very effective and gives each area of the screen its own light, which can be on or off. LED screens can produce massive contrast ratio making the difference between the lights and the blacks appear almost perfect. This technology is expensive. The advantage of using LED is better color quality, clarity, wider viewing angle, faster refresh rates and power savings.

Plasma Monitors: A flat-panel display consists of sandwiching neon/xenon gas between two sealed glass plates with parallel electrodes deposited on their surfaces. When a voltage pulse is passed between two electrodes, the gas lights up as different colours, creating images on a monitor. Plasma monitors provide high resolution but are expensive.

Organic Light Emitting Diode (OLED) Monitors: The panel of OLED is made up of millions of tiny LEDs. The O in OLED stands for organic, which means there is carbon in the light emitting layer of the panel. OLED screens are thinner and lighter than LCDs and LEDs. They can produce better quality images and have a better viewing angle. OLEDs consume less power, but are very expensive.

LCD projector

An LCD projector is a type of video projector for displaying video, images or computer data on a large screen or other flat surface. It is a modern equivalent of the slide projector or overhead projector. A beam of high-intensity light travels through thousands of shifting pixels in an LCD display. This beam of light then passes through a lens which projects and focuses the image on the surface.

Fig. 2.43 : LCD projector

ii. Printer

Printers are used to produce hardcopy output. Based on the technology used, they can be classified as impact or non-impact printers. Impact printers use the typewriting or printing mechanism where a hammer strikes the paper through a ribbon in order to produce output. Dot-matrix printers fall under this category. Non-impact printers do not touch the paper while printing. They use different technologies to print characters on paper. Inkjet, Laser and Thermal printers fall under this category of printers.

Two factors that determine the quality of a printer are its resolution and speed. Resolution is measured in terms of DPI. Speed is measured in terms of number of characters printed in a unit of time and is represented as characters per second (cps), lines per minute (lpm), or pages per minute (ppm).

Dot Matrix Printer (DMP)

Dot matrix printers use small electromagnetically activated pins in the print head and an inked ribbon, to produce images by impact. The most commonly used printer heads consists of 9 pins. Certain printers use 24 pins for better print quality. These printers are slow and noisy, and are not commonly used for personal use. The Dot Matrix Printers are widely used at cash counters in shops due to their low printing cost and for the reason that we get carbon copies from them.

Fig. 2.44: DMP Printer

Inkjet printer

Inkjet printers form the image on the page by spraying tiny droplets of ink from the print head. The printer needs several colours of ink (cyan, yellow, magenta and black) to make colour images. Some photo-quality ink jet printers have more colours of ink. Ink jet printers are inexpensive, but the cost of ink cartridges makes it a costly affair in the long run.

Fig. 2.45: Inkjet Printer

Laser printer

A laser printer produces good quality output. The image to be printed is transferred to a drum using a laser beam. The toner powder from the toner cartridge is then sprayed on the drum. The toner powder sticks onto the portions traced on the drum by the laser beam. It is transferred to a paper by rolling the paper over the drum. Through heating the powder is fused on to the paper.

Fig. 2.46: Laser printer

Monochrome and colour laser printers are available. Colour laser printers use multiple colour toner cartridges to produce colour output and are expensive. Laser printers are faster and their speed is rated in pages per minute (ppm).

Thermal printer

Thermal printing produces a printed image by selectively heating heat-sensitive thermal paper when it passes over the thermal print head. The coating turns black in the areas where it is heated, producing an image. Thermal printers print quiet and faster than dot matrix printers. They are also smaller, lighter and consume less power, making them ideal as portable printers. Thermal printers are popular as printers at Point-of-Sale terminals.

Fig. 2.47 :

Thermal printer

Features	Laser Printers	Inkjet Printers	Thermal Printers	Dot Matrix Printers
Printing material used	Ink powder	Liquid ink	Heat sensitive paper	Ink soaked ribbon
How it prints?	It fuses the powder on the paper through heating.	It sprays liquid ink on paper through microscopic nozzles.	Thermal paper is passed over the thermal print head.	Pins are pushed against ribbon on paper.
Printing speed	20 pages per minute	6 pages per minute	150 mm per second	30 - 550 characters per second
Quality	Printing quality is good. Best for black and white.	Printing quality is good, especially for smaller fonts.	Poor quality printing of images. Good quality text printing.	Poor printing quality for images. In terms of text, printing is good.
Advantages	Quiet, prints faster, high print quality.	Quiet, high print quality, no warm up time, device cost is less.	Quiet, fast, smaller, lighter & consume less power. Portable.	Cheaper to print as ribbon is cheap. Carbon copy possible.
Disadvantages	More susceptible to paper jams. Toner is expensive. Device itself is expensive.	Ink is expensive, ink is not waterproof, and nozzle is prone to clogging.	Requires special thermal quality paper. Poor quality printing.	Initial purchase is expensive, maintenance is expensive, printing is not fast, makes noise.

Table 2.3 : Comparison of printers

iii. Plotter

A plotter is an output device used to produce hardcopies of graphs and designs on the paper. A plotter is typically used to print large-format graphs or maps such as construction maps, engineering drawings and big posters. It is used in the design of cars, ships, aircrafts, buildings, highways etc. Plotters are of two types: Drum plotters and Flatbed plotters.

Drum plotter

A drum plotter is also known as roller plotter. It consists of a drum or roller on which a paper is placed and the drum rotates back and forth to produce the graph on the paper. It also consists of a drawing arm that holds a set of coloured ink pens or pencils. The drawing arm moves side to side as the paper is rolled back and forth through the roller. In this way, a perfect graph or map is created on the paper.

Fig.2.48: Drum plotter

Flatbed plotter

A flatbed plotter is also known as table plotter. It plots on paper that is spread and fixed over a rectangular flatbed table. The flatbed plotter uses two drawing arms, each of which holds a set of coloured ink pens or pencils. The drawing arms move over the stationary paper and draw the graph on the paper. Flatbed plotter is very slow in drawing or printing graphs. The large and complicated drawing can take several hours to print.

Fig.2.49: Flatbed plotter

iv. Three dimensional (3D) printer

A 3D printer is a new generation output device used to print 3D objects. It can produce different kinds of objects in different materials, using the same printer. A 3D printer can print anything from ceramic cups to plastic toys, metal machine parts, stoneware vases, fancy chocolate cakes, etc.

The 3D printing process turns the object to be printed into thousands of horizontal tiny little layers. It then prints these layers from the bottom to top, layer by layer. These tiny layers stick together to form a solid object.

Fig.2.50: 3D Printer

v. Audio output device

The audio output is the ability of the computer to produce sound. Speakers are the output device that produces sound. It is connected to the computer through audio ports. The speaker produces sound by the movement of the diaphragm in the speaker, forward and backward according to the electrical signals coming out of the audio port. For high quality sound reproduction computers use 2.1 (3 speakers), 5.1 (5 speakers) and 7.1 (7 speakers) speaker systems.

Fig.2.51: Speakers

2.2 e-Waste

e-Waste refers to electronic products nearing the end of their "useful life". Electronic waste may be defined as discarded computers, office electronic equipment, entertainment devices, mobile phones, television sets and refrigerators. The used electronics which are destined for reuse, resale, salvage, recycling or disposal are also considered as e-Waste.

Nowadays electronics is part of modern life - desktops, laptops, cell phones, refrigerators, TVs and a growing number of other gadgets. Every year we buy new, updated equipments to satisfy our needs. More than 300 million computers and one billion cell phones are produced every year. All of these electronics goods become obsolete or unwanted, often within two or three years of purchase. This global mountain of waste is expected to continue growing at 8% per year.

Rapid changes in technology, changes in media, falling prices and planned obsolescence have resulted in a fast-growing surplus of electronic waste around the globe. It is estimated that 50 million tons of e-Waste are produced each year. Only 15-20% of e-Waste is recycled, the rest of these materials go directly into landfills and incinerators. Sale of electronic products in countries such as India and China and across continents such as Africa and Latin America are set to rise sharply over the next 10 years.

2.2.1 Why should we be concerned about e-Waste?

Electronic waste is not just waste. It contains some toxic substances such as mercury, lead, cadmium, brominated flame retardants, etc. The toxic materials can cause cancer, reproductive disorders and many other health problems, if not properly managed. It has been estimated that e-Waste may be responsible for upto 40% of the lead found in landfills.

Chemical	Source	Consequence
Lead	Found as solder on printed circuit boards and in computer monitor glass.	Lead can cause damage to the central and peripheral nervous systems, blood systems, and kidneys in humans.
Mercury	Found in printed circuit boards, LCD screen backlights.	Affect a baby's growing brain and nervous system. Adults can suffer organ damage, mental impairment and a variety of other symptoms.
Cadmium	Found in chip resistors and semiconductors.	Cause various types of cancer. Cadmium can also accumulate in, and harm the kidneys.
BFRs-Brominated Flame Retardants	Found in printed circuit boards and some plastics.	These toxins may increase the risk of cancer.

Table 2.4 : Hazardous chemicals, its source and consequence

2.2.2 What happens to the e-Waste?

Unfortunately, an incredibly small percentage of e-Waste is recycled. Even when we take it to a recycling center it is often not actually recycled - in the way most of us expect.

CRTs have a relatively high concentration of lead and phosphors both of which are necessary for the display. The United States Environmental Protection Agency (EPA) included discarded CRT monitors in its category of 'hazardous household waste'.

The majority of e-Waste is most often dumped or burned - either in formal landfills and incinerators or informally dumped or burned. These inappropriate disposal methods for electronic waste fail to reclaim valuable materials or manage the toxic materials safely. In effect, our soil, water and air are easily contaminated.

e-Waste should never be disposed with garbage and other household wastes. This should be segregated at the site and sold or donated to various organisations. Considering the severity of the e-Waste problem, it is necessary that certain

Fig. 2.52: Defective and obsolete electronic items

management options be adopted by government, industries and the public to handle the bulk e-Waste.

Realising the growing concern over e-Waste, Central Pollution Control Board (CPCB) of Government of India has formulated "The e-Waste (Management & Handling) Rules, 2011" and are effective from 01-05-2012. These rules shall apply to every producer, consumer, collection centre, dismantler and recycler of e-Waste involved in the manufacture, sale and processing of electrical and electronic equipment or components. The implementation and monitoring of these guidelines shall be done by the State Pollution Control Boards concerned.

Government of Kerala has introduced strict measures for safe collection and disposal of e-Waste through a government order. The government has defined the role of manufacturers, local bodies and the Pollution Control Board (PCB) in safe disposal of e-Waste. Under the Extended Producer Responsibility, manufacturers of electrical and electronic goods will be required to take back used products from consumers directly or through agents or introduce buyback arrangement. They will also have to supply the e-Waste to authorised recycling units. Consumers have been directed to return used products of known brands to the manufacturers or deposit them at the collection centresset up by local bodies.The PCB will be required to identify agencies for recycling or disposal of e-Waste and organise awareness programmes on e-Waste disposal.

2.2.3 e-Waste disposal methods

The following disposal methods can be used for disposing e-Waste.

a. **Reuse:** It refers to second-hand use or usage after the equipment has been upgraded or modified. Most of the old computers are passed on to relatives/ friends or returned to retailers for exchange or for money. Some computers are also passed on to charitable institutions, educational institutions, etc. Inkjet cartridges and laser toners are also used after refilling. This method reduces the volume of e-Waste generation.

b. **Incineration:** It is a controlled and complete combustion process in which the waste is burned in specially designed incinerators at a high temperature in the range of 900 to 1000 degree Celsius.

c. **Recycling of e-Waste:** Recycling is the process of making or manufacturing new products from a product that has originally served its purpose. Monitors, keyboards, laptops, modems, telephone boards, hard drives, compact disks, mobiles, fax machines, printers, CPUs, memory chips, connecting wires and cables can be recycled.

d. **Land filling:** It is one of the most widely used, but not recommended methods for disposal of e-Waste. In this method soil is excavated from the trenches made and waste material is buried in it, which is covered by a thick layer of soil.

2.2.4 Students' role in e-Waste disposal

• Stop buying unnecessary electronic equipments.

• When electronic equipments get faulty try to repair it instead of buying a new one.

• Try to recycle electronic equipments by selling them or donating them to others extending their useful life and keeping them out of the waste stream.

• If you really need to buy new electronics, choose items with less hazardous substances, greater recycled content, higher energy efficiency, longer life span, and those that will produce less waste.

• Visit the manufacturer's website or call the dealer to find out if they have a take back programme or scheme for your discarded electronics.

• If the device is battery-operated, buy rechargeable instead of disposable batteries.

• Buy products with good warranty and take back policies.

2.3 Green computing or Green IT

Green computing is the study and practice of environmentally sustainable computing or IT. Green computing is the designing, manufacturing, using and disposing of computers and associated components such as monitors, printers, storage devices, etc., efficiently and effectively with minimal or no impact on the environment.

One of the earliest initiatives towards green computing was the voluntary labelling program known as 'Energy Star'. It was conceived by the Environmental Protection Agency (EPA) in 1992 to promote energy efficiency in hardware of all kinds. The Energy Star label has become a common sight, especially in notebook computers and displays. Similar programmes have been

Fig.2.53: Energy Star label

adopted in Europe and Asia. The commonly accepted Energy Star symbol is shown in Figure 2.53.

Government regulation is only a part of an overall green computing idea. The work habits of computer users and businesses have to be modified to minimise adverse impact on the global environment. Here are some steps that can be taken:

• Turn off computer when not in use.

• Power-on the peripherals such as laser printers only when needed

• Use power saver mode.

• Use laptop computers rather than desktop computers whenever possible.

• Take printouts only if necessary.

- Use Liquid Crystal Display (LCD) monitors rather than Cathode Ray Tube (CRT) monitors.
- Use hardware/software with Energy Star label.
- Dispose e-Waste according to central, state and local regulations.
- Employ alternative energy sources like solar energy.

The environmentally responsible and eco-friendly use of computers and their resources is known as green computing.

How to make computers green?

The features that are important in making a computer greener include size, efficiency and materials. Smaller computers are greener because they use fewer materials and require less electricity to run. Efficient use of energy is also an important component of a green computer. Smaller computers such as laptops are more energy-efficient than bigger models and LCD screens use much less energy than the older CRT models. The use of hazardous materials such as lead and mercury should be minimised.

To promote green computing the following four complementary approaches are employed:

Green design: Designing energy-efficient and eco-friendly computers, servers, printers, projectors and other digital devices.

Green manufacturing: Minimising waste during the manufacturing of computers and other components to reduce the environmental impact of these activities.

Green use: Minimising the electricity consumption of computers and peripheral devices and using them in an eco-friendly manner.

Green disposal: Reconstructing used computers or appropriately disposing off or recycling unwanted electronic equipment.

Check yourself

1. The environmentally responsible and eco-friendly use of computers and their resources is known as _____ .
2. The process of making or manufacturing new products from the product that has originally served its purpose is called _____.
3. Compare dot matrix printers and laser printers.
4. List any two input and output devices each.

2.4 Software

Software is a general term used to denote a set of programs that help us to use computer system and other electronic devices efficiently and effectively. If hardware is said to form the body of a computer system, software is its mind.

There are two types of software:
- System software
- Application software

2.4.1 System software

It is a set of one or more programs designed to control the operations of a computer. They are general programs designed to assist humans in the use of computer system by performing tasks such as controlling the operations, move data into and out of a computer system and to do all the steps in executing application programs. In short, system software supports the running of other software, its communication with other peripheral devices. It helps the users to use computer in an effective manner. It implies that system software helps to manage resources of the computer. Figure 2.53 depicts how system software interfaces with user and hardware.
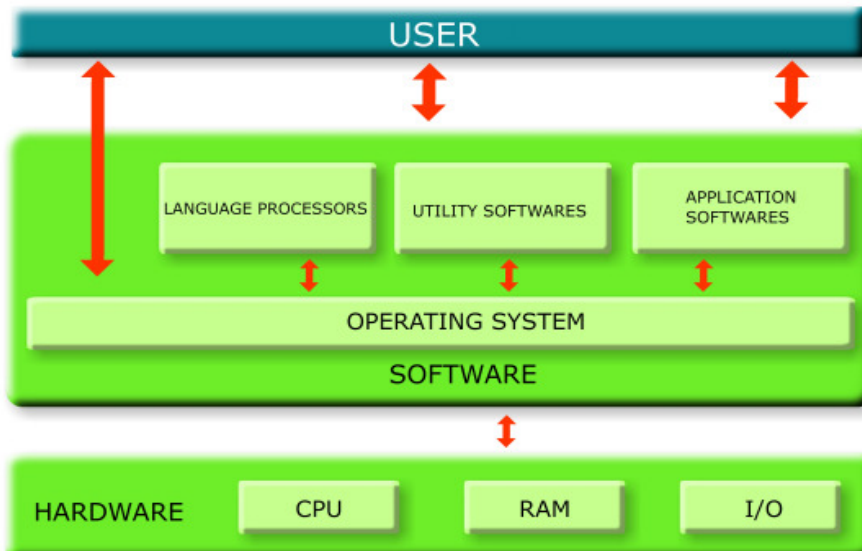
USER

LANGUAGE PROCESSORS   UTILITY SOFTWARES   APPLICATION SOFTWARES

OPERATING SYSTEM

SOFTWARE

HARDWARE   CPU   RAM   I/O

Fig. 2.54: Software with user and hardware interface

System software is a set of system programs which aids in the execution of a general user's computational requirements on a computer system. The following are the components of system software.

    a. Operating system

    b. Language processors

    c. Utility software

a. Operating system

Operating system is a set of programs that acts as an interface between the user and computer hardware. The primary objective of an operating system is to make the computer system convenient to use. Operating system provides an environment for user to execute programs. It also helps to use the computer hardware in an efficient manner.

Operating system controls and co-ordinates the operations of a computer. It acts as the resource manager of the computer system. Operating system is the most important system software. It is the first program to be loaded from hard disk in the computer and it resides in the memory till the system is shut down. It tries to prevent errors and the improper use of computer.

Major functions of an operating system

The major functions of an Operating System are process management, memory management, file management, security management and command interpretation.
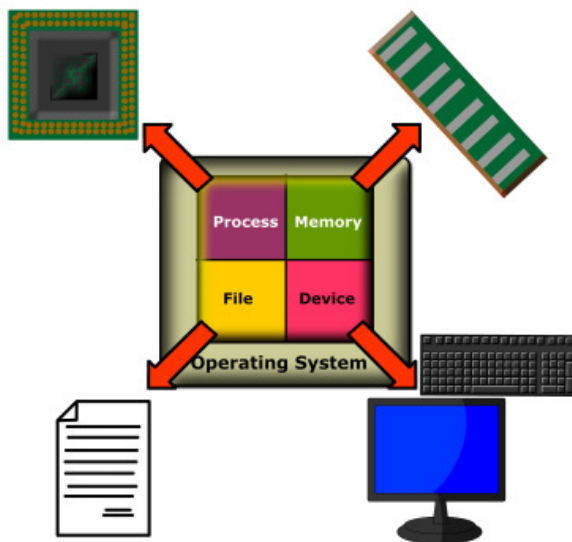

Fig. 2.55: Operating System as a resource manager

i. Process management

By the term process we mean a program in execution. The process management module of an operating system takes care of the allocation and de-allocation of processes and scheduling of various system resources to the different requesting processes.

ii. Memory management

Memory management is the functionality of an operating system which handles or manages primary memory. It keeps track of each and every memory location to ensure whether it is allocated to some process or free. It calculates how much memory is to be allocated to each process and allocates it. It de-allocates memory if it is not needed further.

iii. File management

The file management module of an operating system takes care of file related activities such as organising, naming, storing, retrieving, sharing, protection and recovery.

iv. Device management

Device management module of an operating system performs the management of devices attached to the computer. It handles the devices by combining both hardware and software techniques. The OS communicates with the hardware device via the device driver software.

Examples of various operating systems are DOS, Windows, Unix, Linux, Mac OS X, etc.

b. Language processors

We know that natural languages are the medium of communication among human beings. Similarly, in order to communicate with the computer, the user also needs to have a language that should be understood by the computer. Computer languages may be broadly classified into low level languages and high level languages.

Low-level languages are described as machine-oriented languages. In these languages, programs are written using the memory and registers available on the computer. Since the architecture of computer differs from one machine to another, there is separate low level programming language for each type of computer. Machine language and assembly language are the different low level languages.

Machine language: We know that a computer can understand only special signals, which are represented by 1s and 0s. These two digits are called binary digits. The language, which uses binary digits, is called machine language. Writing a program in machine language is definitely very difficult. It is not possible to memorise a long string of 0s and 1s for every instruction.

Assembly language: Assembly language is an intermediate-level programming language. Assembly languages use mnemonics. Mnemonic is a symbolic name given to an operation. For example ADD for addition operation, SUB for subtraction operation, etc. It is easier to write computer programs in assembly language as compared to machine language. It is machine dependent and programmer requires knowledge of computer architecture.

High Level Languages (HLL): These are like English languages and are simpler to understand than the assembly language or machine language. High level language is not understandable to the computer. A computer program written in a high level language is to be converted into its equivalent machine language program. So these languages require a language translator (compilers or interpreters) for conversion. Examples of high level programming languages are BASIC, C, C++, Java, etc.

Need for language processor

The programs consisting of instructions to the computer, written in assembly language or high level language are not understood by the computer. We need language processors to convert such programs into low level language, as computer can only understand machine language. Language processors are the system programs that translate programs written in high level language or assembly language into its equivalent machine language.

Types of language processors

- **Assembler:** Assembly languages require a translator known as assembler for translating the program code written in assembly language to machine language. Because computer can interpret only the machine code instruction, the program can be executed only after translating. Assembler is highly machine dependent.

- **Interpreter:** Interpreter is another kind of language processor that converts a HLL program into machine language line by line. If there is an error in one line, it reports and the execution of the program is terminated. It will continue the translation only after correcting the error. BASIC is an interpreted language.

- **Compiler:** Compiler is also a language processor that translates a program written in high level language into machine language. It scans the entire program in a single run. If there is any error in the program, the compiler provides a list of error messages along with the line number at the end of the compilation. If there are no syntax errors, the compiler will generate an object file. Translation using compiler is called compilation. After translation compilers are not required in memory to run the program. The programming languages that have a compiler are C, C++, Pascal, etc.

Figure 2.56 shows process involved in the translation of assembly language and high level language programs into machine language programs.
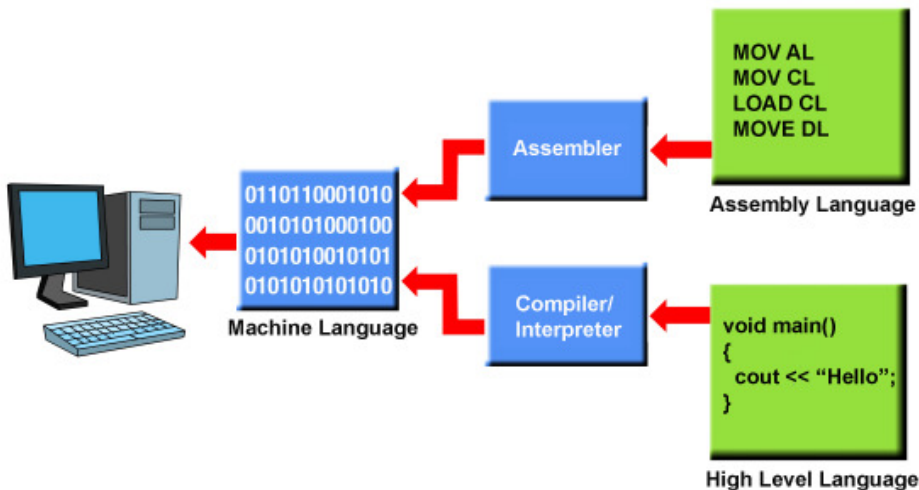

Fig. 2.56: Language processing

c. Utility software

Utility software is a set of programs which help users in system maintenance tasks and in performing tasks of routine nature. Some of the utility programs with their functions are listed below:

- **Compression tools:** Large files can be compressed so that they take less storage area. These compressed files can be decompressed into its original form when needed. Compression of files is known as zipping and decompression is called unzipping. WinZip, WinRAR, etc. are examples of compression tools.

- **Disk defragmenter:** Disk defragmenter is a program that rearranges files on a computer hard disk. The files are arranged in such a way that they are no longer fragmented. This enables the computer to work faster and more efficiently.

- **Backup software:** Backup means duplicating the disk information so that in an event of disk failure or in an event of accidental deletion, this backup may be used. Backup utility programs facilitates the backing up of disk.

- **Antivirus software:** A computer virus is a program that causes abnormality in the functioning of a computer. Antivirus software is a utility program that scans the computer system for viruses and removes them. As new viruses are released frequently, we have to make sure that latest antivirus versions are installed on the computer. Most of the antivirus programs provide an auto-update feature which enables the user to download profiles of new viruses so as to identify and inactivate them. Norton Antivirus, Kaspersky, etc. are examples of antivirus programs.

2.4.2 Application software

Software developed for specific application is called application software. It includes general purpose software packages and specific purpose software. GIMP, Payroll System, Airline Reservation System, Tally, etc. are examples of application software.

a. General purpose software packages

General purpose software are used to perform operations in a particular application area. Such software is developed keeping in mind the various requirements of its users. They provide a vast number of features for its users. General purpose software is classified as word processors, spreadsheet software, presentation software, database software and multimedia software.

- **Word processing software:** Word Processing software is designed for creating and modifying documents. It helps to create, edit, format and print textual matters easily. Formatting features include different font settings, paragraph settings, bullets and numbering, alignments and more. In addition to this it can check spelling and grammar in the document, insertion of pictures, charts and tables. We can specify headers and footers for every page in the document. The most popular examples of this type of software are MS Word, Open Office Writer, etc.

- **Spreadsheet software:** Spreadsheet software allows users to perform calculations using spreadsheets. They simulate paper worksheets by displaying multiple cells that make up a grid. It also allows us to insert drawing objects in the worksheet and create different types of charts for graphical representation of numerical data. Microsoft Excel, Open Office Calc, Lotus 1-2-3 and Apple Numbers are some examples of spreadsheet software.

- **Presentation software:** The software that is used to display information in the form of a slide show is known as presentation software. Presentation software allows preparing slides containing pictures, text, animation, video and sound effects. Microsoft PowerPoint and Open Office Impress are examples for presentation software.

- **Database software:** Database is an organised collection of data arranged in tabular form. Database Management System (DBMS) consists of a collection of interrelated data and a set of programs to access those data. The primary goal of a DBMS is to provide an environment that is both convenient and efficient to use in retrieving and storing database information. They provide privacy and security to data and enforce standards for data. Examples of DBMS software are Microsoft Access, Oracle, PostgreSQL, My SQL, etc.

- **Multimedia software:** Multimedia is the integration of multiple forms of media. This includes text, graphics, audio, video, etc. Multimedia software can process information in a number of media formats. It is capable of playing media files. Some multimedia software allows users to create and edit audio and video files. Audio converters, audio players, and video editing software are some forms of multimedia software. Examples are VLC Player, Adobe Flash, Real Player, Media Player, etc.

b. Specific purpose software

Specific purpose software is a highly specialised software designed to handle particular tasks. These are tailor-made software to satisfy the needs of an organisation or institution. It is also known as customised software. Since customised software is developed for a single customer, it can accommodate that customer's particular preferences and expectations. Some examples of specific purpose application software are listed in Table 2.5.

Application Software	Purpose
• Payroll System	• Payroll system maintains the details of employees of an organisation and keeps track of their salary details.
• Inventory Management System	• It is used for tracking inventory levels, orders, sales and deliveries in a business firm.
• Human Resource Management System	• It is used for managing human resource in an organisation.

Table 2.5 : Examples of Application Software

Check yourself

1. Define operating system.
2. Give two examples for OS.
3. A program in execution is called _____.
4. Mention any two functions of OS
5. Name the software that translates assembly language program into machine language program.
6. Differentiate between compiler and interpreter.
7. DBMS stands for _____.
8. Give two examples for customized software.
9. Duplicating disk information is called _____.

2.5 Free and open source software

Free and open source software gives the user the freedom to use, copy, distribute, examine, change and improve the software. Nowadays free and open source software is widely used throughout the world because of adaptable functionality, less overall costs, vendor independency, adherence to open standards, interoperability and security.

The Free Software Foundation (FSF) defines the four freedoms for free and open source software:

Freedom 0 - The freedom to run program for any purpose.

Freedom 1 - The freedom to study how the program works and adapt it to your needs. Access to source code should be provided.

Freedom 2 - The freedom to distribute copies of the software.

Freedom 3 - The freedom to improve the program and release your improvements to the public, so that the whole community benefits.

The following are some of the examples of free and open source software:

GNU/Linux: GNU/Linux is a computer operating system assembled under the model of free and open source software development and distribution. It was organised in the GNU project introduced in 1983 by Richard Stallman in the FSF.

GIMP: It stands for GNU Image Manipulation Program. It is an image editing software. It can be used for retouching photographs, creating and editing images. It supports graphic files of different formats and allows converting from one format to another.

Mozilla Firefox: It is one of the most popular web browsers created by the Mozilla Corporation. It provides added security features for safe browsing.

OpenOffice.org: It is a complete office suite that contains word processor (Writer) to prepare and format documents, spreadsheets (Calc) and presentations (Impress). It works on both Linux and Windows platforms.

2.6 Freeware and shareware

Freeware refers to copyrighted computer software which is made available for use free of charge for an unlimited period.

The term shareware refers to commercial software that is distributed on a trial basis. It is distributed without payment and with limited functionality. Shareware is commonly offered in a downloadable format on the Internet. The distribution of this kind of software aims at giving the users a chance to analyse the software before purchasing it. Some shareware works for a limited period of time only.

Table 2.6 highlights a comparison between freeware and shareware:

Freeware	Shareware
Freeware refers to software that anyone can download from the Internet and use for free.	Sharewares give users a chance to try the software before buying it.
All the features are free.	All features are not available. To use all the features of the software, user has to purchase it.
Freeware programs can be distributed free of cost.	Shareware may or may not be distributed freely. In many cases, author's permission is needed to distribute the shareware.

Table 2.6 : Comparison of Freeware and Shareware

Let us do

1.   *Conduct a debate on free software and proprietary software.*

2.   *Discuss four freedoms which make up a free software.*

2.7 Proprietary software

Proprietary software is a computer program that is an exclusive property of its developer or publisher and cannot be copied or distributed without licensing agreements. It is sold without any access to source code and is therefore not possible to change or improve by the user. Some examples of proprietary software are Microsoft Windows Operating System, MS Office, Mac OS, etc.

2.8 Humanware or Liveware

Humanware or liveware refers to humans who use computer. It was used in computer industry as early as 1966 to refer to computer users, often in humorous contexts by analogy with software and hardware. It refers to programmers, systems analysts, operating staff and other personnel working in a computer system (refer Table 2.7).

Humanware	Job Description
System Administrators	Upkeep, configuration and reliable operation of computer systems; especially multi-user computers such as servers.
Systems Managers	Ensure optimal level of customer services and maintain expertise in all business unit systems and develop professional relationships with all vendors and contractors.
System Analysts	Design new IT solutions to improve business efficiency and productivity.
Database Administrators	Create, monitor, analyse and implement database solutions.
Computer Engineers	Design either the hardware or software of a computer system.
Computer Programmers	Write the code that computers read in order to operate properly.
Computer Operators	Oversee the running of computer systems, ensuring that the machines are running, physically secured and free of any bugs.

Table 2.7 : Examples of Humanware

Check yourself

1. An example of free and open source software is _____.
2. The software that give users a chance to try it before buying is _____.
3. What do you mean by free and open source software?
4. An example of proprietary software is _____.
5. Give two examples of humanware.

Let us sum up

A computer system consists of hardware and software. Hardware refers to the visible and tangible parts of computer. Processor, motherboard, peripheral and ports, memory devices, input/output devices, etc. belong to hardware components of computer. Memory is classified into primary and secondary memories. Primary storage consists of RAM, ROM and cache. Seciondary storage devices are further classified into magnetic storage, optical storage and semi conductor storage devices. There are variety of input devices which include keyboard, mouse, light pen, touch screen, graphic tablet, touchpad, joystick, microphone, scanner, OMR, barcode reader, biometric sensor, smart card reader, digital camera, etc. Similarly, there are output devices like visual display units, printers, plotters, audio output devices, etc. The used electronics which are destined for reuse, resale, salvage, recycling or disposal are considered as e-Waste. There are different methods for e-Waste disposal which include re-use, incineration, recycling, land filling, etc. Green computing is the study and practice of environmentally sustainable computing or IT.

Software refers to the intangible or invisible part of the computer which is a set of programs that help us to use computer system and other electronic devices efficiently and effectively. Software is classified into system software and application software. System software include operating system, language processors and utility software. Application software is further divided into general purpose software and specific purpose software. Free and open source software are available which give freedom to use, copy, distribute, examine, change and improve the software. The term freeware refers to software that can be downloaded from the Internet and use for free. Shareware give users a chance to try the software before buying it. There is one more component associated with computer, the humanware. It refers to humans who use computer at various level.

Learning outcomes

After the completion of this chapter the learner will be able to

• explain the basic organisation of computer system.

• list the use of different types of input and output devices.

• distinguish between system software and application software.

• identify the importance of e-Waste disposal.

• identify the importance of green computing concept.

• classify the different types of software.

• list the functions of operating system.

• use word processor, electronic spreadsheets and presentation software.

• classify the different types of computer languages.

• list the use of different types of utility software.

• promote open source software.

• explain the term humanware or liveware.

Sample questions

Very short answer type

1. What are the components of a digital computer?

2. Write the main functions of central processing unit.

3. What are the different types of main memory?

4. What is the advantage of EEPROM over EPROM?

5. When do we use ROM?

6. What is an input device? List few commonly used input devices.

7. What do you mean by an output device? List few commonly used output devices.

8. What is a storage device? List few commonly used storage devices.

9. What is the role of ALU?

10. What is a control unit?

11.   What are registers? Write and explain any two of them.

12.   Differentiate Hard copy and Soft copy.

13.   What is e-Waste?

14.   What is operating system?

15.   What is a language processor?

16.   Mention the categories of computer languages.

17.   What is disk defragmenter?

18.   What is proprietary software?

19.   What do you mean by open source software?

Short answer type

1.   Briefly explain any three input devices.

2.   Compare CRT with LED monitor.

3.   Differentiate RAM and ROM.

4.   List and explain e-Waste disposal methods.

5.   Enumerate the steps that can be taken for the implementation of green computing philosophy.

6.   What do you mean by customized software? Give examples.

7.   Distinguish between low level and high level languages.

8.   Differentiate compiler and interpreter.

9.   Describe the use of electronic spreadsheets.

10.   What is utility software? Give two examples.

11.   Categorise the software given below into operating system, application packages and utility programs.

      Linux, Tally, WinZip, MS-Word, Windows, MS-Excel

12.   Differentiate between freeware and shareware.

13.   What are the four freedoms which make up free and open source software?

14.   What do you mean by humanware? Give any two examples.

Long answer type

1. Describe in detail the various units of the Central Processing Unit.

2. Briefly explain various types of memory.

3. Explain the classification of printers.

4. "e-Waste is hazardous to our health and environment." Justify the statement. List and explain the methods commonly used for e-Waste disposal.

5. Define the term green computing. List and explain the approaches that you can adopt to promote green computing concepts at all possible levels.

6. List and explain various categories of software.

7. Describe the use of various utility softwares.

8. Define the term 'operating system'. List and explain the major functions of operating system.

9. List and explain general purpose application software with examples.

3

Principles of Programming and Problem Solving

We have learnt the concept of data processing and the role of computers in data processing. We have also discussed the computer as a system with components such as hardware, software and users. We had a detailed discussion on these components in the previous chapter. Let us recall the definition of software. In its simplest form, we can say that software means a collection of programs to solve problems using computers. As we know, a computer cannot do anything on its own. It must be instructed to perform the desired job. Hence it is necessary to specify a sequence of instructions that the computer must perform to solve a problem. Such a sequence of instructions written in a language that is understood by a computer is called a "computer program". Writing a computer program is a challenging task. However, we can attempt it by procuring the concepts of problem solving techniques and different stages of programming.

3.1 Problem solving using computers

A computer can solve problems only when we give instructions to it. If it understands the tasks contained in the instructions, it will work accordingly. An instruction is an action oriented

statement. It tells the computer what operation it should perform. A computer can execute (carry out the task contained in) an instruction only if the task is specified precisely and accurately. As we learnt in the previous chapter, there are programmers who develop computer programs for solving problems. Once the program is developed and stored permanently in a computer, we can ask the computer to execute it as and when required.

We should be cautious about the clarity of the logic of the solution and the format of instructions while designing a program, because computer does not possess common sense or intuition. As human beings we use judgments based on experience, often on subjective and emotional considerations. Such value oriented judgments often depend on what is called "common sense". As opposed to this, a computer exhibits no emotion and has no common sense. That is why we say that computer has no intelligence of its own.

In a way, computer may be viewed as an 'obedient servant'. Being obedient without exercising 'common sense' can be very annoying and unproductive. Take the instance of a master who sent his obedient servant to a post office with the instruction "Go to the post office and buy ten 5 rupees stamps". The servant goes to the post office with the money and does not return even after a long time. The master gets worried and goes in search of him to the post office and found the servant standing there with the stamps in his hand. When the angry master asks the servant for an explanation, the servant replies that he was ordered to buy ten 5 rupees stamps but not to return with them!

3.2 Approaches in problem solving

A problem may be solved in different ways. Even the approach may be different. In our life, we may seek medical treatment for some diseases. We can consult an allopathic, ayurvedic or homoepathic doctor. Each of their approaches may be different, though all of them are solving the same problem. Similarly in problem solving also different approaches are followed. Let us discuss the two popular designing styles of problem solving – Top down design and Bottom up design.

3.2.1 Top down design

Look at Figure 3.1. If you are asked to draw this picture, how will you proceed? It may be as follows:

1.   Draw the outline of the house.

2.   Draw the chimney
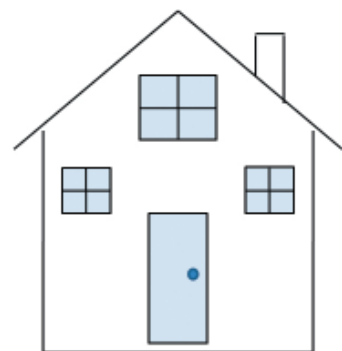
3.   Draw the door

4.   Draw the windows

Fig. 3.1 : Lay-out of a House

While drawing the door in Step 3, the procedure may be as follows: 3.1 Outline of the door 3.2 Shading 3.3 Handle	Similarly the windows may be drawn as follows: 4.1 Outline of the window 4.2 Shading 4.3 Horizontal and Vertical lines

The procedure described above may be summarised as follows:

The whole problem (here drawing the picture) is broken down into smaller tasks. Thus four tasks are identified to solve the problem. Some of these tasks (here steps 3 and 4 for drawing the door and windows) are further subdivided. Thus any complex problem can be solved by breaking it down into different tasks and solving each task by performing simpler activities. This concept is known as **top down design** in problem solving.
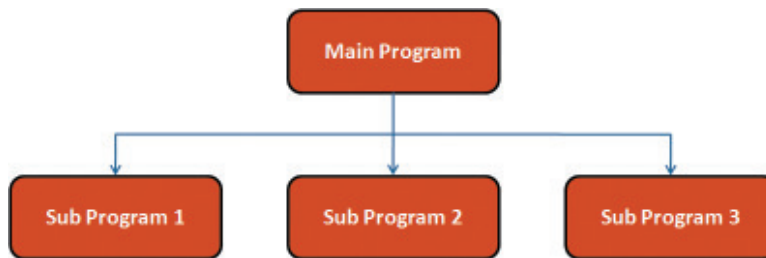

Fig. 3.2 : Decomposition of a problem

It is one of the programming approaches that has been proven the most productive. As shown in Figure 3.2, top down design is the process of breaking the overall procedure or task into component parts (modules) and then subdividing each component module until the lowest level of detail is reached. It is also called top down decomposition since we start "at the top" with a general problem and design specific solutions to its sub problems. In order to obtain an effective solution for the main problem, it is desirable that the sub problems (sub programs) should be independent of each other. So, each sub problem can be solved and tested independently.

The following are the advantages of problem solving by decomposition:

• Breaking the problem into parts helps us to clarify what is to be done in each part.

• At each step of refinement, the new parts become less complicated and therefore, easier to figure out.

• Parts of the solution may turn out to be reusable.

• Breaking the problem into parts allows more than one person to work for the solution.

3.2.2 Bottom up design

Consider the case of constructing a house. We do not follow the top down design, but the bottom up design. The foundation will be the first task and roofing will be the last task. Breaking down of tasks is carried out here too. It is true that some tasks can be carried out only after the completion of some other tasks. However roofing which is the main task will be carried out only after the completion of bottom level tasks.

Similarly in programming, once the overall procedure or task is broken down into component parts (modules) and each component module is further sub divided until the lowest level of detail has been reached, we start solving from the lowest module onwards. The solution for the main module will be developed only after designing specific solutions to its sub modules. This style of approach is known as **bottom up design** for problem solving. Here again, it is desirable that the sub problems (subprograms) should be independent of each other.

> *Youth festivals are conducted every year in our schools. Usually the duties and responsibilities are decomposed. Discuss how the tasks are divided and executed to organise the youth festival*
>
> **Let us do** *successfully.*

3.3 Phases in programming

As we have seen, problem solving using computer is a challenging task. A systematic approach is essential for this. The programs required can be developed only by going through different stages. Though we have in-born problem solving skills, it can be applied effectively only by proper thinking, planning and developing the logical reasoning to solve the problem. We can achieve this by proceeding through the following stages, in succession:

Fig. 3.3 : Phases of Programming
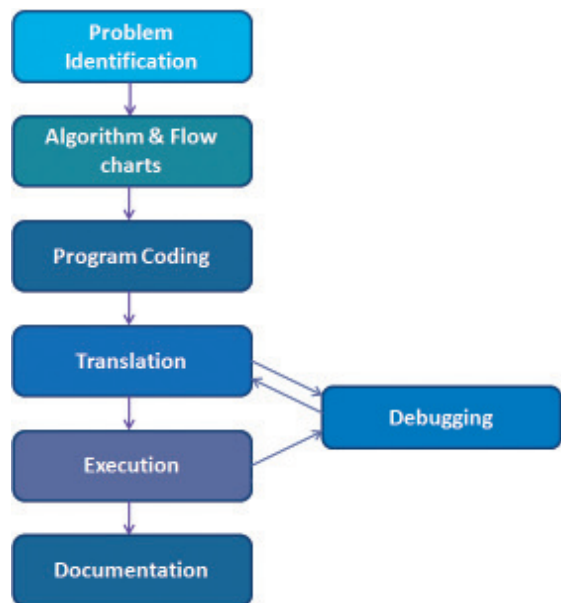
1. Problem identification

2. Preparing algorithms and flow-charts

3.   Coding the program using programming language

4.   Translation

5.   Debugging

6.   Execution and testing

7.   Documentation

Figure 3.3 shows the order of performing the tasks in various stages of programming. Note that the debugging stage is associated with both translation and execution. The activities involved in the seven stages mentioned above are detailed in the following sections.

3.3.1 Problem identification

Let us discuss a real life situation; suppose you are suffering from a stomach ache. As you know this problem can be solved by a doctor. Your doctor may ask you some questions regarding the duration of pain, previous occurrence, your diet etc., and examine some parts of your body using the stethoscope, X-ray or scan. All these are a part of the problem study. After these procedures, your doctor may be able to identify the problem and state it using some medical term. Now the second stage begins with the derivation of some steps for solution known as prescriptions.

It is clear that before deriving the steps for solution, the problem must be analysed. During this phase you will be able to identify the data involved in processing, its type and quantity, formula to be used, activities involved, and the output to be obtained. Once we studied the problem clearly, and are convinced about the sequence of tasks required for the solution, we can go to the next phase. This is the challenging phase as it exploits the efficiency of the programmer (problem solver).

3.3.2 Algorithms and Flowcharts

Once the problem is identified, it is necessary to develop a precise step-by-step procedure to solve the problem. This procedure is not new or confined to computers. It has been in use for a very long time, and in almost all walks of life. One such procedure taken from real life is described below. It is a cooking recipe of an omlette taken from a magazine.

Ingredients

Eggs - 2 Nos, Onion - 1 (small sized, chopped); Green chili - 2 (finely chopped); Oil - 2 tea spoon, Salt - a pinch.

Method

Step 1 : Break the eggs and pour the contents in a vessel and stir.

Step 2 : Mix chopped onion, green chilies and salt with the stirred egg.

Step 3 : Place a pan on the stove and light the stove.

Step 4 : Pour the oil in the pan and wait till it gets heated.

Step 5 : Pour the mixture prepared in step 2 into the pan and wait till the side is fried.

Step 6 : Turn over to get the other side fried well.

Step 7 : Take it out after some seconds.

Result

An omlette is ready to be served with pepper powder.

The recipe given above has the following properties:

1.   It begins with a list of ingredients required for making the omlette. These may be called the inputs.

2.   A sequence of instructions is given to process the inputs.

3.   As a result of carrying out the instructions, some outputs (here, omlette) are obtained.

The instructions given to process the inputs are, however, not precise. They are ambiguous. For example, the interpretation of "till the side is fried" in step 5 and "fried well" in step 6 can vary from person to person. Due to such imprecise instructions, different persons following the same recipe with the same inputs can produce omlettes which differ in size, shape and taste.

The above ambiguities should be avoided while writing steps to solve the problems using the computer.

a. Algorithm

Mathematicians trace the origin of the word algorithm to the name of a famous Arab mathematician, Abu Jafar Mohammed Ibn Musaa Al-Khowarizmi. The word 'algorithm' is derived from the last part of his name Al-Khowarizmi. In computer terminology an **algorithm** may be defined as a finite sequence of instructions to solve a problem. It is a step-by-step procedure to solve a

Fig.3.4 : Abu Jafar Mohammed Ibn Musaa Al-Khowarizmi (780 - 850)

problem, where each step represents a specific task to be carried out. However, in order to qualify as an algorithm, a sequence of instructions must possess the following characteristics:

(i) It should begin with instruction(s) to accept inputs. These inputs are processed by subsequent instructions. Sometimes, the data to be processed will be given along with the problem. In such situations, there will be no input instruction at the beginning of the algorithm.

(ii) Use variables to refer the data, where variables are user-defined words consisting of alphabets and numerals that are similar to those used in mathematics. Variables must be used for inputting data and assigning values or results.

(iii) Each and every instruction should be precise and unambiguous. In other words, the instructions must not be vague. It must also be possible to carry them out. For example, the instruction "Catch the day" is precise, but cannot be carried out.

(iv) Each instruction must be sufficiently basic such that it can, in principle, be carried out in finite time by a person with paper and pencil.

(v) The total time to carry out all the steps in the algorithm must be finite. As algorithm may contain instructions to repetitively carry out a group of instructions, this requirement implies that the number of repetitions must be finite.

(vi) After performing the instructions given in the algorithm, the desired results (outputs) must be obtained.

To gain insight into algorithms, let us consider a simple example. We have to find the sum and average of any three given numbers. Let us write the procedure for solving this problem. It is given below:

> Step 1:   Input three numbers.
> Step 2:   Add these numbers to get the sum
> Step 3:   Divide the sum by 3 to get the average
> Step 4:   Print sum and average

Though the procedure is correct, while preparing an algorithm, we have to follow any of the standard formats. Let us see how the procedure listed above can be written in an algorithm style.

Example 3.1: Algorithm to find the sum and average of three numbers

Let A, B, C be variables for the input numbers; and S, Avg be variables for sum and average.

Step 1:   Start

Step 2:   Input A, B, C

Step 3:   S = A + B + C

Step 4:   Avg = S / 3

Step 5:   Print S, Avg

Step 6:   Stop

The above set of instructions qualifies as an algorithm for the following reasons:

- It has input (The variables A, B and C are used to hold the input data).

- The processing steps are precisely specified (Using proper operators in Steps 3 and 4) and can be carried out by a person using pen and paper.

- Each instruction is basic (Input, Print, Add, Divide) and meaningful.

- It produces two outputs such as sum (S) and average (Avg).

- The beginning and termination points are specified using Start and Stop.

Types of instructions

As we know, a computer can perform only limited types of operations. So we can use only that many instructions to solve problems. Before developing more algorithms, let us identify the types of instructions constituting the algorithm.

- Computer can accept data that we input. So, we can use input instructions. The words **Input**, **Accept** or **Read** may be used for this purpose.

- Computer gives the results as output. So we can use output instructions. The words **Print**, **Display** or **Write** may be used for this purpose.

- Data can directly be stored in a memory location or data may be copied from one location to another. Similarly, results of arithmetic operations on data can also be stored in memory locations. We use assignment (or storing) instruction for this, similar to that used in mathematics. Variables followed by the equal symbol (=) can be used for storing value, where variables refer to memory locations.

- A computer can compare data values (known as logical operation) and make decisions based on the result. Usually, the decision will be in the form of selecting or skipping a set of one or more statements or executing a set of instructions repeatedly.

b. Flowcharts

An idea expressed in picture or diagram is preferred to text form by people. In certain situations, algorithm may be difficult to follow, as it may contain complex tasks and repetitions of steps. Hence, it will be better if it could be expressed in pictorial form. The pictorial representation of an algorithm with specific symbols for instructions and arrows showing the sequence of operations is known as **flowchart**. It is primarily used as an aid in formulating and understanding algorithms. Flowcharts commonly use some basic geometric shapes to denote different types of instructions. The actual instructions are written within these boxes using clear and concise statements. These boxes are connected by solid lines with arrow marks to indicate the flow of operation; that is, the exact sequence in which the instructions are to be executed.

Normally, an algorithm is converted into a flowchart and then the instructions are expressed in some programming language. The main advantage of this two-step approach in program writing is that while drawing a flowchart one is not concerned with the details of the elements of programming language. Hence he/she can fully concentrate on the logic (step-by-step method) of the procedure. Moreover, since a flowchart shows the flow of operations in pictorial form, any error in the logic of the procedure can be detected more easily than in a program. The algorithm and flow chart are always a reference to the programmer. Once these are ready and found correct as far as the logic of the solution is concerned, the programmer can concentrate on coding the operations following the constructs of programming language. This will normally ensure an error-free program.

Flowchart symbols

The communication of program logic through flowcharts is made easier through the use of symbols that have standardised meanings. We will only discuss a few symbols that are needed to indicate the necessary operations. These symbols are standardized by the American National Standards Institute (ANSI).

1. Terminal

As the name implies, it is used to indicate the beginning (START) and ending (STOP) in the program logic flow. It is the first symbol and the last symbol in the flowchart.

It has the shape of an ellipse. When it is used as START, an exit flow will be attached. But when it is used as a STOP symbol, an entry flow will be attached.

2. Input / Output

The parallelogram is used as the input/output symbol. It denotes the function of an input/output device in the program. All the input/output instructions are expressed using this symbol. It will be attached with one entry flow and one exit flow.

3. Process

A rectangle is used to represent the processing step. Arithmetic operations such as addition, subtraction, multiplication, division as well as assigning a value to a variable are expressed using this symbol. Assigning a value to a variable means moving data from one memory location to another (e.g. a=b) or moving the result from ALU to memory location (e.g. a=b+5) or even storing a value to a memory location (e.g. a=2). Process symbol also has one entry flow and one exit flow.

4. Decision

The rhombus is used as decision symbol and is used to indicate a point at which a decision has to be made. A branch to one of two or more alternative points is possible here. All the possible exit paths will be specified, but only one of these will be selected based on the result of the decision. Usually this symbol has one entry flow and two exit flows - one towards the action based on the truth of the condition and the other towards the alternative action.
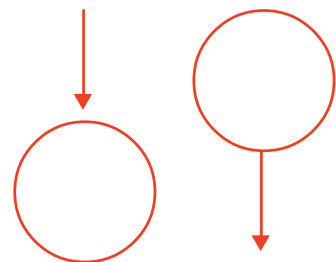
5. Flow lines

Flow lines with arrow heads are used to indicate the flow of operation, that is, the exact sequence in which the instructions are to be executed. The normal flow is from top to bottom and left to right. But in some cases, it can be from right to left and bottom to top. Good practice also suggests that flow lines should not cross each other and such intersections should be avoided wherever possible.

6. Connector

When a flowchart becomes bigger, the flow lines start criss-crossing at many places causing confusion and reducing comprehension of the flowchart. Moreover, when the flowchart becomes too long to fit into a single page the use of flow lines becomes impossible. Whenever a flowchart

becomes complex and the number and direction of flow lines is confusing or it spreads over more than one page, a pair of connector symbols can be used to join the flow lines that are broken. This symbol represents an "entry from", or an "exit to" another part of the flowchart. A connector symbol is represented by a circle and a letter or digit is placed within the circle to indicate the link. A pair of identically labelled connector symbols is commonly used to indicate a continuous flow. So two connectors with identical labels serve the same function as a long flow line. In a pair of identically labelled connectors, one shows an exit to some other chart section and the other indicates an entry from another part of the chart.

Figure 3.5 shows that flowchart of the problem discussed in Example 3.1.

Fig. 3.5 : Flow chart for Sum and Average

The instruction given in each step in the algorithm is represented using the concerned symbol. Each symbol is labelled properly with the respective instruction. The flow of operations is clearly shown using the flow lines.

Advantages of flowcharts

Flowcharts are beneficial in many ways in program planning.

• **Better communication:** Since a flowchart is a pictorial representation of a program, it is easier for a programmer to explain the logic of the program to some other programmer through a flowchart rather than the program itself.

• **Effective analysis:** The whole program can be analysed effectively through the flowchart as it clearly specifies the flow of the steps constituting the program.

• **Effective synthesis:** If a problem is divided into different modules and the solution for each module is represented in flowcharts separately, they can finally be placed together to visualize the overall system design.

• **Efficient coding:** Once a flowchart is ready, programmers find it very easy to write the concerned program because the flowchart acts as a road map for them. It guides them to go from the starting point of the program to the final point ensuring that no steps are omitted.
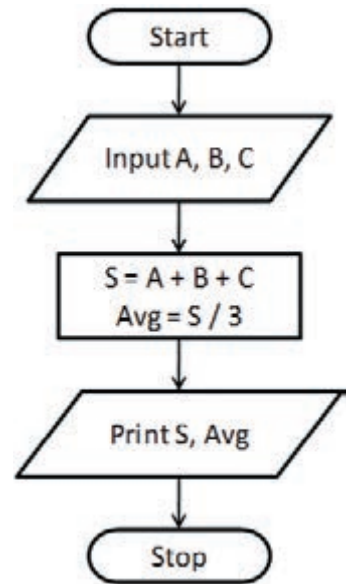
Limitations of flowcharts

In spite of their many obvious advantages, flowcharts have some limitations:

- Flowcharts are very time consuming and laborious to draw with proper symbols and spacing, especially for large complex algorithms.

- Owing to the symbol-string nature of flowcharting, any change or modification in the logic of the algorithm usually requires a completely new flowchart.

- There are no standards determining the amount of detail that should be included in a flowchart.

Now let us develop algorithms and draw flowcharts for solving various problems.

Example 3.2: To find the area and perimeter of a rectangle

We know that this problem can be solved, if the length and breadth of the rectangle are given. The result can be obtained by using the following formula:

Perimeter $= 2$ (Length $+$ Breadth), Area $=$ Length $\times$ Breadth

Let L and B be variables for length and breadth; and P, A be variables for perimeter and area.

Step 1: Start

Step 2: Input L, B

Step 3: $P = 2 * (L + B)$

Step 4: $A = L * B$
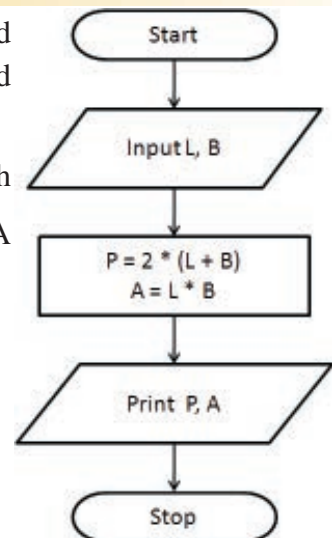
Step 5: Print P, A

Step 6: Stop

The flowchart is given in Figure 3.6.

Fig. 3.6 : Flow chart for Area and Perimeter

The algorithms developed in Examples 3.1 and 3.2 consist of six instructions each. In both the cases, the instructions will be executed one by one in a sequential fashion as shown in Figure 3.7. The order of execution of instructions is known as flow of control. We can say that the two algorithms follow in a sequential flow of control.
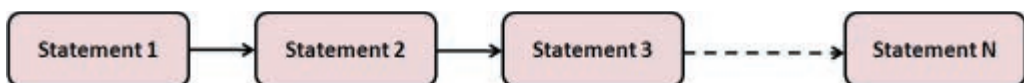

Fig. 3.7 : Sequential flow of control

Develop an algorithm and draw the flow chart to input a time in seconds and convert it into the Hr: Min: Sec format. (For example, if 3700 is given as input, the output should be 1 Hr: 1 Min: 40 Sec).

Let us do

Example 3.3: Find the height of the taller one among two students

Here, two numbers representing the height of two students are to be input. The larger number is to be identified as the result. We know that a comparison between these numbers is to be made for this. The algorithm is given below:

Step 1:   Start
Step 2:   Input H1, H2
Step 3:   If H1 > H2 Then
Step 4:   Print H1
Step 5:   Else
Step 6:   Print H2
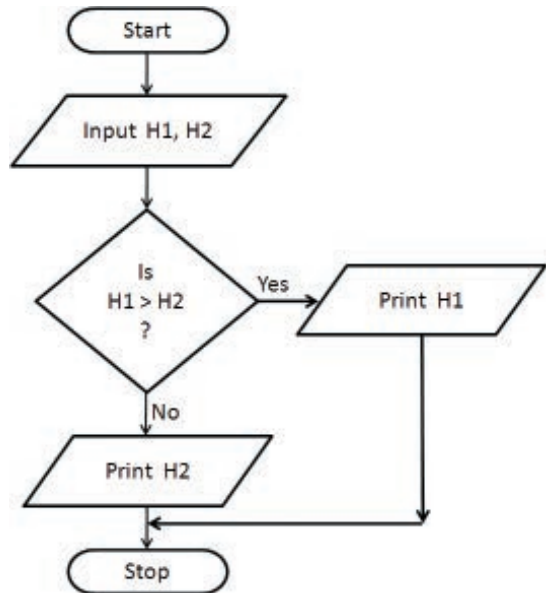Step 7:   End of If
Step 8:   Stop

Fig. 3.8 : Flowchart to find larger value

The flowchart of this algorithm is shown in Figure 3.8. This algorithm uses the decision making aspect. In step 3, a condition is checked. Obviously the result may be True or False based on the values of variables H1 and H2. The decision is made on the basis of this result. If the result is True, step 4 will be selected for execution, otherwise step 6 will be executed. Here one of the two statements (either step 4 or step 6) is selected for execution based on the condition. A branching is done in step 3. That is, this algorithm uses the selection structure to solve the problem. As shown in Figure 3.9, the condition branches the flow to one of the two sets based on the result of condition.
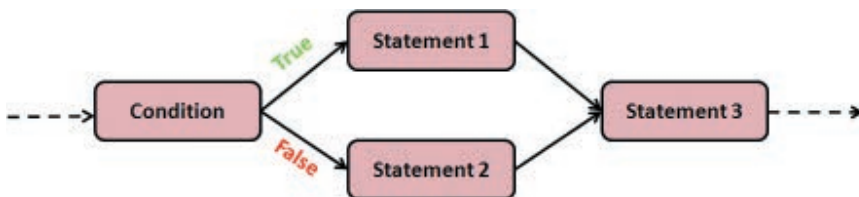

Fig. 3.9 : Selection structure

The working of selection construct is as shown in Figure 3.10. The flow of control comes to the condition; it will be evaluated to True or False. If the condition is True, the instructions given in the true block will be executed and false block will be skipped. But if the condition is False, the true block will be skipped and the false block will be executed. Now let us solve another problem.
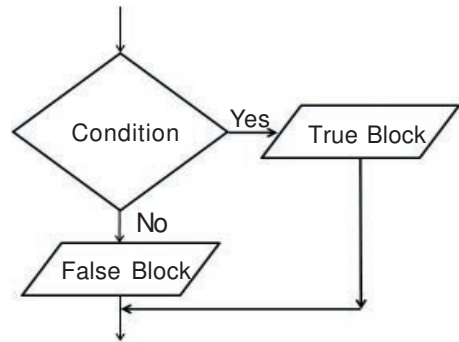

Fig. 3.10 : Flowchart of selection

Example 3.4: To input the scores obtained in 3 unit-tests and find the highest score

Here we have to input three numbers representing the scores and find the largest number among them. The algorithm is given below and flowchart is shown in Figure 3.11.

Step 1:   Start

Step 2:   Input M1, M2, M3

Step 3:   If M1>M2 And M1>M3 Then

Step 4:   Print M1

Step 5:   Else If M2 > M3 Then

Step 6:   Print M2

Step 7:   Else

Step 8:   Print M3

Step 9:   End of If

Step 10:   Stop

This algorithm uses multiple branching based on different conditions. Three different actions are provided, but only one of them is executed. Another point we have to notice is that the first condition consists of two comparisons. This kind of condition is known as compond condition.
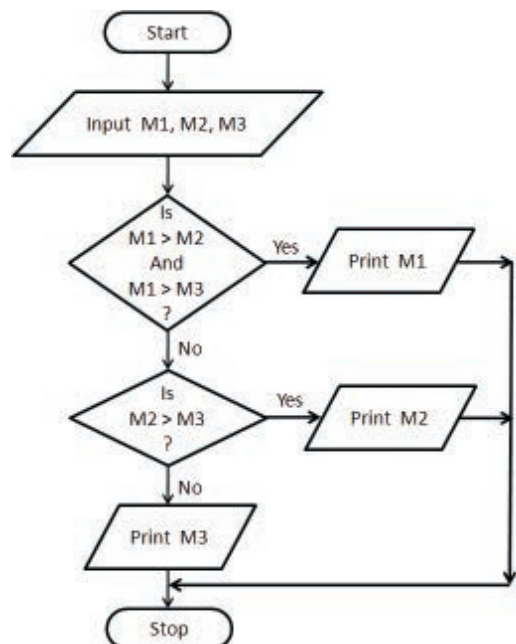

Fig. 3.11 : Flowchart to find the largest of three numbers

Let us do

1. *Develop an algorithm and draw the flowchart to check whether a given number is even or odd.*

2. *Design an algorithm and flow chart to input a day number and display the name of the day. (For example, if 1 is the input, the output should be Sunday. If it is 2, the output should be Monday. If the number is other than 1 to 7, the output should be "Invalid data").*

3. *Based on the evaluation system for standard X, develop an algorithm to accept a score out of 100 and find the grade.*

Now, consider a case in which some task is to be performed in a repeated fashion. Suppose we want to print the first 100 natural numbers. How do we do it? We know that the first number is 1. It should be printed. The next number is obtained by adding 1 to the first number. Again it should be printed. It is clear that the two tasks - printing a number and adding 1 to it - are to be executed repeatedly. The execution should be terminated when the last number is printed. Let us develop the algorithm for this.

Example 3.5: To print the numbers from 1 to 100

Step 1:   Start

Step 2:   N = 1

Step 3:   Print N

Step 4:   N = N + 1

Step 5:   If N <=100 Then Go To Step 3

Step 7:   Stop

In the algorithm given as Example 3.5, a condition is checked at step 5. If the condition is found true, the flow of control is transferred back to step 3. So the steps 3, 4 and 5 are executed repeatedly as long as the condition is true. We will say that a loop is formed here. Steps 3, 4 and 5 constitute a loop. The control comes out of the loop only when the condition becomes false. The flowchart of this algorithm is shown in Figure 3.12.

The above algorithm can be simplified as follows:
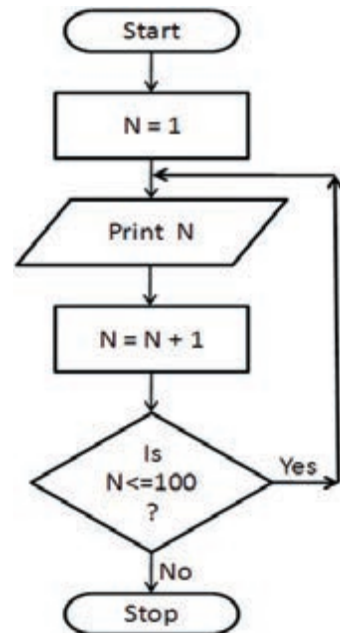
Step 1:   Start

Step 2:   N = 1

Fig. 3.12 : Flowchart to print numbers from 1 to 100

Step 3:   Repeat Steps 4 and 5 While (N<=100)

Step 4:   Print N

Step 5:   N = N + 1

Step 6:   Stop

Note that in step 3, the words "**Repeat**" and "**While**" are used to construct a loop. The statements (step numbers) that need repeated execution is specified with the word "**Repeat**" and the condition is given with "**While**". As the algorithm looks slightly different, the flow chart also differs slightly as in Figure 3.13. The execution style of a loop is shown in Figure 3.14.
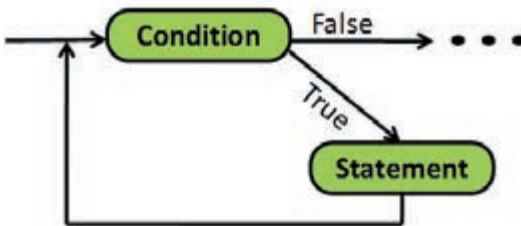

Fig. 3.13 : Flowchart to print numbers from 1 to 100

Fig. 3.14 : Looping construct

A loop has four elements. Obviously, one of them is the *condition*. We know that at least one variable will be used to put up a condition and let us call it loop control variable. Before the condition being checked, the loop control variable should get a value. It is possible through input or assignment. Such an instruction is called **initialisation instruction** for the loop. The third element, called **update instruction**, changes the value of the loop control variable. It is essential; otherwise the execution of the loop will never be terminated. The fourth element is the **loop body**, which is the set of instructions to be executed repeatedly. The flowchart shown in Figure 3.15 depicts the working of looping structure.

Fig. 3.15 : Flowchart of Looping

The initialisation instruction will be executed first and then the condition will be checked. If the condition is true, the body of the loop will be executed followed by the update instruction. After the execution of the update instruction, the condition will be checked again. This process will be continued until the condition
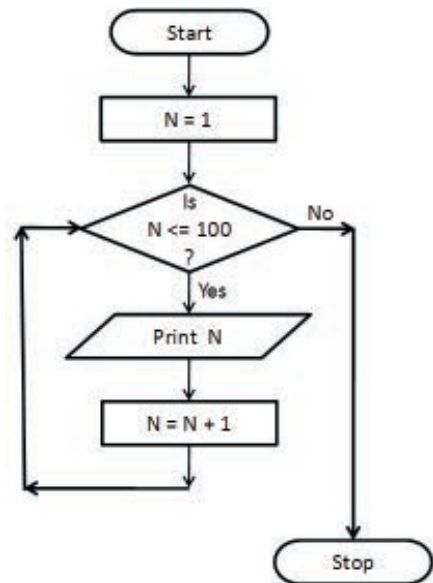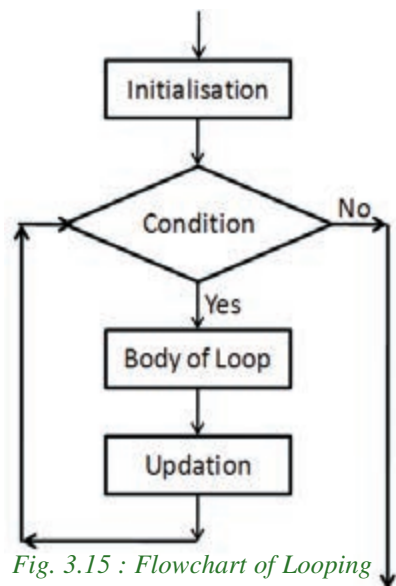
becomes false. The loop that checks the condition before executing the body is called entry-controlled loop. There is another style of looping construct. In this case, the condition will be checked only after the execution of loop-body and update instruction. Such a loop is called exit-controlled loop.

Example 3.6: To print the sum of the first N natural numbers

Here we have to input the value of N. The sum of numbers from 1 to the input number N is to be found out. Let S be the variable to store the sum. Figure 3.16 shows the flowchart of this algorithm

Step 1:   Start

Step 2:   Input N

Step 3:   A = 1, S = 0

Step 4:   Repeat Steps 5 and 6 While (A <= N)

Step 5:   S = S + A

Step 6:   A = A + 1

Step 7:   Print S

Step 8:   Stop
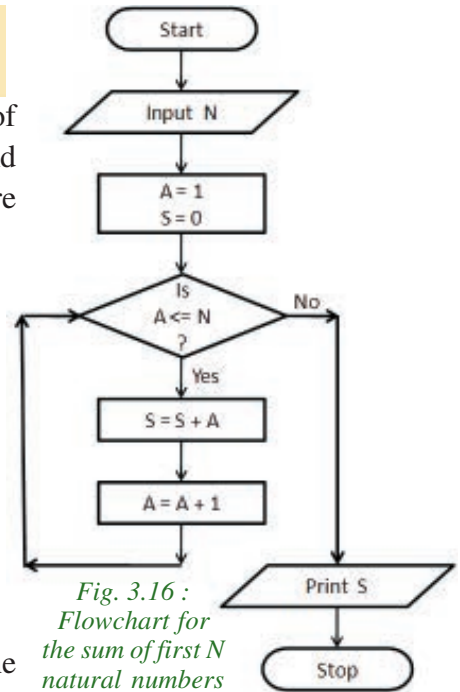
Fig. 3.16 :
Flowchart for
the sum of first N
natural numbers

This algorithm uses an entry-controlled loop. In the next example we can see an algorithm that uses exit-controlled loop for problem solving.

Example 3.7: To print the first 10 multiples of a given number

Step 1:   Start

Step 2:   Input N

Step 3:   A = 1

Step 4:   M = A × N

Step 5:   Print M

Step 6:   A = A + 1

Step 7:   Repeat Steps 4 to 6 While (A <= 10)

Step 8:   Stop

This algorithm and the corresponding flowchart shown in Figure 3.17 contain a loop in which the condition is checked only after executing the body. Table 3.1 shows comparison between entry controlled loops and exit controlled loops.
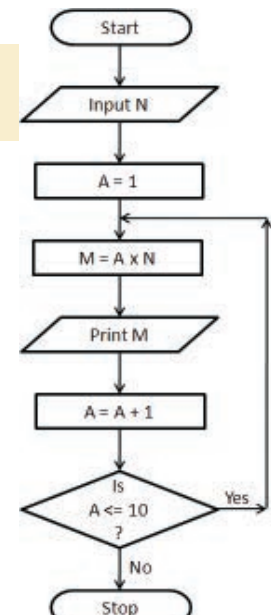
Fig. 3.17 : Flowchart for
the first 10 multiples of a number

Entry controlled loop	Exit controlled loop
• Condition is checked before the execution of the body	• Condition is checked after the execution of the body
• Body may never be executed.	• Body will surely be executed at least once.
• Suitable when skipping of the body from being executed is required	• Suitable when normal execution of the body is to be ensured.

Table 3.1 : Comparison of loops

Let us practice with more algorithms and flowcharts for solving the problems that are given as learning activities.

Develop an algorithm and draw the flowcharts for the following problems:

Let us do

1. *To print all even numbers below 100 in the descending order.*
2. *To find the sum of odd numbers between 100 and 200.*
3. *To print the multiplication table of a given number.*
4. *To find the factorial of a number.*
5. *To input a number and check whether it is prime or not.*

3.3.3 Coding the program

Once we have developed the skill to design algorithms and flowcharts, the next step in programming is to express the instructions in a more precise and concise notation. That is, the instructions are to be expressed in a programming language. The process of writing such program instructions to solve a problem is called **coding**. Text editor programs are available to write the code in computer.

A language is a system of communication. We communicate our ideas and emotions to one another through natural languages such as English, Malayalam, etc. Similarly, a computer language is used to communicate between user and computer. A human being who writes a computer program is to be familiar with a language that is understandable to the computer also. We have already seen that computer knows only the binary language which is very difficult for human beings to understand and use. As we saw in Chapter 2, we can use a human friendly language, known as High Level Language (HLL) that looks similar to English. Again there is a facility of using language processors to convert or translate the program

written in HLL into machine language. The program written in any HLL is known as **source code**.

Hence, to be a programmer we have to be well-versed in any HLL such as BASIC, COBOL, Pascal, C++ etc. to express the instructions in a program. Each language has its own character set, vocabulary, grammar (we call it syntax) to write programs. Once the program is written using a language, it should be saved in a file (called source file), and then proceed to the next phase of programming.

3.3.4 Translation

While selecting a language for developing source code, certain criteria such as volume of data, complexity in process, usage of files, etc. are to be considered. Once a language is selected and the source code is prepared, it should be translated using the concerned language processor. **Translation** is the process of converting a program written in high level language into its equivalent version in machine language. The compiler or interpreter is used for this purpose. During this step, the syntax errors of the program will be displayed. These errors are to be corrected by opening the file that contains the source code. The source code is again given for compilation (translation). This process will be continued till we get a message such as "No errors or warnings" or "Successful compilation". Now we have a program fully constituted by machine language instructions. This version of the source code is known as **object code** and it will be usually stored in a file by the compiler itself.

Fig. 3.18 : Translation Process

Once the object code is obtained it should be present in the system as long as you want the program to be used.

3.3.5 Debugging

Debugging is the stage where programming errors are discovered and corrected. As long as computers are programmed by human beings, the programs will be subject to errors. Programming errors are known as 'bugs' and the process of detecting and correcting these errors is called **debugging**. In general there are two types of errors that occur in a program - syntax errors and logical errors. **Syntax errors** result when the rules or syntax of the programming language are not followed. Such program errors typically involve incorrect punctuation, incorrect word sequence, undefined term, or illegal use of terms or constructs. Almost all language processors detect syntax errors when the program is given for translation. They print error messages that include the

line number of the statement having errors and give hints about the nature of the error. In the case of interpreters, the syntax errors will be detected and displayed during execution. The programmer's efficiency in using the language decides the time and effort for the debugging process. The object program will be generated only if all the syntax errors are rectified.

The second type of error, named **logical error**, is due to improper planning of the program's logic. The language processor successfully translates the source code into machine code if there are no syntax errors. During the execution of the program, computer actually follows the program instructions and gives the output as per the instructions. But the output may not be correct. This is known as logical error. When a logical error occurs, all you know is that the computer is not giving the correct output. The computers do not tell us what is wrong. It should be identified by the programmer or user. In order to determine whether or not there is a logical error, the program must be tested. So, let us move on to the next stage of programming.

3.3.6 Execution and testing

As we have seen in the previous section, the program is said to be error-free only when logical errors are also rectified. Hence when the compiled version of the program is formed, it should be executed for testing. The purpose of testing is to determine whether the results are correct. The testing procedure involves running the program to process the test data that will produce 'known results'. That is, the operations involved in the program should be done manually and the output thus obtained should be compared with the one given by the computer. The accuracy of the program logic can be determined by this testing. While selecting the test data, we should ensure that all aspects of the program logic will be tested. Hence the selection of proper test data is important in program testing.

Till now, we have discussed incorrect outputs due to incorrect logic. But there is a chance of another type of error, which will interrupt the program execution. This may be due to the inappropriate data that is encountered in an operation. For example consider an instruction A= B/C. This statement causes interruption in execution if the value of C happens to be zero. In such a situation, the error messages may be displayed by the error-handling function of the language. These errors are known as **Run-time error**. These errors can be rectified by providing instructions for checking the validity of the data before it gets processed by the subsequent instructions in the program.

3.3.7 Documentation

A computerised system cannot be considered to be complete until it is properly documented. In fact documentation is an on-going process that starts in the problem-study phase of the system and continues till its implementation and operation. We can write comments in the source code as part of documentation. It is known as **internal documentation**. It helps the debugging process as well as program modification at a later stage. The logic that we applied in the program may not be remembered when we go through our own program at a later stage. Besides, the program written by one person may need to be modified by some other person in future. If the program is documented, it will help to understand the logic we applied, the reason why a particular statement has been used and so on. However, the documentation part of the program will not be considered by the language processor when you give the program for translation.

Writing comments in programs is only a part of documentation. Another version of documentation is the preparation of system manual and user manual. These are hard copy documents that contain functioning of the system, its requirements etc. and the procedure for installing and using the programs. While developing software for various applications, these manuals are mandatory. This kind of documentation is known as **external documentation**.

Now you have analysed the problem, derived the logic of the solution, expressed in a flow chart, developed the code in a programming language, translated it after removing the syntax errors, checked the accuracy of the output after removing all the possible logical and run-time errors, and we have documented the program.

> **Check yourself**
>
> 1. What is an algorithm?
> 2. Pictorial representation of algorithm is known as _____.
> 3. Which flow chart symbol is always used in pair?
> 4. Which flow chart symbol has one entry flow and two exit flows?
> 5. Program written in HLL is known as _____.
> 6. What is debugging?
> 7. What is an object code?

3.4 Performance evaluation of algorithms

We have developed algorithms for solving various problems. You may think that some of these problems would have been solved by following a different logic. Of course, it is true that the same problem can be solved by different sets of instructions. But an efficient programmer is the one who develops algorithms that require minimum computer resources for execution and give results with high accuracy in lesser time. The performance of an algorithm is evaluated based on the concept of time and space complexity. The algorithm which will be executed faster with minimum amount of memory space is considered as the best algorithm for the problem.

Algorithm-1	Algorithm-2
Step 1: Start Step 2: Input A, B, C Step 3: S = A + B + C Step 4: Avg = S / 3 Step 5: Print S, Avg Step 6: Stop	Step 1: Start Step 2: Input A, B, C Step 3: S = A + B + C Step 4: Avg = (A + B + C) / 3 Step 5: Print S, Avg Step 6: Stop

Table 3.2 : Algorithms to find the sum and average of three numbers

Let us compare the two algorithms given in Table 3.2, developed to find the sum and average of three numbers. The two algorithms differ in step 4. Algorithm-2 uses two steps (steps 3 and 4) for addition operation on the same data. Naturally, that algorithm will take more time for execution than Algorithm-1. So Algorithm-1 is better for coding.

Now let us take another case, where comparison operations are involved for the selection of a statement. We have already discussed an algorithm to find the largest among three numbers in Example 3.4. The two algorithms given in Table 3.3 can also be used for solving the same problem.

Algorithm-1	Algorithm-2
Step 1: Start Step 2: Input M1, M2, M3 Step 3: If M1>M2 And M1>M3 Then Step 4: Print M1 Step 5: If M2 > M1 And M2 > M3 Then Step 6: Print M2 Step 7: If M3 > M1 And M3 > M2 Then Step 8: Print M3 Step 9: Stop	Step 1: Start Step 2 Input M1, M2, M3 Step 3: If M1 > M2 Then Step 4: Big = M1 Step 5: Else Step 6: Big = M2 Step 7: If M3 > Big Then Big = M3 Step 8: Print Big Step 9: Stop

Table 3.3 : Algorithms to find the largest among three numbers

The algorithm in Example 3.4 has three comparison operations and one logical operation altogether. All these operations are to be carried out only when the largest value is in M3 (the third variable). To identify the speed of execution in each case, let us assume that 1 second is required for one comparison operation. We can see that fastest result will be in 3 seconds and slowest in 4 seconds. So the average speed is 3.5 seconds.

Now let us analyse Algorithm-1 in Table 3.3. There are three **If** statements, each with three comparison operations. If we follow the assumptions specified above, we can see that the result will be obtained in 9 seconds, irrespective of the values in the variables. So the average speed is 9 seconds. But the Algorithm-2 in Table 3.3 uses two If structures. The algorithm shows that whatever be the values in the variables, the time required for comparison will be 2 seconds. Thus the average speed is 2 seconds. So, we can say that the Algorithm-2 is better than the other two.

Let us consider one more case where loop is involved. The two algorithms given in Table 3.4 find the sum of all even numbers and sum of all odd numbers between 100 and 200.

Algorithm-1	Algorithm-2
Step 1: Start	Step 1: Start
Step 2: N = 100, ES = 0	Step 2: N = 100, ES = 0, OS = 0

Step 3:   Repeat Steps 4 to 6
            While (N <= 200)
	Step 3: Repeat Steps 4 to 8
	            While (N <= 200)

Step 4: If Remainder of N/2 = 0 Then	Step 4: If Remainder of N/2 = 0 Then
Step 5: ES = ES + N	Step 5: ES = ES + N
Step 6: N = N + 1	Step 6: Else
Step 7: Print ES	Step 7: OS = OS + N
Step 8: N = 100, OS = 0	Step 8: N = N + 1
Step 9: Repeat Steps 10 to 12 While (N <= 200)	Step 9: Print ES
Step 10: If Remainder of N/2 = 1 Then	Step 10: Print OS
Step 11: OS = OS + N	Step 11: Stop
Step 12: N = N + 1	
Step 13: Print OS	
Step 14: Stop	

Table 3.4 : Algorithms to find sum of even and odd numbers

Algorithm-1 uses two loops. Obviously, time taken will be double for the intialisation, testing and updation of loop control variable compared to Algorithm-2. From the table, it is clear that Algorithm-2 is better and efficient. So, think divergently and differently to develop logic for solving problems.

Let us sum up

Program is a sequence of instructions written in a computer language. The process of programming proceeds through some stages. Preparation of algorithms and flowcharts help develop the logic. The program written in HLL is known as source code and it is to be converted into machine language. The resultant code is known as object code. The errors occurred in a program has to be removed through a process known as debugging. The translated version is executed by the computer. Proper documentation of the program helps us to modify it at a later stage. While solving problems different logics may be applied, but the performance is measured in terms of time and space complexity.

Learning outcomes

After the completion of this chapter the learner will be able to
*   explain various aspects of problem solving.
*   develop algorithms for solving problems.
*   draw flowcharts to ensure the correctness of algorithms.
*   select the best algorithm for solving a problem.

Sample questions

Very short answer type
1. What is an algorithm?
2. What is the role of a computer in problem solving?
3. What is the use of connector in a flow chart?
4. What do you mean by logical errors in a program?

Short answer type
1. What is a computer program? How do algorithms help to write programs?
2. Write an algorithm to find the sum and average of 3 numbers.
3. Draw a flowchart to display the first 100 natural numbers.
4. What are the limitations of a flow chart?
5. What is debugging?
6. What is the need of documentation for a program?

Long answer type
1. What are the characteristics of an algorithm?
2. What are the advantages of using a flowchart?
3. Briefly explain different phases in programming.

4

Getting Started with C++

C++ (pronounced "C plus plus") is a powerful, popular object oriented programming (OOP) language developed by Bjarne Stroustrup. The idea of C++ comes from the C increment operator ++, thereby suggesting that C++ is an added (incremented) version of C language.

The C++ language can be used to practice various programming concepts such as sequence, selection and iteration which we have already discussed in Chapter 3. In this chapter, we will have a brief overview of the fundamentals of C++. We will also familiarise different language processor packages that are used to write C++ programs.

Just like any other language, the learning of C++ language begins with the familiarisation of its basic symbols called characters. The lerning hierarchy proceeds through words, phrases (expressions), statements, etc. Let us begin with the learning of characters.

4.1 Character set

As we know, the study of any language, such as English, Malayalam or Hindi begins with the alphabet. Similarly, the C++ language also has its own alphabet. With regard to a programming language the alphabet is known as character set. It is a set of valid symbols, called characters

Dr. Bjarne Stroustrup developed C++ at AT&T Bell Laboratories in Murray Hill, New Jersey, USA. Now he is a visiting Professor at Columbia University and holder of the College of Engineering Chair in Computer Science at Texas A&M University. He has received numerous honors. Initial name of this language was 'C with classes'. Later it was renamed to C++, in 1983.

Bjarne Stroustrup

set. It is a set of valid symbols, called characters that a language can recognize. A character represents any letter, digit, or any other symbol. The set of valid characters in a language which is the fundamental units of that language, is collectively known as **character set.** The character set of C++ is categorized as follows:

(i) Letters : A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

(ii) Digits : 0 1 2 3 4 5 6 7 8 9

(iii) Special characters : + - * / ^ \ ( ) [ ] { } = < > . ' "
$, ; : % ! & ? _ (underscore) # @

(iv) White spaces : Space bar (Blank space), Horizontal Tab (→), Carriage Return (↵) , Newline, Form feed

(v) Other characters : C++ can process any of the 256 ASCII characters as data or as literals.

Spaces, tabs and newlines (line breaks) are called white spaces. White space is required to separate adjacent words and numbers.

4.2 Tokens

After learning the alphabet the second stage is learning words constituted by the alphabet (or characters). The term 'token' in the C++ language is similar to the term 'word' in natural languages. **Tokens** are the fundamental building blocks of the program. They are also known as lexical units. C++ has five types of tokens as listed below:

1. Keywords
2. Identifiers
3. Literals
4. Punctuators
5. Operators

4.2.1 Keywords

The words (tokens) that convey a specific meaning to the language compiler are called **keywords**. These are also known as reserved words as they are reserved by the language for special purposes and cannot be redefined for any other purposes. The set of 48 keywords in C++ are listed in Table 4.1. Their meaning will be explained in due course.

asm	continue	float	new	signed	try
auto	default	for	operator	sizeof	typedef
break	delete	friend	private	static	union
case	do	goto	protected	struct	unsigned
catch	double	if	public	switch	virtual
char	else	inline	register	template	void
class	enum	int	return	this	volatile
const	extern	long	short	throw	while

Table 4.1 : Keywords of C++

4.2.2 Identifiers

We usually assign names to places, people, objects, etc. in our day to day life, to identify them from one another. In C++ we use identifiers for this purpose. **Identifiers** are the user-defined words that are used to name different program elements such as memory locations, statements, functions, objects, classes etc. The identifiers of memory locations are called *variables*. The identifiers assigned to statements are called *labels*. The identifiers used to refer a set of statements are called *function names*.

While constructing identifiers certain rules are to be strictly followed for their validity in the program. The rules are as follows:

- Identifier is an arbitrary long sequence of letters, digits and underscores (_).

- The first character must be a letter or underscore (_).

- White space and special characters are not allowed.

- Keywords cannot be used as identifiers.

- Upper and lower case letters are treated differently, i.e. C++ is case sensitive.

Examples for some valid identifiers are `Count`, `Sumof2numbers`, `_1stRank`, `Main`, `Average_Height`, `FOR`

The following are some invalid identifiers due to the specified reasons:

Sum of Digits → Blank space is used

1styear → Digit is used as the first character

First.Jan → Special character (.) is used

for → It is a keyword

Identify invalid identifiers from the following list and give reasons:

Let us do

```
Data_rec,   _data,   Idata,   data1,   my.file,
asm,   switch,   goto,   break
```

4.2.3 Literals

Consider the case of the Single Window System for the admission of Plus One students. You may have given your date of birth in the application form. As an applicant, your date of birth remains the same throughout your life. Once they are assigned their initial values, they never change their value. In mathematics, we know that the value of π is a constant and the value of gravitational constant 'g' never changes, i.e. it remains 9.8m/s². Like that, in C++, we use the type of tokens called **literals** to represent data items that never change their value during the program run. They are often referred to as constants. Literals can be divided into four types as follows:

1. Integer literals
2. Floating point literals
3. Character literals
4. String literals

Integer literals

Consider the numbers 1776, 707, -273. They are integer constants that identify integer decimal values. The tokens constituted only by digits are called **integer literals** and they are whole numbers without fractional part. The following are the characteristics of integer literals:

• An integer constant must have at least one digit and must not contain any decimal point.

• It may contain either + or – sign as the first character, which indicates whether the number is positive or negative.

- A number with no sign is treated as positive.

- No other characters are allowed.

Classify the following into valid and invalid integer constants and give your reasons:

Let us do

| 77,000 | 70 | 314. | -5432 | +15346 |
| +23267 | -.7563 | -02281+0 | 1234E56 | -9999 |

In addition to decimal numbers (base 10), C++ allows the use of octal numbers (base 8) and hexadecimal numbers (base 16) as literals (constants). To express an octal number we have to precede it with a 0 (zero character) and in order to express a hexadecimal number we have to precede it with the characters 0x (zero, x). For example, the integer constants 75, 0113 and 0x4B are all equivalent to each other. All of these represent the same number 75 (seventy-five), expressed as a base-10 numeral, octal numeral and hexadecimal numeral, respectively.

Floating point literals

You may have come across numbers like 3.14159, 3.0×10^8, 1.6×10^{-19} and 3.0 during your course of study. These are four valid numbers. The first number is π (Pi), the second one is the speed of light in meter/sec, the third is the electric charge of an electron (an extremely small number) – all of them are approximated, and the last one is the number three expressed as a floating-point numeric literal.

Floating point literals, also known as real constants are numbers having fractional parts. These can be written in one of the two forms called fractional form or exponential form.

A real constant in fractional form consists of signed or unsigned digits including a decimal point between digits. The rules for writing a real constant in fractional form are given below:

- A real constant in fractional form must have at least one digit and a decimal point.

- It may also have either + (plus) or – (minus) sign preceding it.

- A real constant with no sign is assumed to be positive.

A real constant in exponential form consists of two parts: *mantissa* and *exponent*. For instance, 5.8 can be written as $0.58 \times 10^1 = 0.58E1$ where mantissa part is 0.58 (the part appearing before **E**) and exponential part is 1 (the part appearing after **E**). The number E1 represents 10^1. The rules for writing a real constant in exponential form are given below:

- A real constant in exponent form has two parts: a mantissa and an exponent.

- The mantissa must be either an integer or a valid fractional form.

- The mantissa is followed by a letter **E** or **e** and the exponent.

- The exponent must be an integer.

The following are valid real constants.

52.0	107.5	-713.8	-.00925
453.E-5	1.25E08	.212E04	562.0E09
152E+8	1520E04	-0.573E-7	-.097

Some invalid real constants are given along with the reason:

58,250.262 (Comma is used), 5.8E (No exponent part), 0.58E2.3 (Fractional number is used as exponent).

> *Classify the following into valid and invalid real constants and justify your answer:*
>
> | 77, 00,000 | 7.0 | 3.14 | -5.0E5.4 | +53.45E-6 |
> |------------|-----|------|----------|-----------|
> | +532.67. | .756E-3 | -0.528E10 | 1234.56789 | 34,56.24 |
> | 4353 | +34/2 | 5.6E | 4356 | 0 |
>
> **Let us do**

Character literals

When we want to store the letter code for gender usually we use 'f' or 'F' for *Female* and 'm' or 'M' for *Male*. Similarly, we may use the letter 'y' or 'Y' to indicate *Yes* and the letter 'n' or 'N' to indicate *No*. These are single characters. When we refer a single character enclosed in single quotes that never changes its value during the program run, we call it a **character literal** or **character constant**.

Note that x without single quote is an identifier whereas 'x' is a character constant. The value of a single character constant is the ASCII value of the character. For instance, the value of 'c' will be 99 which is the ASCII value of 'c' and the value of 'A' will be the ASCII value 65.

C++ language has certain non-graphic character constants, which cannot be typed directly from the keyboard. For example, there is no way to express the Carriage Return or Enter key, Tab key and Backspace key. These non-graphic symbols can be represented by using **escape sequences**, which consists of a backslash (\) followed by one or more characters. It should be noted that even though escape sequences contain more than one character enclosed in single quotes, it uses only one corresponding ASCII code to represent it. That is why they are treated as character constants. Table 4.2 lists escape sequences and corresponding characters.

Escape sequence	Corresponding non-graphic character
\a	Audible bell (alert)
\b	Back Space
\f	Form feed
\n	New line or Line feed
\r	Carriage Return
\t	Horizontal Tab
\v	Vertical Tab
\\	Back slash
\'	Single quote
\"	Double quote
\?	Question mark
\0	Null character

Table 4.2 : Escape Sequences

In Table 4.2, you can also see sequences representing \', \" and \?. These characters can be typed from the keyboard but when used without escape sequence, they carry a special meaning and have a special purpose. However, if these are to be displayed or printed as it is, then escape sequences should be used. Examples of some valid character constants are: 's', 'S', '$', '\n', '+', '9'

C++ represents Octal Number and Hexadecimal Number with the help of escape sequences. The \0n and \xHn represent a number in the Octal Number System and the Hexadecimal Number System respectively.

Some invalid character constants are also given with the reason for invalidity:

A (No single quotes), '82' (More than one character), "K" (Double quotes instead of single quotes), '\g' (Invalid escape sequence or Multiple characters).

String literals

Nandana is a student and she lives in Bapuji Nagar. Here, "Nandana" is the name of a girl and "Bapuji Nagar" is the name of a place. These kinds of data may need to be processed with the help of programs. Such data are considered as string constants and they are enclosed within double quotes. A sequence of one or more characters enclosed

within a pair of double quotes is called **string constant**. For instance, `"Hello friends"`, `"123"`, `"C++"`, `"Baby\'s Day Out"`, etc. are valid string constants.

Classify the following into different categories of literals.

```
        'a'       "rita"     -124     12.5     -12e-1

"raju\'s pen"  0      -11.999   '\\'     32760
```

Let us do

4.2.4 Punctuators

In languages like English, Malayalam, etc. punctuation marks are used for grammatical perfection of sentences. Consider the statement: *Who developed C++?* Here '?' is the punctuation mark that tells that the statement is a question. Similarly at the end of each sentence we put a full stop (.). In the same way C++ also has some special symbols that have syntactic or semantic meaning to the compiler. These are called **punctuators**. Examples are: `#` `;` ' " ( ) [ ] { }

The purpose of each punctuator will be discussed in due course.

4.2.5 Operators

When we have to add 5 and 3, we express it as 5 + 3. Here + is an operator that represents the addition operation. Similarly, C++ has a rich collection of operators. An **operator** is a symbol that tells the compiler about a specific operation. They are the tokens that trigger some kind of operation. The operator is applied on a set of data called **operands**. C++ provides different types of operators like arithmetic, relational, logical, assignment, conditional, etc. We will discuss more about operators in the next chapter.

Classify the following into different categories of tokens.

```
    /        -124      +    -12e-1    '    "KL01"

    Sum "raju\'s pen"if   rita   '\\'  break  }
```

Let us do

4.3 Integrated Development Environment (IDE)

Now we have learned the basic elements of a C++ program. Before we start writing C++ programs, we must know where we will type this program. Like other programming languages, a text editor is used to create a C++ program. The compilers such as GCC (GNU Compiler Collection), Turbo C++, Borland C++, and many other similar

compilers provide an Integrated Development Environment (IDE) for developing C++ programs. Many of these IDEs provide facilities for typing, editing, searching, compiling, linking and executing a C++ program. We use GCC with Geany IDE (IT@School Ubuntu Linux 12.04) for the purpose of illustrating the procedure for coding, compiling and executing C++ programs.

GCC with Geany IDE

There are many compilers available like GCC, Borland C++, Turbo C++, Microsoft C++ (Visual C++), Unix AT & T C++, etc. Out of these, GCC is a free software available with Linux operating system. GCC stands for GNU Compiler Collection and is one of the popular C++ compilers which works with ISO C++ standard. GCC versions are available for Windows platforms also. Geany is a cross-platform IDE that works with GCC for writing, compiling and executing C++ programs.

a. Opening the edit window

The edit window of Geany IDE can be opend from the **Applications** menu of Ubuntu Linux by proceeding as follows:

<div align="center">

Applications → Programming → Geany

</div>

Geany IDE opens its windows as shown in Figure 4.1. It has a title bar, menu bar, tool bar and a code edit area. We can see a tab named **untitled**, which indicates the file
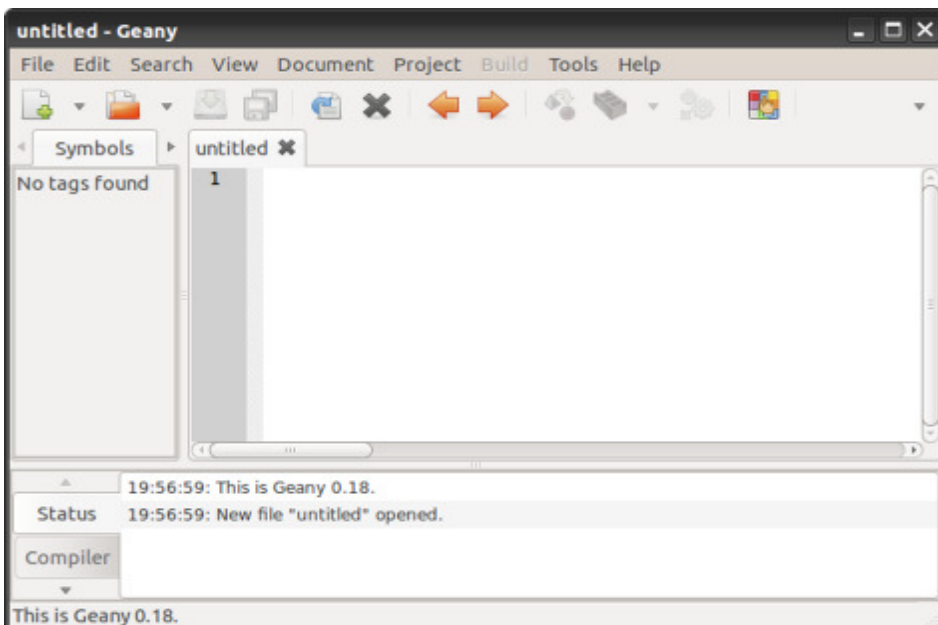

Fig. 4.1: Edit window of Geany IDE for Ubuntu Linux 12.04

name for the opened edit area. If we use Geany 1.24 on Windows OS, the opening window will be as shown in Figure 4.2. We can see that both of these are quite the same.
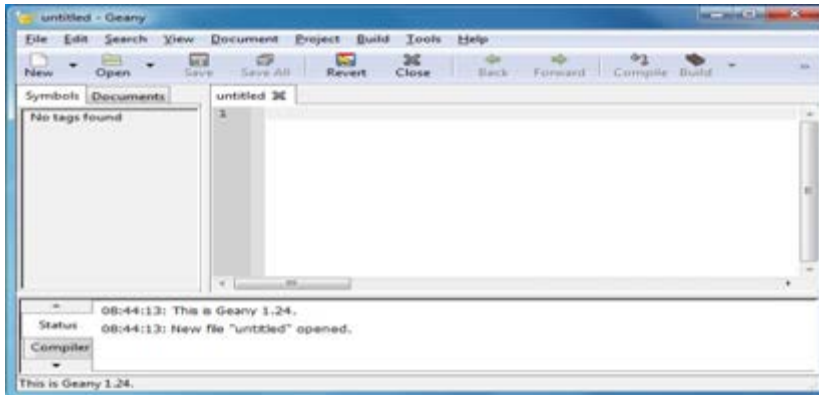

Fig. 4.2: Edit window of Geany IDE 1.24

In this window, we type the program in a file with the default name untitled. To open a new file, choose **File** menu, then select **New** option or click New button on the toolbar.

b. Saving the program

Once a file is opened, enter the C++ program and save it with a suitable file name with extension **.cpp**. GCC being a collection of compilers, the extension decides which compiler is to be selected for the compilation of the code. Therefore we have to specify the extension without fail. If we give the file name before typing the program, GCC provides different colours automatically to distinguish the types of tokens used in the program. It also uses indentation to identify the level of statements in the source code. We will disucss the concept of indentation later.

Let us write a simple program given as Program 4.1 and save with the name **welcome.cpp**.

Program 4.1: **A program to familiarise the IDE**

```cpp
// my first C++ program
#include<iostream>
using namespace std;
int main()
{
    cout << "Welcome to the world of C++";
    return 0;
}   //end of program
```

The IDE window after entering Program 4.1 is shown in Figure 4.3. Observe the diffrence in colours used for the tokens.
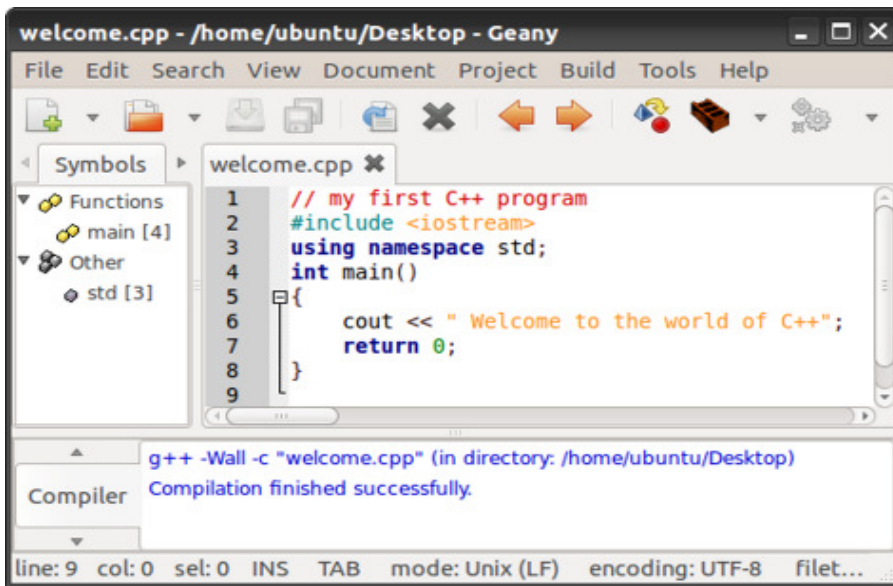
Fig. 4.3 : Program saved with a name in Geany

To save the program, choose **File** menu and select **Save** option or use the keyboard shortcut **Ctrl+S**. Alternatively the file can be saved by clicking the Save button ⬇️ in the toolbar.

It is a good practice to save the program every now and then, just by pressing **Ctrl+S**. This helps to avoid the loss of data due to power failures or due to unexpected system errors. Once the program typing is completed, it is better to save the file before compiling or modifying. Copying the files from the temporary volatile primary memory to permanent non volatile secondary memory for storage is known as **saving** the program.

C++ program files should have a proper extension depending upon the implementation of C++. Different extensions are followed by different compilers. Examples are .cpp, .cxx, .cc, .c++
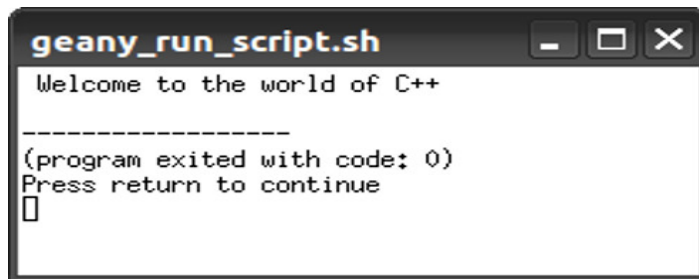
c. Compiling and linking the program

The next step is to compile the program and modify it, if errors are detected. For this, choose **Build** menu and select **Compile** option. Alternatively, we can use the Compile button. If there are some errors, those errors will be displayed in the compiler status window at the bottom, otherwise the message Compilation finished successfully will be displayed. (refer Figure 4.3).

After successful compilation, choose **Build** menu and select **Build** option for linking or click the Build button in the toolbar. Now the program is now ready for execution.

d. Running/Executing the program

Running the program is the process by which a computer carries out the instructions of a computer program. To run the program, choose **Build** menu and select **Execute** option. The program can also be executed by pressing the function key **F5**. The output will be displayed in a new window as shown in Figure 4.5.

Fig. 4.5 : Output window of Geany

e. Closing the IDE

Once we have executed the program and desired output is obtained, it can be closed by selecting **Close** option from **File** menu or by clicking the **X** symbol in the active tab or in the title bar of the IDE. For writing another program, a new file can be opened by the **New** option from the **File** menu or by clicking the **New** button in the tool bar. The key combination **Ctrl+N** can also be used for the same.

After developing program, we can come out of Geany IDE by choosing **File** menu and selecting **Quit** option. The same can be achieved by clicking the Close button of the IDE window or by using the key combination **Ctrl+Q**.

1. Write a program to print the message "SMOKING IS INJURIOUS TO HEALTH" on screen.

2. Write a program to display the message "TOBACCO CAUSES CANCER" on monitor.

Let us do

Let us sum up

C++ was developed by Bjarne Stroustrup in early 1980s. C++ has its own character set. Tokens are the smallest unit of a program and are constituted by one or more characters in C++. There are five types of tokens namely keywords, identifiers, literals, punctuators and operators. Programs are written in computer with the help of an editor. Software like GCC and Geany IDE provide facilities to enter the source code in the computer, compile it and execute the object code.

Learning outcomes

After the completion of this chapter the learner will be able to

- list the C++ character set.
- categorise various tokens.
- identify keywords.
- write valid identifiers.
- classify various literals.
- identify the main components of Geany IDE.
- write, compile and run a simple program.

Sample questions

Very short answer type

1. What are the different types of characters in C++ character set?

2. What is meant by escape sequences?

3. Who developed C++?

4. What is meant by tokens? Name the tokens available in C++.

5. What is a character constant in C++ ?

6. How are non-graphic characters represented in C++? Give an example.

7. Why are the characters \ (slash), ' (single quote), " (double quote) and ? (question mark) typed using escape sequences?

8. Which escape sequences represent newline character and null character?

9. An escape sequence represents _____ characters.

10. Which of the following are valid character/string constants in C++?

    ```
    'c'        'anu'         "anu"       mine        'main's'      " "
    'char                 '\ \'
    ```

11. What is a floating point constant? What are the different ways to represent a floating point constant?

12. What are string-literals in C++? What is the difference between character constants and string literals?

13. What is the extension of C++ program file used for running?

14. Find out the invalid identifiers among the following. Give reason for their invalidity

    a) `Principal amount`   b) `Continue`   c) `Area`

    d) `Date-of-join`   e) `9B`

15. A label in C++ is _____.

    a) Keyword   b) Identifier   c) Operator   d) Function

16. The following tokens are taken from a C++ program. Fill up the given table by placing them at the proper places

    (`int`, `cin`, `%`, `do`, `=`, `"break"`, `25.7`, `digit`)

Keywords	Identifiers	Literals	Operators

Short answer type

1. Write down the rules governing identifiers.
2. What are tokens in C++ ? List different types of tokens allowed in C++.
3. Distinguish between identifiers and keywords.
4. How are integer constants represented in C++? Explain with examples.
5. What are character constants in C++? How are they implemented?

Long answer type

1. Briefly describe different types of tokens.
2. Explain different types of literals with examples.
3. Briefly describe the Geany IDE and its important features.

5

Data Types and Operators

Key concepts

- **Concept of Data Types**
- **C++ Data Types**
- **Fundamental Data Types**
- **Variables**
- **Operators**
  - o Arithmetic
  - o Relational
  - o Logical,
  - o Input/Output
  - o Assignment
- **Expressions**
  - o Arithmetic
  - o Relational
  - o Logical
- **Statements**
  - o Declaration
  - o Assignment
  - o Input /Output

In the previous chapter we learned the basic building blocks of C++ language. As we know, data processing is the main activity carried out in computers. All programming languages give importance to data handling. The input data is arranged and stored in computers using some structures. C++ has a predefined template for storing data. The stored data is further processed using proper operators. In this chapter, we will explore the main concepts of the C++ language like data types, operators, expressions and statements in detail.

5.1 Concept of data types

Consider the case of preparing the progress card of a student after an examination. We need data like admission number, roll number, name, address, scores in different subjects, the grades obtained in each subject, etc. Further, we need to display the percentage of marks scored by the student and the attendance in percentage. If we consider a case of scientific data processing, it may require data in the form of numbers representing the velocity of light (3×10^8 m/s), acceleration due to gravity (9.8 m/s), electric charge of an electron (-1.6×10^{-19}), etc.

From the above cases, we can infer that data can be of different types like character, integer number, real number, string, etc. In the last chapter we saw that any valid character of C++ enclosed in single quotes represents character data in C++. Numbers without fractions represent integer data. Numbers with fractions represent floating point data and anything enclosed in double quotes represents a string data. Since the data to be dealt with are of many types, a programming language must provide ways and facilities to handle all types of data. C++ provides facilities to handle different types of data by providing data type names. **Data types** are the means to identify the nature of the data and the set of operations that can be performed on the data.

In Chapter 3, we used variables to refer data in algorithms. Variables are also used in programs for referencing data. When we write programs in the C++ language, variables are to be declared before their use. Data types are necessary to declare these variables.

5.2 C++ data types

C++ provides a rich set of data types. Based on nature, size and associated operations, they are classified as shown in Figure 5.1. Basically, they are classified into fundamental or built-in data types, derived data types and user-defined data types.

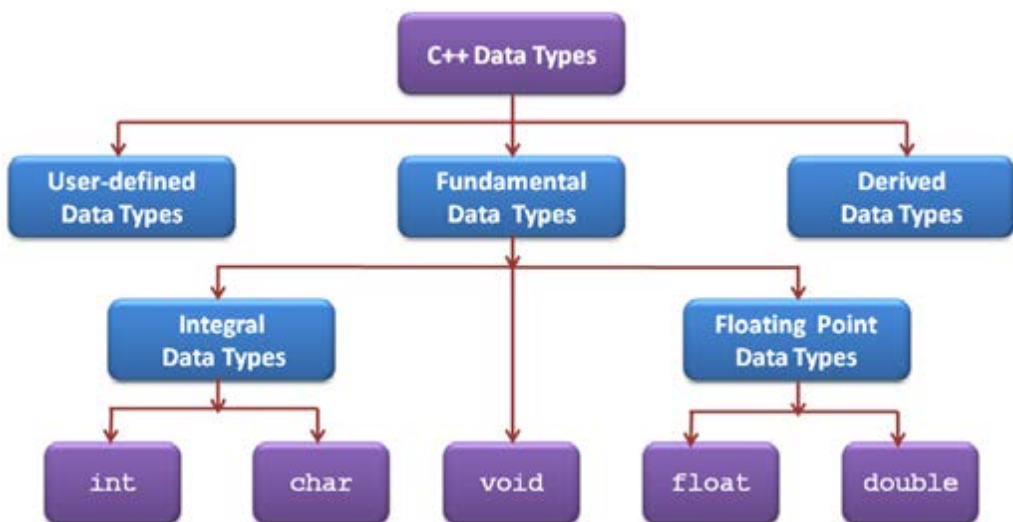
Fig. 5.1: Classification of C++ data types

Fundamental data types

Fundamental data types are defined in C++ compiler. They are also known as built-in data types. They are atomic in nature and cannot be further broken into small units. The

five fundamental data types in C++ are **char**, **int**, **float**, **double** and **void**. Among these, **int** and **char** come under integral data type as they can handle only integers. The numbers with fractions (real numbers) are generally known as floating type and are further divided into **float** and **double** based on precision and range.

User-defined data types

C++ is flexible enough to allow programmers to define their own data types. Structure (struct), enumeration (enum), union, class, etc. are examples for such data types.

Derived data types

Derived data types are constructed from fundamental data types through some grouping or alteration in the size. Arrays, pointers, functions, etc. are examples of derived data types.

5.3 Fundamental data types

Fundamental data types are basic in nature. They cannot be further broken into small units. Since these are defined in compiler, the size (memory space allocated) depends on the compiler. We use the compiler available in GCC and hence the size as well as the range of data supported by the data type are given accordingly. It may be different if you use other compilers like Turbo C++ IDE. The five fundamental data types are described below:

int data type (*for integer numbers*)

Integers are whole numbers without a fractional part. They can be positive, zero and negative. The keyword **int** represents integer numbers within a specific range. GCC allows 4 bytes of memory for integers belonging to int data type. So the range of values that can be represented by int data type is from -2147483648 to +2147483647. The data items 6900100, 0, -112, 17, -32768, +32767, etc. are examples of int data type. The numbers 2200000000 and -2147483649 do not belong to int data type as they are out of the allowed range.

char data type (*for character constants*)

Characters are the symbols covered by the character set of the C++ language. All letters, digits, special symbols, punctuations, etc. come under this category. When these characters are used as data they are considered as **char** type data in C++. We can say that the keyword char represents character literals of C++. Each char type data is allowed one byte of memory. The data items 'A', '+', '\t', '0', etc. belong to char data type. The char data type is internally treated as integers, because computer

recognises the character through its ASCII code. Character data is stored in the memory with the corresponding ASCII code. As ASCII codes are integers and need to be stored in one byte (8 bits), the range of `char` data type is from -128 to +127.

`float` data type (*for floating point numbers*)

Numbers with a fractional part are called floating point numbers. Internally, floating point numbers are stored in a manner similar to scientific notation. The number 47281.97 is expressed as 0.4728197×10^5 in scientific notation. The first part of the number, 0.4728197 is called the mantissa. The power 5 of 10 is called exponent. Computers typically use exponent form (*E notation*) to represent floating point values. In E notation, the number 47281.97 would be 0.4728197E5. The part of the number before the E is the mantissa, and the part after the E is the exponent. In C++, the keyword `float` is used to denote such numbers. GCC allows 4 bytes of memory for numbers belonging to `float` data type. The numbers of this data type has normally a precision of 7 digits.

`double` data type (*for double precision floating point numbers*)

In some cases, floating point numbers require more precision. Such numbers are represented by `double` data type. The range of numbers that can be handled by `float` type is extended by this data type, because it consumes double the size of `float` data type. In C++, it is assured that the range and precision of `double` will be at least as big as `float`. GCC reserves 8 bytes for storing a double precision value. The precision of `double` data type is generally 15 digits.

`void` data type (*for null or empty set of values*)

The data type `void` is a keyword and it indicates an empty set of data. Obviously it does not require any memory space. The use of this data type will be discussed in the next chapter. Table 5.1 shows the size and range of the fundamental data types of C++ based on GCC. The size of fundamental data types decreases in the order `double`, `float`, `int` and `char`.

Name	Description	Size	Range
char	Character	1 byte	-128 to 127
int	Integer	4 bytes	-2147483648 to +2147483647
float	Floating point number	4 bytes	$-3.4 \times 10^{+/-38}$ to $+3.4 \times 10^{+/-38}$ with approximately 7 significant digits
double	Double precision floating point number	8 bytes	$-1.7 \times 10^{+/-308}$ to $+1.7 \times 10^{+/-308}$ with approximately 15 significant digits
void	Null data	0 bytes	Empty set

Table 5.1: Data types and their characteristics

> The values listed above are only sample values to give you a general idea of how the types differ. The values for any of these entries may be different on your system

5.4 Variables

Memory locations are to be identified to refer data. **Variables** are the names given to memory locations. These are identifiers of C++ by which memory locations are referenced to store or retrieve data. The size and nature of data stored in a variable depends on the data type used to declare it. There are three important aspects for a variable.

i. Variable name

It is a symbolic name (identifier) given to the memory location through which the content of the location is referred to.

ii. Memory address

The RAM of a computer consists of collection of cells each of which can store one byte of data. Every cell (or byte) in RAM will be assigned a unique address to refer it. All the variables are connected to one or more memory locations in RAM. The base address of a variable is the starting address of the allocated memory space. In the normal situation, the address is given implicitly by the compiler. The address is also called the L-value of a variable. In Figure 5.2 the base address of the variable **Num** is 1001.

1001	1002	1003	1004
	18		

Num
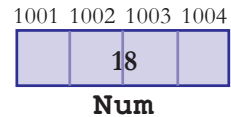
Fig 5.2 : Memory representation of a Variable

iii. Content

The value stored in the location is called content of the variable. This is also called the R-value of the variable. Type and size of the content depends on the data type of the variable.

Figure 5.2 shows the memory representaion of a variable. Here the variable name is **Num** and it consumes 4 bytes of memory at memory addresses 1001, 1002, 1003 and 1004. The content of this variable is 18. That is L-value of **Num** is 1001 and R-value is 18.

5.5 Operators

Operators are tokens constituted by predefined symbols that can trigger computer to carry out operations. The participants of an operation are called **operands**. An operand may be either a constant or a variable.

For example, a+b triggers an arithmetic operation in which **+** (addition) is the operator and a, b are operands. Operators in C++ are classified based on various criteria. Based on number of operands required for the operation, operators are classified into three. They are unary, binary and ternary.

Unary operators

A unary operator operates on a single operand. Commonly used unary operators are unary**+** (positive) and unary**−** (negative). These are used to represent the signs of a number. If we apply unary**+** operator on a signed number, the existing sign will not change. If we apply unary**−** operator on a signed number, the sign of the existing number will be negated. Examples of the use of unary operators are given in Table 5.2.

Variable	Unary +	Unary-
x	+x	-x
8	8	-8
0	0	0
-9	-9	9

Table 5.2 : Unary operators

Some other examples of unary operators are increment (**++**) and decrement (**−−**) operators. We will discuss these operators later in this chapter.

Binary operators

Binary operators operate on two operands. Arithmetic operators, relational operators, logical operators, etc. are the commonly used binary operators.

Ternary operator

Ternary operator operates on three operands. The typical example is the conditional operator (**? :**).

The operations triggered by the operators mentioned above will be discussed in detail in the coming sections and some of them will be dealt with in Chapter 7. Based on the nature of the operation, operators are classified into arithmetic, relational, logical, input/output, assignment, short-hand, increment/decrement, etc.

5.5.1 Arithmetic operators

Arithmetic operators are defined to perform basic arithmetic operations such as addition, subtraction, multiplication and division. The symbols used for this are **+**, **−**, ***** and **/** respectively. C++ also provides a special operator **%** (modulus operator) for getting remainder during division. All these operators are binary operators. Note that **+** and **−** are used as unary operators too. The operands required for these operations are numeric data. The result of these operations will also be numeric. Table 5.3 shows some examples of binary arithmetic operations.

Variable x	Variable y	Addition x + y	Subtraction x − y	Multiplication x * y	Division x / y
10	5	15	5	50	2
−11	3	−8	−14	−33	−3.66667
11	−3	8	14	−33	−3.66667
−50	−10	−60	−40	500	5

Table 5.3 : Arithmetic operators

Modulus operator (%)

The modulus operator, also called as mod operator, gives the remainder value during arithmetic division. This operator can only be applied over integer operands. Table 5.4 shows some examples of modulus operation. Note that the sign of the result is the sign of first operand. Here in the table the first operand is **x**.

Variable x	Variable y	Modulus Operation x % y	Variable x	Variable y	Modulus Operation x % y
10	5	0	100	100	0
5	10	5	32	11	10
−5	11	−5	11	−5	1
5	−11	5	−11	5	−1
−11	−5	−1	−5	−11	−5

Table 5.4 : Operations using modulus operator

Check yourself

1. Arrange the fundamental data types in ascending order of size.
2. The name given to a storage location is known as _____.
3. Name a ternary operator in C++.
4. Predict the output of the following operations if x = -5 and y = 3 initially

   a. -x   f. x+y
   b. -y   g. x%y
   c. -x + -y   h. x/y
   d. -x - y   i. x * -y
   e. x % -11   j. -x % -5

5.5.2 Relational operators

Relational operators are used for comparing numeric data. These are binary operators. The result of any relational operation will be either **True** or **False**. In C++, True is represented by **1** and False is represented by **0**. There are six relational operators in C++. They are **<** (less than), **>** (greater than), **==** (equal to), **<=** (less than or equal to), **>=** (greater than or equal to) and **!=** (not equal to). Note that equality checking requires two equal symbols (==). Some examples for the use of various relational operators and their results are shown in Table 5.5:

m	n	m<n	m>n	m<=n	m>=n	m!=n	m==n
12	5	0	1	0	1	1	0
-7	2	1	0	1	0	1	0
4	4	0	0	1	1	0	1

Table 5.5 : Operations using relational operators

5.5.3 Logical operators

Using relational operators, we can compare values. Examples are 3<5, num!=10, etc. These comparison operations are called relational expressions in C++. In some cases, two or more comparisons may need to be combined. In Mathematics we may use expressions like a>b>c. But in C++, it is not possible. We have to separate this into two, as a>b and b>c and these are to be combined using the logical operator **&&**, i.e. (a>b)&&(b>c). The result of such logical combinations will also be either True or False (i.e. 1 or 0). The logical operators are **&&** (logical AND), **||** (logical OR) and **!** (logical NOT).

Logical AND (&&) operator

If two relational expressions E1 and E2 are combined using logical AND (**&&**) operation, the result will be 1 (True) only if both E1 and E2 have values 1 (True). In all other cases the result will be 0 (False). The results of evaluation of **&&** operation for different possible combination of inputs are shown in Table 5.6

E1	E2	E1&&E2
0	0	0
0	1	0
1	0	0
1	1	1

Table 5.6 : Logical AND

Examples: `10>5 && 15<25` evaluates to 1 (True)

`10>5 && 100<25` evaluates to 0 (False)

Logical OR (| |) operator

If two relational expressions El and E2 are combined using logical OR (| |) operations, the result will be 0 (False) only if both El and E2 are having value 0 (False). In all other cases the result will be 1 (True). The results of evaluation of | | operation for different possible combination of inputs are shown in Table 5.7

E1	E2	E1 ‖ E2
0	0	0
0	1	1
1	0	1
1	1	1

Table 5.7 : Logical OR

Examples: `10>5 || 100<25` evaluates to 1 (True)

`10>15 || 100<90` evaluates to 0 (False)

Logical NOT operator (!)

This operator is used to negate the result of a relational expression. This is a unary operation. The results of evaluation of ! operation for different possible inputs are as shown in Table 5.8.

E1	!E1
0	1
1	0

Table 5.8 : Logical NOT

Examples: `!(100<2)` evaluates to 1 (True)

`!(100>2)` evaluates to 0 (False)

5.5.4 Input / Output operators

Usually input operation requires user's intervention. In the process of input operation, the data given through the keyboard is stored in a memory location. C++ provides **>>** operator for this operation. This operator is known as *get from* or *extraction* operator. This symbol is constituted by two greater than symbols.

Similarly in output operation, data is transferred from RAM to an output device. Usually the monitor is the standard output device to get the results directly. The operator **<<** is used for output operation and is called *put to* or *insertion* operator. It is constituted by two less than symbols.

5.5.5 Assignment operator (=)

When we have to store a value in a memory location, assignment operator (**=**) is used. This is a binary operator and hence two operands are required. The first operand should be a variable where the value of the second operand is to be stored. Some examples are shown in Table 5.9.

Item	Description
a=b	The value of variable **b** is stored in **a**
a=3	The constant **3** is stored in variable **a**

Table 5.9 : Assignment operator

We discussed the relational operator == in Section 5.5.2. See the difference between these two operators. The = symbol assigns a value to a variable, whereas == symbol compares two values and gives True or False as the result.

5.6 Expressions

An expression is composed of operators and operands. The operands may be either constants or variables. All expressions can be evaluated to get a result. This result is known as the value returned by the expression. On the basis of the operators used, expressions are mainly classified into arithmetic expressions, relational expressions and logical expressions.

5.6.1 Arithmetic expressions

An expression in which only arithmetic operators are used is called arithmetic expression. The operands are numeric data and they may be variables or constants. The value returned by these expressions is also numeric. Arithmetic expressions are further classified into integer expressions, floating point (real) expressions and constant expressions.

Integer expressions

If an arithmetic expression contains only integer operands, it is called integer expression and it produces integer result after performing all the operations given in the expression. For example, if x and y are integer variables, some integer expressions and their results are shown in Table 5.10.

x	y	x + y	x / y	−x + x * y	5 + x / y	x % y
5	2	7	2	5	7	1
6	3	9	2	12	7	0

Table 5.10 : Integer expressions and their results

Note that all the above expressions produce integer values as the results.

Floating point expressions (Real expressions)

An arithmetic expression that is composed of only floating point data is called floating point or real expression and it returns a floating point result after performing all the

operations given in the expression. Table 5.11 shows some real expressions and their results, assuming that x and y are floating point variables.

x	y	x + y	x / y	−x + x * y	5 + x / y	x * x / y
5.0	2.0	7.0	2.5	5.0	7.5	12.5
6.0	3.0	9.0	2.0	12.0	7.0	12.0

Table 5.11 : Floating point expressions and their results

It can be seen that all the above expressions produce floating point values as the results.

In an arithmetic expression, if all the operands are constant values, then it is called *Constant expression*. The expression 20+5/2.0 is an example. The constants like 15,3.14,'A' are also known as constant expressions.

5.6.2 Relational expressions

When relational operators are used in an expression, it is called relational expression and it produces Boolean type results like True (1) or False (0). In these expressions, the operands are numeric data. Let us see some examples of relational expressions in Table 5.12.

x	y	x > y	x == y	x+y !=y	x−2 == y+1	x*y == 6*y
5.0	2.0	1(True)	0(False)	1(True)	1 (True)	0 (False)
6	13	0(False)	0(False)	1(True)	0 (False)	1 (True)

Table 5.12 : Relational expressions and their results

We know that arithmetic operators have a higher priority than relational operators. So when arithmetic expressions are used on either side of a relational operator, arithmetic operations will be carried out first and then the results are compared. The table contains some expressions in which both arithmetic and relational operators are involved. Though they contain mixed type of operators, they are called relational expressions since the final result will be either True or False.

5.6.3 Logical expressions

Logical expressions combine two or more relational expressions with logical operators and produce either True or False as the result. A logical expression may contain constants, variables, logical operators and relational operators. Let us see some examples in Table 5.13.

x	y	x>=y && x==20	x==5||y==0	x==y && y+2==0	!(x==y)
5.0	2.0	0 (False)	1 (True)	0 (False)	1 (True)
20	13	1 (True)	0 (False)	0 (False)	1 (True)

Table 5.13: Logical expressions and their results

Check yourself

1. Predict the output of the following operations if x=5 and y=3.

    a. x>=10 && y>=4   f. x>=10 || y>=4

    b. x>=1 && y>=3   g. x>=1 || y>=3

    c. x>5 && y>=2   h. x>5 || y>=2

    d. x<10 && y>2   i. x<10 || y>2

    e. x<10 && y>x   j. x<10 || y>x

2. Predict the output if p=5, q=3, r=2

    a. p-q*r/2   c. p-q-r*2+p

    b. p*q/r   d. p+5*q+r*r/2

5.7 Statements

Can you recollect the learning hierarchy of a natural language? Alphabet, words, phrases, sentences, paragraphs and so on. In the learning process of C++ language we have covered character set, tokens and expressions. Now we have come to the stage where we start communication with the computer sensibly and meaningfully with the help of statements. **Statements** are the smallest executable unit of a programming language. C++ uses the symbol semicolon (;) as the delimiter of a statements. Different types of statements used in C++ are declaration statements, assignment statements, input statements, output statements, control statements, etc. Each statement has its own purpose in a C++ program. All these statements except declaration statements are executable statements as they possess some operations to be done by the computer. Executable statements are the instructions to the computer. The execution of control statements will be discussed in Chapter 7. Let us discuss the other statements.

5.7.1 Declaration statements

Every user-defined word should be defined in the program before it is used. We have seen that a variable is a user-defined word and it is an identifier of a memory location. It must be declared in the program before its use. When we declare a variable, we tell the compiler about the type of data that will be stored in it. The syntax of variable declaration is:

```
data_type <variable1>[, <variable2>, <variable3>, ...] ;
```

The data_type in the syntax should be any valid data type of C++. The syntax shows that when there are more than one variables in the declaration, they are separated by comma. The declaration statement ends with a semicolon. Typically, variables are declared either just before they are used or at the beginning of the program. In the syntax, everything given inside the symbols [and] are optional. The following statements are examples for variable declaration:

```
int  rollnumber;
double  wgpa,avg_score;
```

The first statement declares the variable rollnumber as **int** type so that it will be allocated four bytes of memory (as per GCC) and it can hold an integer number within the range from -2147483648 to +2147483647. The second statement defines the identifiers wgpa and avg_score as variables to hold data of **double** type. Each of them will be allocated 8 bytes of memory. The memory is allocated to the variables during the compilation of the program.

5.7.2 Assignment statements

When the assignment operator (**=**) is used to assign a value to a variable, it forms an assignment statement. It can take any of the following syntax:

```
variable = constant;
variable1 = variable2;
variable = expression;
```

The first statement assigns the constant to variable, the second assigns the value of variable2 to the variable1 and the last statement stores the result of the expression in variable. The following are some examples of assignment statements:

```
a = 15;             b = 5.8;
c = a + b;          d = (a + b)*(c + d);
```

The left hand side (LHS) of an assignment statement must be a variable. During execution, the expression at the right hand side (RHS) is evaluated first. The result is then assigned (stored) to the variable at LHS.

Assignment statement can be chained for doing multiple assignments at a time. For instance, the statement x=y=z=13; assigns the value 13 in three variables in the order of z, y and x. The variables should be declared before this assignment. If we assign a value to a variable, the previous value in it, if any, will be replaced by the new value.

5.7.3 Input statements

Input statement is a means that allows the user to store data in the memory during the execution of the program. We saw that the **get from** or **extraction** operator (**>>**) specifies the input operation. The operands required for this operator are the input device and a location in RAM where the data is to be stored. Keyboard being a standard console device, the stream (sequence) of data is extracted from the keyboard and stored in memory locations identified by variables. Since C++ is an object oriented language, keyboard is considered as the standard input stream device and is identified as an object by the name **cin** (pronounced as 'see in'). The simplest form of an input statement is:

```
streamobject >> variable;
```

Since we use keyboard as the input device, the streamobject in the syntax will be substituted by cin. The operand after the **>>** operator should strictly be a variable. For example, the following statement reads data from the keyboard and stores in the variable **num**.

```
cin >> num;
```

Figure 5.3 shows how data is extracted from keyboard and stored in the variable.

Fig 5.3: Input procedure in C++

5.7.4 Output statements

Output statements make the results available to the users through any output device. The **put to** or **insertion** operator (**<<**) is used to specify this operation. The operands in this case are the output device and the data for the output. The syntax of an output statement is:

```
streamobject << data;
```

The `streamobject` may be any output device and the data may be a constant, a variable or an expression. We use monitor as the commonly used output device and C++ identifies it as an object by the name `cout` (pronounced as 'see out'). The following are some examples of output statement with monitor as the output device:

```
cout << num;
cout << "hello friends";
cout << num+12;
```

The first statement displays the content in the variable **num**. The second statement displays the string constant `"hello friends"` and the last statement shows the value returned by the expression `num+12` (assuming that `num` contains numeric value). Figure 5.4 shows how data is inserted into the output stream object (monitor) from the memory location `num`.
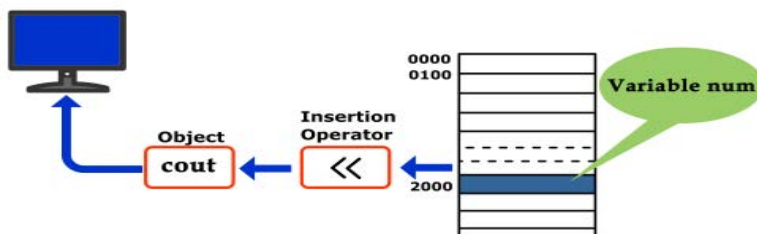

Fig. 5.4: Output procedure in C++

The tokens `cin` and `cout` are not keywords. They are predefined words that are not part of the core C++ language, and you are allowed to redefine them. They are defined in libraries required by the C++ language standard. Needless to say, using a predefined identifier for anything other than its standard meaning can be confusing and dangerous and such practices should be avoided. The safest and easiest practice is to treat all predefined identifiers as if they were keywords.

Cascading of I/O operators

Suppose you want to input three values to different variables, say x, y and z. You may use the following statements:

```
cin>>x;
cin>>y;
cin>>z;
```

But these three statements can be combined to form a single statement as given below:

```
cin>>x>>y>>z;
```

The multiple use of input or output operators in a single statement is called **cascading of I/O operators**. In the use of cascading of input operators, the values input are assigned to the variables from left to right. In the example cin>>x>>y>>z; the first value is assigned to x, the second to y and the third to z. While entering values to the variables x, y and z during execution the values should be separated by Space bar, Tab or Carriage return.

Similarly, if you want to display the contents of different variables (say x, y, z) use the following statement:

```
cout<<x<<y<<z;
```

If variables, constants and expressions appear together for output operations, the above technique can be applied as in the following example:

```
cout<<"The number is "<<z;
```

While cascading output operators, the values for the output will be retrieved from right to left. It is to be noted that both **<<** and **>>** operators cannot be used in a single statement.

Let us sum up

Data types are means to identify the type of data and associated operations handling it. Each data type has a specific size and a range of data. Data types are used to declare variables. Different types of operators are available in C++ for various operations. When operators are combined with operands (data), expressions are formed. There are mainly three types of expressions - arithmetic, relational and logical. Statements are the smallest executable unit of a program. Variable declaration statements define the variables in the program and they will be allocated memory space. The executable statements like assignment statements, input statements, output statements, etc. help giving instructions to the computer.

Learning outcomes

After the completion of this chapter the learner will be able to

• identify the various data types in C++.

• choose appropriate variables.

• experiment with various operators.

• apply the various I/O operators.

• write various expressions and statements.

Sample questions

Very short answer type

1. What are data types? List all predefined data types in C++?

2. What is the use of void data type?

3. What is a constant?

4. What is dynamic initialisation of variables?

5. Define operators.

6. What is meant by a unary operation?

7. What is declaration statement?

8. What is the input operator ">>" and output operator "<<" called ?

9. What will be the result of a = 5/3 if a is (i) float (ii) int ?

10. Write an expression that uses a relational operator to return true if the variable *total* is greater than or equal to *final*.

11. Given that i =4, j =5, k =2 what will be the result of following expressions?

    (i) j <k   (ii) j <j   (iii) j <= k   (iv) i == j   (v) i == k

    (vi) j > k   (vii) j >= i   (viii) j! = i   (ix) j! = k   (x) j <= k

12. What will be the order of evaluation for following expressions?

    (i)   i+5>=j-6   (ii)   s+10<p-2+2*q

15. What will be the result of the following if *ans* is 6 initially?

    (i) cout <<ans = 8 ;   (ii) cout << ans == 8

Short answer type

1. What is a variable? List the two values associated with a variable.

2. Explain logical operators.

3. Find out the errors, if any, in the following C++ statements

   (i) cout<< "a=" a;   (v) cin >> "\n" >> y ;

   (ii) m=5, n=l2;015   (vi) cout >> \n "abc"

   (iii) cout << "x" ; <<x;   (vii) a = b + c

   (iv) cin >> y   (viii) break = x

4. What is the role of relational operators? Distinguish between == and =.

Long answer type

1. Explain the operators used in C++ in detail.

2. Explain different types of expressions in C++.

Introduction to Programming

Key concepts

- **Structure of C++ Program**
  - ○ Pre-processor directives
  - ○ Header files
  - ○ A sample C++ program
- **Stylistic guidelines for variables and comments**
- **Variable initialization**
- **Use of const access modifier**
- **Type modifiers**
- **More operators**
  - ○ Arithmetic assignment
  - ○ Increment and decrement
  - ○ Precedence of operators
- **Type conversions**
  - ○ Implicit type conversion
  - ○ Explicit type conversion

We have familiarized ourselves with the IDE used for C++ program. In the last chapter we discussed how operations can be expressed in C++. We also learnt different types of statements in C++ to give instructions to the computer. We are now in a position to solve simple problems by using these statements. But a set of statements alone does not constitute a program. A C++ program has a typical structure. In this chapter we discuss the structure of C++ programs and we write programs accordingly. More operators are introduced for making the programs compact and the execution faster.

6.1 Structure of a C++ program

C++ program is a collection of one or more functions. A function means a set of instructions to perform a particular task referred to by a name. Since there can be many functions in a C++ program, they are usually identified by unique names. The most essential function needed for every C++ program is the `main()` function. Let us look at the structure of a simple C++ program.

```
#include <headerfile>
using namespace identifier;
int main()
{
    statements;
        :
        :
        :
    return 0;
}
```

The first line is called preprocessor directive and the second line is the namespace statement. The third line is the function header which is followed by a set of statements enclosed by a pair of braces. Let us discuss each of these parts of the program.

6.1.1 Preprocessor directives

A C++ program starts with pre-processor directives. **Preprocessors** are the compiler directive statements which give instruction to the compiler to process the information provided before actual compilation starts. Preprocessor directives are lines included in the code that are not program statements. These lines always start with a **#** (hash) symbol. The preprocessor directive #include is used to link the header files available in the C++ library by which the facilities required in the program can be obtained. No semicolon (;) is needed at the end of such lines. Separate #include statements should be used for different header files. There are some other preprocessor directives such as #define, #undef, etc.

6.1.2 Header files

Header files contain the information about functions, objects and predefined derived data types and are available along with compiler. There are a number of such files to support C++ programs and they are kept in the standard library. Whichever program requires the support of any of these resources, the concerned header file is to be included. For example, if we want to use the predefined objects cin and cout, we have to use the following statement at the beginning of the program:

```
#include <iostream>
```

The header file iostream contains the information about the objects cin and cout. Eventhough header files have the extension **.h**, it should not be specified for GCC. But the extension is essential for some other compilers like Turbo C++ IDE.

6.1.3 Concept of namespace

A program cannot have the same name for more than one identifier (variables or functions) in the same scope. For example, in our home two or more persons (or even living

beings) will not have the same name. If there are, it will surely make conflicts in the identity within the home. So, within the scope of our home, a name should be unique. But our neighbouring home may have a person (or any living being) with the same name as that of one of us. It will not make any confusion of identity within the respective scopes. But an outsider cannot access a particular person by simply using the name; but the house name is also to be mentioned.

The concept of namespace is similar to a house name. Different identifiers are associated to a particular namespace. It is actually a group name in which each item is unique in its name. User is allowed to create own namespaces for variables and functions. We can use an identifier to give a name to the namespace. The keyword **using** technically tells the compiler about a namespace where it should search for the elements used in the program. In C++, **std** is an abbreviation of 'standard' and it is the standard namespace in which cout, cin and a lot of other objects are defined. So, when we want to use them in a program, we need to follow the format std::cout and std::cin. This kind of explicit referencing can be avoided with the statement using namespace std; in the program. In such a case, the compiler searches this namespace for the elements cin, cout, endl, etc. So whenever the computer comes across cin, cout, endl or anything of that matter in the program, it will read it as std::cout, std::cin or std::endl.

The statement **using namespace std;** doesn't really add a function, it is the #include <iostream> that "loads" cin, cout, endl and all the like.

6.1.4 The `main()` function

Every C++ program consists of a function named main(). The execution starts at main() and ends within main(). If we use any other function in the program, it is called (or invoked) from main(). Usually a data type precedes the main() and in GCC, it should be int.

The function header main() is followed by its body, which is a set of one or more statements within a pair of braces { }. This structure is known as the definition of the main() function. Each statement is delimited by a semicolon (;). The statements may be executable and non-executable. The executable statements represent instructions to be carried out by the computer. The non-executable statements are intended for compiler or programmer. They are informative statements. The last statement in the body of main() is return 0;. Even though we do not use this statement, it will not make any error. Its relavance will be discussed in Class XII.

C++ is a free form language in the sense that it is not necessary to write each statement in new lines. Also a single statement can take more than one line.

6.1.5 A sample program

Let us look at a complete program and familiarise ourselves with its features, in detail. This program on execution will display a text on the screen.

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Hello, Welcome to C++";
    return 0;
}
```

The program has seven lines as detailed below:

Line 1: The preprocessor directive #include is used to link the header file iostream with the program.

Line 2: The using namespace statement makes available the identifier cout in the program.

Line 3: The header of the essential function for a C++ program, i.e., int main().

Line 4: An opening brace { that marks the beginning of the instruction set (program).

Line 5: An output statement, which will be executed when we run the program, to display the text "Hello, Welcome to C++" on the monitor. The header file iostream is included in this program to use cout in this statement.

Line 6: The return statement stops the execution of the main() function. This statement is optional as far as main() is concerned.

Line 7: A closing brace } that marks the end of the program.

6.2 Guidelines for coding

A source code looks good when the coding is legible, logic is communicative and errors if any are easily detectable. These features can be experienced if certain styles are followed while writing program. Some guidelines are discussed below to write stylistic programs:

Use suitable naming convention for identifiers

Suppose we have to calculate the salary for an employee after deductions. We may code it as: A = B - C;
where A is the net salary, B the total salary and C total deduction. The variable names A, B and C do not reflect the quantities they denote. If the same instruction is expressed as follows, it would be better:

```
Net_salary = Gross_salary - Deduction;
```

The variable names used in this case help us to remember the quantity they possess. They readily reflect their purpose. This kind of identifiers are called *mnemonic names*. The points given below are to be remembered in the choice of names:

- Choose good mnemonic names for all variables, functions and procedures.

    e.g. `avg_hgt, Roll_No, emp_code, SumOfDigits`, etc.

- Use standardised prefixes and suffixes for related variables.

    e.g. `num1, num2, num3` for three numbers

- Assign names to constants in the beginning of the program.

    e.g. `float PI = 3.14;`

Use clear and simple expressions

Some people have a tendency to reduce the execution time by sacrificing simplicity. This should be avoided. Consider the following example. To find out the remainder after division of `x` by `n`, we can code as: `y = x-(x/n)*n;`

The same thing is achieved by a simpler and more elegant piece of code as shown below:

    `y = x % n;`

So it is better to use simpler codes in programming to make the program more simple and clear.

Use comments wherever needed

Comments play a very important role as they provide internal documentation of a program. They are lines in code that are added to describe the program. They are ignored by the compiler. There are two ways to write comments in C++:

Single line comment: The characters `//` (two slashes) is used to write single line comments. The text appearing after `//` in a line is treated as a comment by the C++ compiler.

Multiline comments: Anything written within `/*` and `*/` is treated as comment so that the comment can take any number of lines.

But care should be taken that no relevant code of the program is included accidently inside the comment. The following points are to be noted while commenting:

- Always insert prologues, the comments in the beginning of a program that summarises the purpose of the program.
- Comment each variable and constant declaration.
- Use comments to explain complex program steps.

- It is better to include comments while writing the program itself.
- Write short and clear comments.

Relevance of indentation

In computer programming, an indent style is a convention governing the indentation of blocks of code to convey the program's structure, for good visibility and better clarity. An indentation makes the statements clear and readable. It shows the levels of statements in the program.

The usages of these guidelines can be observed in the programs given in a later section of this chapter.

6.3 Variable initialisation

In the last chapter we discussed the significance of variables in programs. A variable is associated with two values: L-value (its address) and R-value (its content). When a variable is declared, a memory location with an address will be allocated and it is the L-value of the variable. What will its content be? It is not blank or 0 or space! If the variable is declared with `int` data type, the content or the R-value will be any integer within the allowed range. But this number cannot be predicted or will not always be the same. So we call it *garbage value*. When we store a value into the variable, the existing content will be replaced by the new one. The value can be stored in the variable either at the time of compilation or execution. Supplying value to a variable at the time of its declaration is called **variable initialisation**. This value will be stored in the respective memory location during compile-time. The assignment operator (=) is used for this. It can be done in two ways as given below:

```
data_type variable = value;
```

OR `data_type  variable(value);`

The statements: `int xyz=120;` and `int xyz(120);` are examples of variable initialisation statements. Both of these statements declare an integer variable `xyz` and store the value 120 in it as shown in Figure 6.1.

More examples are:
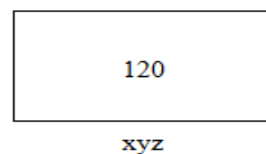
```
float val=0.12, b=5.234;
char k='A';
```


Fig 6.1: Variable initialisation

A variable can also be initialised during the execution of the program and is known as dynamic initialisation. This is done by assigning an expression to a variable as shown in the following statements:

```
float product =  x * y;
float interest= p*n*r/100.0;
```

In the first statement, the variable `product` is initialised with the product of the values stored in `x` and `y` at runtime. In the second case, the expression `p*n*r/100.0` is evaluated and the value returned by it will be stored in the variable `interest` at runtime.

Note that during dynamic initialisation, the variables included in the expression at the right of assignment operator should have valid data, otherwise it will produce unexpected results.

6.4 `const` - The access modifier

It is a good practice to use symbolic constants rather than using numeric constants directly. For example, we can use symbolic name like **pi** instead of using `22.0/7.0` or 3.14. The keyword **const** is used to create such symbolic constants whose value can never be changed during execution.

Consider the statement: `float pi=3.14;`

The floating point variable `pi` is initialised with the value 3.14. The content of `pi` can be changed during the execution of the program. But if we modify the declaration as:

```
const float pi=3.14;
```

the value of `pi` remains constant (unaltered) throughout the execution of the program. The read/write accessibility of the variable is modified as read only. Thus, the `const` acts as an access modifier.

> During software development, larger programs are developed using collaborative effort. Several people may work together on different portions of the same program. They may share the same variable. In these situations, there may be occasions where one may modify the content of the variable which will adversely affect other person's coding. In these situations we have to keep the content of variables unaffected by the activity of others. This can be done by using **const**.

6.5 Type modifiers

Have you ever seen travel bags that can alter its size/volume to include extra bit of luggage? Usually we don't use that extra space. But the zipper attached with the bag helps us to alter its volume either by increasing it or by decreasing. In C++ too, we need data types that can accommodate data of slightly bigger/smaller size. C++ provides data **type modifiers** which help us to alter the size, range or precision. Modifiers precede the data type name in the variable declaration. It alters the range of values

permitted to a data type by altering the memory size and/or sign of values. Important modifiers are **signed**, **unsigned**, **long** and **short**.

The exact sizes of these data types depend on the compiler and computer you are using. It is guaranteed that:

- a `double` is at least as big as a `float`.
- a `long double` is at least as big as a `double`.

Each type and their modifiers are listed in Table 6.1 (based on GCC compiler) with their features.

Name	Description	Size	Range
char	Character	1 byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer	2 bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer	4 bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer	4 bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
float	Floating point number	4 bytes	$-3.4 \times 10^{+/-38}$ to $+3.4 \times 10^{+/-38}$ with approximately 7 significant digits
double	Double precision floating point number	8 bytes	$-1.7 \times 10^{+/-308}$ to $+1.7 \times 10^{+/-308}$ with approximately 15 significant digits
long double	Long double precision floating point number	10 bytes	$-3.4 \times 10^{+/-4932}$ to $+3.4 \times 10^{+/-4932}$ With approximately 19 significant digits

Table 6.1 : Data type and type modifiers

> The values listed above are only sample values to give you a general idea of how the types differ. The values for any of these entries may be different on your system.

6.6 More Operators

We discussed the operators in C++ for the commonly used operations. There are some special operators in C++ which make programs more condensed. They combine two operations together. We will discuss a few of them which combine assignment and

arithmetic operations into a single operator. Arithmetic assignment, increment and decrement operators are some examples.

6.6.1 Arithmetic assignment operators

A simple arithmetic statement can be expressed in a more condensed form using arithmetic assignment operators. For example, `a=a+10` can be represented as `a+=10`. Here **+=** is an arithmetic assignment operator. This method is applicable to all arithmetic operators and they are shown in Table 6.2. The arithmetic assignment operators in C++ are **+=, −=, *=, /=, %=**. These are also known as C++ short-hands. These are all binary operators and the first operand should be a variable. The use of these operators makes the two operations (arithmetic and assignment) faster than the usual method.

Arithmetic assigment operation	Equivalent arithmetic operation
x += 10	x = x + 10
x -= 10	x = x - 10
x *= 10	x = x * 10
x /= 10	x = x / 10
x %= 10	x = x % 10

Table 6.2: C++ short hands

6.6.2 Increment (++) and Decrement (−−) operators

Increment and decrement operators are two special operators in C++. These are unary operators and the operand should be a variable. These operators help keeping the source code compact.

Increment operator (**++**)

This operator is used for incrementing the content of an integer variable by one. This can be written in two ways `++x` (pre increment) and `x++` (post increment). Both are equivalent to `x=x+1` as well as `x+=1`.

Decrement operator (**−−**)

As a counterpart of increment operator, there is a decrement operator which decrements the content of an integer variable by one. This operator is also used in two ways `−−x` (pre decrement) and `x−−` (post decrement). These are equivalent to `x=x-1` and `x-=1`.

The two usages of these operators are called prefix form and postfix form of increment/decrement operation. Both the forms make the same effect on the operand variable, but the mode of operation will be different when these are used with other operators.

Prefix form of increment/decrement operators

In the prefix form, the operator is placed before the operand and the increment/decrement operation is carried out first. The incremented/decremented value is used for the other operation. So, this method is often called ***change, then use*** method.

Consider the variables a, b, c and d with values a=10, b=5. If an operation specified as c=++a, the value of a will be 11 and that of c will also be 11. Here the value of a is incremented by 1 at first and then the changed value of a is assigned to c. That is why both the variables get the same value. Similarly, after the execution of d=--b the value of d and b will be 4.

Postfix form of increment/decrement operators

When increment/decrement operation performed in postfix form, the operator is placed after the operand. The current value of the variable is used for the remaining operation and after that the increment/decrement operation is carried out. So, this method is often called *use, then change* method.

Consider the same variables used above with the same initial values. After the operation performed with c=a++, the value of a will be 11, but that of c will be 10. Here the value of a is assigned to c at first and then a is incremented by 1. That is, before changing the value of a it is used to assign to c. Similarly, after the execution of d=b-- the value of d will be 5 but that of b will be 4.

6.6.3 Precedence of operators

Let us consider the case where different operators are used with the required operands. We should know in which order the operations will be carried out. C++ gives priority to the operators for execution. During execution, a pair of parentheses is given the first priority. If the expression is not parenthesised, it is evaluated according to some precedence order. The order is given in Table 6.3. In an expression, if the operations of same priority level occur, the precedence of execution will be from left to right in most of the cases.

Consider the variables with the values: a=3, b=5, c=4, d=2, x

After the operations specified in the following:

        x = a + b * c - d;

the value in x will be 21. Here * (multiplication) has higher priority than + (addition) and − (subtraction). Therefore the variables b and c are multiplied, then that result is added to a. From that result, d is subtracted to get the final result.

It is important to note that the operator priority can be changed in an expression as per the need of the programmer by using parentheses ( ).

For example, if a=5, b=4, c=3, d=2 then the result of a+b−c*d will be 3. Suppose the programmer wants to perform subtraction first, then addition and multiplication, you need to use proper parentheses as (a+(b−c))*d. Now the output will be 12. For changing operator priority, brackets [ ] and braces { } cannot be used.

Priority	Operations
1	( ) parentheses
2	++, −−, ! , Unary+ , Unary −, sizeof
3	* (multiplication), / (division), **%** (Modulus)
4	+ (addition), − (subtraction)
5	< (less than), <= (less than or equal to), > (greater than), >= (greater than or equal to)
6	== (equal to), ! = (not equal to)
7	&& (logical AND)
8	|| (logical OR)
9	? : (Conditional expression)
10	= (Assignment operator), *=, /=, %=, +=, −= (arithmetic assignment operators)
11	, (Comma)

Table 6.3: Precedence of operators

6.7 Type conversion

We discussed in the last chapter that arithmetic operations are represented by arithmetic expressions, which may be either integer expression or real expressions. In both cases, the operands involved in the arithmetic operation are of the same data type. But there are situations where different types of numeric data may be involved. For example in C++, the integer expression 5 /2 gives 2 and the real expression 5.0/2.0 gives 2.5. But what will the result of 5/2.0 or 5.0/2 be? Conversion techniques are applied in such situations. The data type of one operand will be converted to another. It is called **type conversion** and can be done in two ways: implicitly and explicitly.

6.7.1 Implicit type conversion (Type promotion)

Implicit type conversion is performed by C++ compiler internally. In expressions where different types of data are involved, C++ converts the lower sized operands to the data type of highest sized operand. Since the conversion is always from lower type to higher, it is also known as **type promotion**. Data types in the decreasing order of size are as follows: `long double`, `double`, `float`, `unsigned long`, `long int`, `unsigned int`, `int` and `short int`. The type of the result will also be the type of the highest sized operand.

For example, consider the expression 5 / 2 * 3 + 2.5 which gives the result 8.5. The evaluation steps are as follows:

Step 1:   5 / 2 → 2   (Integer division)

Step 2:   2 * 3 → 6   (Integer multiplication)

Step 3:   6 + 2.5 → 8.5   (Floating point addition, 6 is converted into 6.0)

6.7.2 Explicit type conversion (Type casting)

Unlike implicit type conversion, sometimes the programmer may decide the data type of the result of evaluation. This is done by the programmer by specifying the data type within parentheses to the left of the operand. Since the programmer explicitly casts a data to the desired type, it is also known as explicit type conversion or **type casting**. Usually, type casting is applied on the variables in the expressions. The data type compatibility is to be seriously considered in assignment statements where expressions are involved.

Type compatibility in assignment statement

During the execution of an assignment statement, if the data type of the RHS expression is different from that of the LHS variable, there are two possibilities.

- The size of the data type of the variable at LHS is higher than that of the variable or expression at RHS. In this case data type of the value at RHS is promoted (type promotion) to that of the variable at LHS. Consider the following code snippet:

```
int a=5, b=2;
float p, q;
p = b;
q = a / p;
```

Here the data type of b is promoted to float and 2.0 is stored in p. So when the expression a/p is evaluated, the result will be 2.5 due to the type promotion of a. So, q will be assigned with 2.5.

- The second possibility is that the size of the data type of LHS variable is smaller than the size of RHS value. In this case, the higher order bits of the result will be truncated to fit in the variable location of LHS. The following code illustrates this.

```
float a=2.6;
int p, q;
p = a;
q = a * 4;
```

Here the value of p will be 2 and that of q will be 10. The expression a*4 is evaluated to 10.4, but q being int type it will hold only 10.

Programmer can apply the explicit conversion technique to get the desired results when there are mismatches in the data types of operands. Consider the following code segment.

```
int p=5, q=2;
float x, y;
x = p/q;
y = (x+p)/q;
```

After executing the above code, the value of x will be 2.0 and that of y will be 3.5. The expression p/q being an integer expression gives 2 as the result and is stored in x as floating point value. In the last statement, the pair of parentheses gives priority to x+p and the result will be 7.0 due to the type promotion of p. Then the result 7.0 will be the first operand for the division operation and hence the result will be 3.5 since q is converted into float. If we have to get the floating point result from p/q to store in x, the statement should be modified as x=(float)p/q; or x=p/(float)q; by applying type casting.

Program Gallery

Let us write programs to solve some problems following the guidelines. Program 6.1 displays a message.

Program 6.1: To display a message

```
/* This is a program which displays the message
   smoking is injurious to health          Multiline comment
   on the monitor */
#include <iostream>   // To use the cout object
using namespace std; // To access the cout object
int main() //program begins here    Single line comment
{   //The following statement displays the message
    cout << "Smoking is injurious to health";
    return 0;
}   //end of the program
```

Indentation   On executing Program 6.1, the output will be as follows:

```
Smoking is injurious to health
```

More illustrations on the usage of indentation can be seen in the programs given in Chapter 7. Program 6.2 accepts two numbers from user, finds their sum and displays the result. It uses all types of statements that we discussed in Chapter 5.

Program 6.2: To find the sum of two numbers

```cpp
#include <iostream>
using namespace std;
int main()
{    //Program begins
     /* Two variables are declared to read user inputs
     and the variable sum is declared to store the result  */
     int num1, num2, sum;
     cout<<"Enter two numbers: ";//Used as prompt for input
     cin>>num1>>num2; //Cascading facility to get two numbers
     sum=num1+num2;  //Assignment statement to store the sum
     /* The result is displayed with proper message.
        Cascading of output operator is utilized       */
     cout<<"Sum of the entered numbers = "<<sum;
     return 0;
}
```

A sample output of Program 6.2 is given below:

```
Enter two numbers: 5    7
Sum of the entered numbers = 12
```

User inputs separated by spaces

Run Program 6.2 with some other inputs like 5.2 and 2.7 and check whether you get correct answer. If not, discuss with your friends to identify the problem and find solution.

Let us do

Let's consider another problem. A student is awarded the scores in three Continuous Evaluation activities. The maximum score of an activity is 20. Find the average score of the student.

Program 6.3: To find the average of three CE scores

```cpp
#include <iostream>
using namespace std;
int main()
{
  int score_1, score_2, score_3;
  float avg;//Average of 3 numbers may be a floating point
  cout << "Enter the three scores: ";
  cin >> score_1 >> score_2 >> score_3;
```

```
  avg = (score_1 + score_2 + score_3) / 3.0;
  /* The result of addition will be an integer value.
    If 3 is written instead of 3.0, integer division
    will take place and result will not be correct  */
  cout << "Average CE score is: " << avg;
  return 0;
}
```

Program 6.3 gives the following output for the scores given below:

```
Enter the three scores: 17   19   20
Average CE score is: 18.666666
```

The assignment statement to find the average value uses an expression to the right of assignment operator (**=**). This expression has two **+** operators and one **/** operator. The precedence of **/** over **+** is changed by using parentheses for addition. The operands for the addition operators are all `int` type data and hence the result will be an integer. When this integer result is divided by 3, the output will again be an integer. If it was so, the output of Program 6.3 would have been 18, which is not accurate. Hence floating point constant 3.0 is used as the second operand for **/** operator. It makes the integer numerator `float` by type promotion.

Suppose the radius of a circle is given and you have to calculate its area and perimeter. As you know area of a circle is calculated using the formula πr^2 and perimeter by $2\pi r$, where $\pi = 3.14$. Program 6.4 solves this problem.

Program 6.4: To find the area and perimeter of a circle

```
#include <iostream>
using namespace std;
int main()
{
  const float PI = 22.0/7;//Use of const access modifier
  float radius, area, perimeter;
  cout<<"Enter the radius of a circle: ";
  cin>>radius;
  area = PI * radius * radius;
  perimeter = 2 * PI * radius;
  cout<<"Area of the circle = "<<area<< "\n";
  cout<<"Perimeter of the circle = "<<perimeter;
  return 0;
}
```

Escape sequence '\n' prints a new line after displaying the value of area

A sample output of Program 6.4 is shown below:

```
Enter the radius of a circle: 2.5
Area of the circle = 19.642857
Perimeter of the circle = 15.714285
```

The last two output statements of Program 6.4 display both the results in separate lines. The escape sequence character `'\n'` brings the cursor to the new line before the last output statement gets executed.

Let's develop yet another program to find simple interest. As you know, principal amount, rate of interest and period are to be input to get the result.

Program 6.5: To find the simple interest

```cpp
#include <iostream>
using namespace std;
int main()
{
  float p_Amount, n_Year, i_Rate, int_Amount;
  cout<<"Enter the principal amount in Rupees: ";
  cin>>p_Amount;
  cout<<"Enter the number of years of deposit: ";
  cin>>n_Year;
  cout<<"Enter the rate of interest in percentage: ";
  cin>>i_Rate;
  int_Amount = p_Amount * n_Year * i_Rate /100;
  cout << "Simple interest for the principal amount "
       << p_Amount
       << " Rupees for a period of " << n_Year
       << " years at the rate of interest " << i_rate
       << " is " << int_Amount << " Rupees";
  return 0;
}
```

A sample output of Program 6.5 is given below:

```
Enter the principal amount in Rupees: 100
Enter the number of years of deposit: 2
Enter the rate of interest in percentage: 10
Simple interest for the principal amount 100 Rupees for a
period of 2 years at the rate of interest 10 is 20 Rupees
```

The last statement in Program 6.5 is an output statement and it spans over five lines. It is a single statement. Note that there is no semicolon at the end of each line. On execution the result may be displayed in two lines depending on the size and resolution of the monitor of your computer.

Program 6.6 solves a conversion problem. The temperature in degree Celsius will be the input and the output will be its equivalent in Fahrenheit.

Program 6.6: To convert temperature from Celsius to Fahrenheit

```
#include <iostream>
using namespace std;
int main()
{
    float celsius, fahrenheit;
    cout<<"Enter the Temperature in Celsius: ";
    cin>>celsius;
    fahrenheit=1.8*celsius+32;
    cout<<celsius<<" Degree Celsius = "
        <<fahrenheit<<" Degree Fahrenheit";
    return 0;
}
```

Program 6.6 gives a sample output as follows:

```
Enter the Temperature in Celsius: 37
37 Degree Celsius = 98.599998 Degree Fahrenheit
```

We know that each character literal in C++ has a unique value called its ASCII code. These codes are integers. Let us write a program to find the ASCII code of a given character.

Program 6.7: To find the ASCII code of a character

```
#include <iostream>
using namespace std;
int main( )
{
    char ch;
    int asc;
    cout << "Enter the character: ";
    cin >> ch;
    asc = ch;
    cout << "ASCII of "<<ch<<"  = " << asc;
    return 0;
}
```

A sample output of Program 6.7 is given below:

```
Enter the character A
ASCII of A   = 65
```

Let us sum up

C++ program has a typical structure and it must be followed while writing programs. Stylistic guidelines shall be followed to make the program attractive and communicative among humans. While declaring variables, they can be initialised with user supplied values. The value thus assigned cannot be changed if we use const access modifier in the variable initialization statement. Type modifiers help handling a higher range of data and are used with data types to declare variables. There are some special operators in C++ for performing arithmetic and assignment operations together. Type conversion methods are available in C++ and are used to get desired results from arithmetic expressions.

Learning outcomes

After the completion of this chapter the learner will be able to

- list and choose appropriate data type modifiers.
- experiment with various operators.
- apply the various I/O operators.
- identify the structure of a simple C++ program.
- identify the need for stylistic guidelines while writing a program.
- write simple programs using C++ IDE.

Lab activity

1. Write a program that asks the user to enter the weight in grams, and then display the equivalent in Kilogram.

2. Write a program to generate the following table

2013	100%
2012	99.9%
2011	95.5%
2010	90.81%
2009	85%

Use a single cout statement for output. (Hint: Make use of \n and \t)

3. Write a short program that asks for your height in Meter and Centimeter and converts it to Feet and inches. (1 foot = 12 inches, 1 inch = 2.54 cm).

4. Write a program to find area of a triangle.

5. Write a program to compute simple interest and compound interest.

6. Write a program to : (i) print ASCII code for a given digit. (ii) print ASCII code for backspace. (Hint : Store escape sequence for backspace in an integer variable).

7. Write a program to accept a time in seconds and convert into hrs: mins: secs format. For example, if 3700 seconds is the input, the output should be 1hr: 1 min: 40 secs.

Sample questions

Very short answer type

1. What is dynamic initialisation of variables?

2. What is type promotion?

3. What do you mean by type casting? What is type cast operator?

Short answer type

1. What type of variable declaration is used in the following program code ?

```
{ int area, length = 10, width = 12, perimeter;
  area = length * width;
  perimeter = 2*(length + width);
}
```

2. Modify the above program code to have dynamic initialisation for variables area and perimeter.

3. Explain type modifiers in C++.

4. Why are so many data types provided in C++?

5. What is the role of the keyword 'const'?

6. Explain pre increment and post increment operations.

7. Explain the methods of type conversions.

8. If the file iostream.h is not included in a program, what happens?

9. What would happen if main() is not present in a program?

10. Identify the errors in the following code segments

    i.
```
int main ( ) { cout << "Enter two numbers"
cin >> num >> auto
float area = Length * breadth ; }
```

    ii.
```
#include <iostream>
using namespace;
int Main()
{ int a, b
  cin <<a >>b
  max = a % b
  cout > max
}
```

11. Comments are a useful and an easy way to enhance readability and understand-ability of a program. Justify this statement with examples.

Long answer type

1. Explain different types of expressions in C++ and different methods of type conversions in detail.

2. Write the working of an assignment operator? Explain all arithmetic assignment operators with the help of examples.

7

Control Statements

In the previous chapters we discussed some executable statements of C++ to perform operations such as input, output and assignment. We know how to write simple programs. The execution of these programs is sequential in nature. That is the statements constituting the program are executed one by one. In this chapter, we discuss C++ statements used for altering the default flow of execution. As we discussed in Chapter 3, selection, skipping or repeated execution of some statements may be required for solving problems. Usually this decision will be based on some condition(s). C++ provides statements to facilitate this requirement with the help of control statements. These statements are used for altering the normal flow of program execution. Control statements are classified into two: (i) decision making/selection statements and (ii) iteration statements. Let us discuss these statements, their syntax and mode of execution.

7.1 Decision making statements

At times it so happens that the computer may not execute all statements while solving problems. Some statements may be executed in a situation, while they may not be executed

in other situations. The computer has to take the required decision in this respect. For this, we have to provide appropriate conditions which will be evaluated by the computer. It will then take a decision on the basis of the result. The decision will be in the form of selecting a particular statement for execution or skipping some statements from being executed. The statements provided by C++ for the selected execution are called **decision making statements** or **selection statements**. **if** and **switch** are the two types of selection statements in C++.

7.1.1 `if` statement

The if statement is used to select a set of statements for execution based on a condition. In C++, conditions, otherwise known as test expressions are provided by relational or logical expressions. The syntax (general form) of if statement is as follows:

```
if (test expression)
{
    statement block;
}
```

Body of if statement that consists of the statement(s) to be executed when the condition is True

Here the test expression represents a condition which is either a relational expression or logical expression. If the test expression evaluates to True (non-zero value), a statement or a block of statements associated with **if** is executed. Otherwise, the control moves to the statement following the if construct. Figure 7.1 shows the execution mode of if statement. While using if, certain points are to be remembered.

- The test expression is always enclosed in parentheses.

- The expression may be a simple expression constituted by relational expression or a compound expression constituted by logical expression.

- The statement block may contain a single statement or multiple statements. If there is a single statement, then it is not mandatory to enclose it in curly braces { }. If there are multiple statements, then they must be enclosed in curly braces.
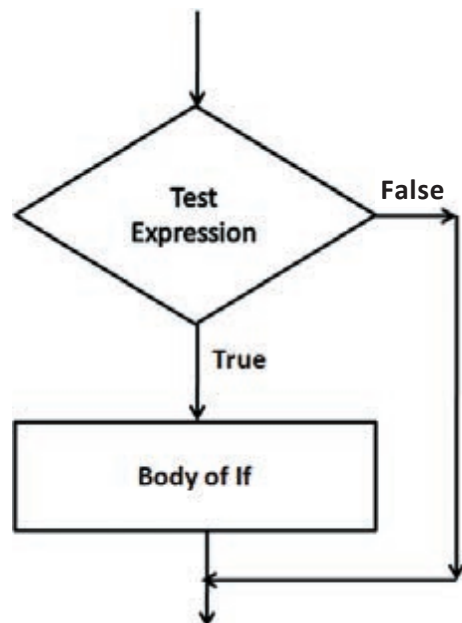

Fig. 7.1 : Working of if statement

Program 7.1 accepts the score of a student and displays the text "You have passed" only if he/she has passed. (assume that 18 is the minimum for pass).

Program 7.1: To display 'You have Passed' if score is 18 or more

```cpp
#include <iostream>
using namespace std;
int main()
{
    int score ;
    cout << "Enter your score: ";
    cin >> score;
    if (score >= 18)                    ⟵ Body of if
        cout << "You have Passed";
    return 0;
}
```

The following is a sample output of program 7.1:

```
Enter your score : 25
You have passed
```

In Program 7.1, the score of a student is entered and stored in the variable score. The test expression compares the value of score with 18. The body of if will be executed only if the test expression evaluates to True. That means, when the score is greater than or equal to18, the output "You have Passed" will be displayed on the screen. Otherwise, there will be no output.

Note that, the statement block associated with if is written after a tab space. We call it indentation. This is a style of coding which enhances the readability of the source code. Indentation helps the debugging process greatly. But it has no impact on the execution of the programme.

Consider the following C++ program segement. It checks whether a given character is an alphabet or a digit.

```cpp
char ch;                        Logical expression is evaluated
cin >> ch;
if (ch >= 'a' && ch <= 'z')                 Only a single
                                            statement. No
    cout <<"You entered an alphabet";       need of braces { }
if (ch >= '0' && ch <= '9')
{                                           More than
                                            one statement;
    cout<< "\nYou entered a digit";         Must be
    cout<< "\nIt is a decimal number";      enclosed in
}                                           braces { }
```

7.1.2 `if...else` statement

Consider the `if` statement in Program 7.1:

```
if (score >= 18)
    cout << "You have Passed";
```

Here, the output is obtained only if the score is greater than or equal to 18. What will happen if the score entered is less than 18? It is clear that there will be no output. Actually we don't have the option of selecting another set of statements if the test expression evaluates to False. If we want to execute some actions when the condition becomes False, we introduce another form of `if` statement, `if..else`. The syntax is:

```
if (test expression)
{
    statement block 1;
}
else
{
    statement block 2;
}
```

Here, if the test expression evaluates to True, only the statement block 1 is executed. If the test expression evaluates to False, statement block 2 is executed. The flowchart shown in Figure 7.2 explains the execution of if...else statement.
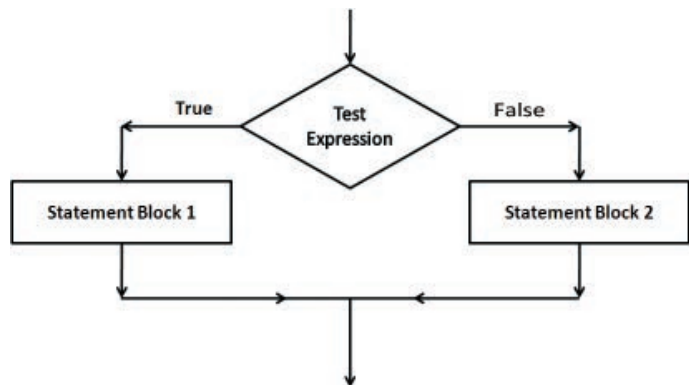
Fig 7.2: Flowchart of if - else

The following code segment illustrates the working of `if...else` statement.

```
if (score >= 18)
    cout << "Passed";
else
    cout << "Failed";
```

This statement is executed only when score is 18 or more (Test expression returns True)

This statement is executed only when score is less than 18 (Test expression returns False)

Let us write a program to input the heights of two students and find the taller.

Program 7.2: To find the taller student by comparing their heights

```
#include <iostream>
using namespace std;
int main()
{
    int ht1, ht2;
    cout << "Enter heights of the two students: ";
    cin >> ht1 >> ht2;
    if (ht1 > ht2)    //decision making based on condition
        cout << "Student with height "<<ht1<<" is taller";
    else
        cout << "Student with height "<<ht2<<" is taller";
    return 0:
}
```

When program 7.2 is executed, one of the output statements will be displayed. The selection depends upon the relational expression ht1>ht2. The following are sample outputs:

Output 1:
```
Enter heights of the two students: 170    165
Student with height 170 is taller
```

Output 2:
```
Enter heights of the two students: 160    171
Student with height 171 is taller
```

In the first output, we input 170 for ht1 and 165 for ht2. So, the test expression (ht1>ht2) is evaluated to True and hence the statement block of if is selected and executed. In the second output, we input 160 for ht1 and 171 for ht2. The test expression, (ht1>ht2) is evaluated and found False. Hence the statement block of else is selected and executed.

In if...else statement, either the code associated with if (statement block1) or the code associated with else (statement block 2) is executed.

Let us see another program that uses an arithmetic expression as one of the operands in the test expression. Program 7.3 uses this concept to check whether an input number is even or odd.

Program 7.3: **To check whether a given number is even or odd**

```cpp
#include <iostream>
using namespace std;
int main()
{
    int num;
    cout<<"\nEnter the number: ";
    cin >> num;
    if (num%2 == 0)
        cout << "The given number is Even";
    else
        cout << "The given number is Odd";
    return 0;
}
```

Some sample outputs of Program 7.3 are shown below:

Output 1:

```
Enter the number: 7
The given number is Odd
```

Output 2:

```
Enter the number: 10
The given number is Even
```

In this program, the expression (num%2) finds the remainder when num is divided by 2 and compares it with the value 0. If they are equal, the if block is executed, otherwise the else block is executed.

Let us do

1. *Write a program to check whether a given number is a non-zero integer number and is positive or negative.*

2. *Write a program to input enter a single character for sex and print out the gender.*

   If the input character is 'M' display "Male". If the input is 'F' display "Female"

3. *Write a program to input your age and check whether you are eligible to cast vote (the eligibility is 18 years and above).*

7.1.3 Nested `if`

In some situations there may arise the need to take a decision within if block. When we write an if statement inside another if block, it is called nesting. Nested means one inside another.

Consider the following program segment:

```
if (score >= 60)          outer if
{
        if(age >= 18)          inner if
            cout<<"You are selected for the course!";
}
```

In this code fragment, if the value of score is greater than or equal to 60, the flow of control enters the statement block of outer if. Then the test expression of the inner if is evaluated (i.e. whether the value of age is greater than or equal to 18). If it is evaluated to True, the code displays the message, "You are selected for the course!". Then the program continues to execute the statement following the outer if statement. An if statement, inside another if statement is termed as a **nested if** statement. The following is an expanded form of nested if.

```
if (test expression 1)
{
    if (test expression 2)          It will be executed if both the test
        statement 1;                     expressions are True.
    else
        statement 2;          It will be executed if test
}                             expression 1 is True, but test
else                          expression 2 is False.
{
    body of else ;          It will be executed if test
}                           expression 1 is False. The test
                            expression 2 is not evaluated.
```

The important point to remember about nested if is that an else statement always refers to the nearest if statement within the same block. Let us discuss this case with an example. Consider the following program segment:

```
cout<<"Enter your scores in Computer Science exam: ";
cin>>score;
if (score >= 18)
        cout<<"You have passed";
        if(score >= 54)
            cout<<" with A+ grade !";
else
        cout<<"\nYou have failed";
```

If we input the value 45 for `score`, the output will be as follows:

```
You have passed
You have failed
```

We know that this is logically not correct. Though the indentation of the code is proper, that doesn't matter in execution. The second `if` statement will not be considered as nested `if`, rather it is counted as an independent `if` with an `else` block. So, when the first `if` statement is executed, the `if` block is selected for execution since the test expression is evaluated to True. It causes the first line in the output. After that, while considering the second `if` statement, the test expression is evaluated to False and hence the second line in the output is obtained. So to get the correct output, the code should be modified as follows:

```
cout<<"Enter your scores in Computer Science exam: ";
cin>>score;
if (score >= 18)
{
    cout<<"You have passed";
    if(score >= 54)
        cout<<" with A+ grade !";
}
else
    cout<<"\nYou have failed";
```

Nesting is enforced by putting a pair of braces

It is now associated with the outer if

If we input the same value 45 as in the case of previous example, the output will be as follows:

```
You have passed
```

Program 7.4 uses nested `if` to find the largest among three given numbers. In this program, `if` statement is used in both the `if` block and `else` block.

Program 7.4: To find the largest among three numbers

```
#include <iostream>
using namespace std;
int main()
{
    int x, y, z;
    cout << "Enter three different numbers: ";
    cin >> x >> y >> z ;
```

```
    if (x > y)
    {
        if (x > z)
            cout << "The largest number is: " << x;
        else
            cout << "The largest number is: " << z;
    }
    else
    {
        if (y > z)
            cout << "The largest number is: " << y;
        else
            cout << "The largest number is: " << z;
    }
    return 0;
}
```

A sample output of program 7.4 is given below:

```
    Enter three different numbers: 6    2    7
    The largest number is: 7
```

As per the input given above, the test expression (x>y) in the outer if evaluated to True and hence the control enters the inner if. Here the test expression (x>z) is evaluated to False and so its else block is executed. Thus the value of z is displayed as the output.

Check yourself

1. Write a program to input an integer and check whether it is positive, negative or zero.

2. Write a program to input three numbers and print the smallest one.

7.1.4 The else if ladder

There are situations where an if statement is used within an else block. It is used in programs when multiple branching is required. Different conditions will be given and each condition will decide which statement is to be selected for execution. A common programming construct based on if statement is the else if ladder, also referred to as the else if staircase because of its appearance. It is also known as if...else if statement.

The general form of `else if` ladder is:

```
if (test expression 1)
    statement block 1;
        else if (test expression 2)
            statement block 2;
                else if (test expression 3)
                    statement block 3;
                        . . . . . . . . . . . . . .
                            else
                                statement block n;
```

At first, the `test expression 1` is evaluated and if it is True, the `statement block 1` is executed and the control comes out of the ladder. That means, the rest of the ladder is bypassed. If `test expression 1` evaluates to False, then the `test expression 2` is evaluated and so on. If any one of the test expressions evaluates to True, the corresponding statement block is executed and control comes out of the ladder. If all the test expressions are evaluated to False, the `statement block n` after the final `else` is executed. Observe the indentation provided in the syntax and follow this style to use `else if` ladder.

Let us illustrate the working of the `else if` ladder by writing a program to find the grade of a student in a subject when the score out of 100 is given. The grade is found out by the criteria given in the following table:

Score	Grade
80 or more	A
From 60 To 79	B
From 40 To 59	C
From 30 To 39	D
Below 30	E

Program 7.5: To find the grade of a student for a given score

```cpp
#include <iostream>
using namespace std;
int main()
{
    int score;
    cout << "Enter the score: ";
```

```
      cin >> score;
      if (score >= 80)
         cout<<"A Grade";
            else if (score >= 60)
               cout<<"B Grade ";
                  else if (score >= 40)
                     cout<<"C grade";
                        else if (score >= 30)
                           cout<<"D grade";
                              else
                                 cout<<"E Grade";
}
```

The following are the sample outputs of program 7.5:

Output 1:
```
      Enter the score: 75
      B Grade
```

Output 2:
```
      Enter the score: 25
      E Grade
```

In program 7.5, initially the test expression score>=80 is evaluated. Since the input is 75 in Output 1, the test expression is evaluated to False, and the next test expression score>=60 is evaluated. Here it is True, and hence "B Grade" is displayed and the remaining part of the else if ladder is bypassed. But in the case of Output 2, all the test expressions are evaluated to False, and so the final else block is executed which makes "E Grade" as the output.

Let us write a program to check whether the given year is a leap year or not. The input value should be checked to know whether it is century year (year divisible by 100). If it is a century year, it becomes a leap year only if it is divisible by 400 also. If the input value is not a century year, then we have to check whether it is divisible by 4. If it is divisible, the given year is a leap year, otherwise it is not a leap year.

Program 7.6: To check whether the given year is leap year or not

```
#include <iostream>
using namespace std;
int main()
{
    int year ;
    cout << "Enter year (in 4-digits): ";
    cin >> year;
```

```
        if (year % 100 == 0)// Checks for century year
        {
            if (year % 400 == 0)
                cout << "Leap Year\n" ;
            else
                cout<< "Not a leap year\n" ;
        }
        else if (year % 4 == 0)
            cout << "Leap Year\n" ;
        else
            cout<< "Not a leap year\n" ;
}
```

A century year is leap year only if it is divisible by 400

Non - century year is leap year only if it is divisible by 4

Let us see some sample outputs of Program 7.6:

Output 1:

```
    Enter year (in 4-digits): 2000
    Leap Year
```

Output 2:

```
    Enter year (in 4-digits): 2014
    Not a leap Year
```

Output 3:

```
    Enter year (in 4-digits): 2100
    Not a leap Year
```

Output 4:

```
    Enter year (in 4-digits): 2004
    Leap Year
```

We write one more program to illustrate the use of else if ladder. Program 7.7 allows to input a number between 1 and 7 to denote the day of a week and displays the name of the corresponding day. The input 1 will give you "Sunday" as output, 2 will give "Monday" and so on. If the input is outside the range 1 to 7, the output will be "Wrong input".

Program 7.7: To display the name of the day for a given day number

```
#include <iostream>
using namespace std;
int main()
{
    int day;
    cout << "Enter day number (1-7): ";
    cin >> day;
    if (day == 1)
      cout<<"Sunday" ;
        else if (day == 2)
```

```
            cout<<"Monday" ;
          else if (day == 3)
            cout<<"Tuesday" ;
              else if (day == 4)
                cout<<"Wednesday" ;
                  else if (day == 5)
                    cout<<"Thursday" ;
                      else if (day == 6)
                        cout<<"Friday" ;
                          else if (day == 7)
                            cout<<"Saturday" ;
                              else
                                cout<<"Wrong input" ;
    return 0;
}
```

The following are some sample outputs of Program 7.7:

Output 1:
```
Enter day number (1-7): 5
Thursday
```

Output 2:
```
Enter day number (1-7): 9
Wrong input
```

Check yourself

1. Write a program to input an integer number and check whether it is positive, negative or zero using if...else if statement.

2. Write a program to input a character (a, b, c or d) and print as follows:
   a - "abacus", b - "boolean", c - "computer" and d - "debugging"

3. Write a program to input a character and print whether it is an alphabet, digit or any other character.

7.1.5 switch statement

We have seen the concept of multiple branching with the help of else if ladder. Some of these programs can be written using another construct of C++ known as switch statement. This selection statement successively tests the value of a variable or an expression against a list of integer or character constants.

The syntax of switch statement is given below:

```
switch (expression)
{
    case constant_1    : statement block 1;
                         break;
    case constant_2    : statement block 2;
                         break;
    case constant_3    : statement block 3;
                         break;
                        :
                        :
    case constant_n-1  : statement block n-1;
                         break;
    default            : statement block n;
}
```

In the syntax **switch**, **case**, **break** and **default** are keywords. The expression is evaluated to get an integer or character constant and it is matched against the constants specified in the case statements. When a match is found, the statement block associated with that case is executed until the break statement or the end of switch statement is reached. If no match is found, the statements in the default block get executed. The default statement is optional and if it is missing, no action takes place when all matches fail. The break statement, used inside switch, is one of the jump statements in C++. When a break statement is encountered, the program control goes to the statements following the switch statement.

Program 7.7 can be written using switch statement. It enhances the readability and effectiveness of the code. Observe the modification in Program 7.8.

Program 7.8: To display the day of a week using switch statement

```
#include <iostream>
using namespace std;
int main()
{    int day;
     cout << "Enter a number between 1 and 7: ";
     cin >> day;
     switch (day)
     {
          case 1:    cout << "Sunday";
                     break;
          case 2:    cout << "Monday";
                     break;
          case 3:    cout << "Tuesday";
                     break;
```

```
            case 4:    cout << "Wednesday";
                       break;
            case 5:    cout << "Thursday";
                       break;
            case 6:    cout << "Friday";
                       break;
            case 7:    cout << "Saturday";
                       break;
            default:   cout << "Wrong input";
      }
}
```

The output of Program 7.8 will be the same as in Program 7.7. The following are some samples:

Output 1:
```
      Enter a number between 1 and 7: 5
      Thursday
```

Output 2:
```
      Enter a number between 1 and 7: 8
      Wrong input
```

In Program 7.8, value of the variable day is compared against the constants specified in the case statements. When a match is found, the output statement associated with that case is executed. If we input the value 5 for the variable day, then the match occurs for the fifth case statement and the statement cout << "Thursday"; is executed. If the input is 8 then no match occurs and hence the default block is executed.

Can you predict the output of Program 7.8, if all the break statements are omitted? The value returned by day is compared with the case constants. When the first match is found, the associated statements will be executed and the following statements will also be executed irrespective of the remaining constants. There are situations where we omit the break statement purposefully. If the statements associated with all the case in a switch are the same we only need to write the statement against the last case. Program 7.9 illustrates this concept.

Program 7.9: **To check whether the given character is a vowel or not**
```
#include <iostream>
using namespace std;
int main()
{
  char ch;
  cout<<"Enter the character to check: ";
  cin>>ch;
```

```
switch(ch)
{
    case 'A':
    case 'a':
    case 'E':
    case 'e':
    case 'I':
    case 'i':
    case 'O':
    case 'o':
    case 'U':
    case 'u': cout<<"Given character is a vowel";
              break;
    default : cout<<"Given character is not a vowel";
}
return 0;
}
```

Some of the outputs given by Program 7.9 are shown below:

Output 1:

```
Enter the character to check: E
Given character is a vowel
```

Output 2:

```
Enter the character to check: k
Given character is not a vowel
```

Suitability and requirements for using `switch`

Though switch statement and else if ladder cause multiple branching, they do not work in the same fashion. In C++ all switch statements can be replaced by else if ladders, but all else if ladders cannot be substituted by switch. The following are the requirements to implement a multi branching using switch statement:

- Conditions involve only equality checking. In other cases, it should be converted into equality expression.
- The first operand in all the equality expressions should be the same variable or expression.
- The second operand in these expressions should be integer or character constants.

Among the programs we discussed so far in this chapter, only the branching in Programs 7.3 and 7.7 can be replaced by switch. In Program 7.5, we can use switch if we modify the test expressions as score/10==10, score/10==9, score/10==8, and so on.

That is the following code fragment may be used instead of `else if` ladder.

```
switch(score/10)
{
    case 10 :
    case 9  : case 8 : cout<< "A Grade"; break;
    case 7  : case 6 : cout<< "B Grade"; break;
    case 5  : case 4 : cout<< "C Grade"; break;
    case 3  :          cout<< "D Grade"; break;
    default :          cout<< "E Grade";
}
```

Let us have comparison between `switch` and `else if` ladder as indicated in table 7.1

switch statement	**else if ladder**
• Permits multiple branching.	• Permits multiple branching.
• Evaluates conditions with equality operator only.	• Evaluate any relational or logical expression.
• Case constant must be an integer or a character type value.	• Condition may include range of values and floating point constants.
• When no match is found, `default` statement is executed.	• When no expression evaluates to True, `else` block is executed.
• `break` statement is required for exit from the `switch` statement.	• Program control automatically goes out after the completion of a block.
• More efficient when the same variable or expression is compared against a set of values for equality.	• More flexible and versatile compared to `switch`.

Table 7.1: Comparison between switch and else if ladder

7.1.6 The conditional operator (? :)

As we mentioned in Chapter 5, C++ has a ternary operator. It is the conditional operator (**?:**) consisting of symbols **?** and **:** (a question mark and a colon). It requires three operands to operate upon. It can be used as an alternative to `if...else` statement. Its general form is:

```
Test expression ? True_case code : False_case code;
```

The `Test expression` can be any relational or logical expression and `True_case code` and `False_case code` can be constants, variables, expressions or statement.

The operation performed by this operator is shown below with the help of an `if` statement.

```
if (Test expression)
{
      True_case  code;
}
else
{
      False_case  code;
}
```

The conditional operator works in the same way as `if...else` works. It evaluates the `Test expression` and if it is True, the `True_case code` is executed. Otherwise, `False_case code` is executed. Program 7.10 illustrates the working of conditional operator.

Program 7.10: To find the larger number using conditional operator

```
#include  <iostream>
using namespace std;
int main()
{
int n1, n2;
cout << "Enter two numbers: ";
cin>> n1 >> n2 ;
(n1>n2)? cout<<n1<<" is larger" : cout<<n2<<" is larger";
}
```

The last statement of the program may be called conditional statement as it uses conditional operator. This statement may be replaced by the following code segment:

```
      int big = (n1>n2) ? n1 : n2;
      cout << big << " is larger";
```

If the test expression evaluates to True, the value of `n1` will be assigned to `big`, otherwise that of `n2`. Here conditional operator is used to construct a conditional expression. The value returned by this expression will be assigned to `big`.

Check yourself

1. Write a program to input a number in the range 1 to 12 and display the corresponding month of the year. (January for the value 1, February for 2, etc.)

2. What is the significance of `break` statement within a `switch` statement?

3. Rewrite the following statement using `if...else` statement

```
      result= mark>30 ? 'p' :' f';
```

7.2 Iteration statements

In Chapter 3, we discussed some problems for which the solution contains some tasks that were executed repeatedly. While writing programs, we use some specific constructs of the language to perform the repeated execution of a set of one or more statements. Such constructs are called **iteration statements** or **looping statements**. In C++, we have three iteration statements and all of them allow a set of instructions to be executed repeatedly when a condition is True.

We use the concept of loop in everyday life. Let us consider a situation. Suppose your class teacher has announced a gift for every student securing A+ grade in an examination. You are assigned the duty of wrapping the gifts. The teacher has explained the procedure for wrapping the gifts as follows:

Step 1	:	Take the gift
Step 2	:	Cut the wrapping paper
Step 3	:	Wrap the gift
Step 4	:	Tie the cover with a ribbon
Step 5	:	Fill up a name card and paste it on the gift pack

If there are 30 students with A+ grade in the examination, you have to repeat the same procedure 30 times. To repeat the wrapping process 30 times, the instructions can be restructured in the following way.

Repeat the following steps 30 times
{   Take the next gift
*   Cut the wrapping paper*
*   Wrap the gift*
*   Tie the cover with a ribbon*
*   Fill up a name card and paste on the gift pack*

}

Let us take another example. Suppose we want to find the class average of scores obtained in Computer Applications. The following steps are to be performed:

Initially Total_Score has no value
Repeat the following steps starting from the first student till the last
{   Add Score of the student to the Total_Score
*   Take the Score of the next student*

}
Average = Total_Score /No. of students in the class

In both the examples, we perform certain steps for a number of times. We use a counter to know how many times the process is executed. The value of this counter decides whether to continue the execution or not. Since loops work on the basis of such

conditions, a variable like the counter will be used to construct a loop. This variable is generally known as *loop control variable* because it actually controls the execution of the loop. In Chapter 3, we discussed four elements of a loop. Let us refresh them:

1. **Initialisation**: Before entering a loop, its control variable must be initialised. During initialisation, the loop control variable gets its first value. The initialisation statement is executed only once, at the beginning of the loop.

2. **Test expression**: It is a relational or logical expression whose value is either True or False. It decides whether the loop-body will be executed or not. If the test expression evaluates to True, the loop-body gets executed, otherwise it will not be executed.

3. **Update statement**: The update statement modifies the loop control variable by changing its value. The update statement is executed before the next iteration.

4. **Body of the loop**: The statements that need to be executed repeatedly constitute the body of the loop. It may be a simple statement or a compound statement.

We learnt in Chapter 3 that loops are generally classified into entry-controlled loops and exit-controlled loops. C++ provides three loop statements: **while** loop, **for** loop and **do...while** loop. Let us discuss the working of each one in detail.

7.2.1 **while** statement

The while loop is an entry-controlled loop. The condition is checked first and if it is found True the body of the loop will be executed. That is the body will be executed as long as the condition is True. The syntax of while loop is:

```
initialisation of loop control variable;
while(test expression)
     {
     body of the loop;
     updation of loop control variable;
     }
```

Here, the test expression defines the condition which controls the loop. The body of the loop may be a single statement or a compound statement or without any statement. The body is the set of statements for repeated execution. Update expression refers to a statement that changes the value of the loop control variable. In a while loop, a loop control variable should be initialised before the loop begins and it should be updated inside the body of the loop. The flowchart in Figure 7.3 illustrates the working of a while loop.
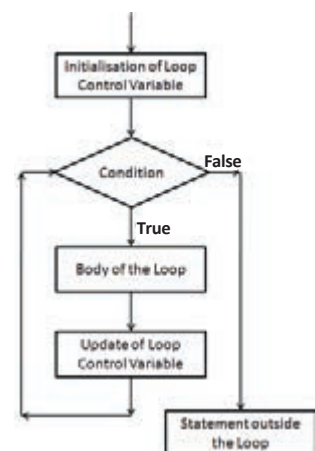
Fig 7.3: Working of while *loop*

The initialisation of the loop control variable takes place first. Then the `test expression` is evaluated. If it returns True the body of the loop is executed. That is why `while` loop is called an **entry controlled loop**. Along with the loop body, the loop control variable is updated. After completing the execution of the loop body, test expression is again evaluated. The process is continued as long as the condition is True. Now, let us consider a code segment to illustrate the execution of `while` loop.

```
int k=1;          ← Initialization before loop begins
while(k<=3 )      ← Test expression
{
     cout << k << '\t';     ← Body of loop
     ++k;
}                 ← Updation inside the loop body
```

In this code segment, the value 1 is assigned to the variable **k** (loop control variable) at first. Then the test expression k<=3 is evaluated. Since it is True, the body of the loop is executed. That is the value of **k** is printed as 1 on the screen. After that the update statement ++k is executed and the value of **k** becomes 2. The condition k<=3 is checked again and found to be True. Program control enters the body of the loop and prints the value of **k** as 2 on the screen. Again the update statement is executed and the value of **k** is changed to 3. Since the condition is still True, body is executed and 3 is displayed on the screen. The value of **k** is again updated to 4 and now the test expression is evaluated to be False. The control comes out of the loop and executes the next statement after the `while` loop. In short, the output of the code will be:

> 1   2   3

Imagine what will happen if the initial value of **k** is 5? The test expression is evaluated to False in the first evaluation and the loop body will not be executed. This clearly shows that `while` loop controls the entry into the body of the loop.

Let us see a program that uses `while` loop to print the first 10 natural numbers.

Program 7.11: To print the first 10 natural numbers

```
#include <iostream>
using namespace std;
int main()
{                              ← Initialisation of loop variable
     int n = 1;
     while(n <= 10)            ← Test expression
     {
          cout << n << "   ";  ← Body of loop
          ++n;
     }                         ← Updating loop variable
}
```

The output of Program 7.11 will be as follows:

```
1   2   3   4   5   6   7   8   9   10
```

Program 7.12 uses `while` loop to find the sum of even numbers upto 20. This program shows that the loop control variable can be updated using any operation.

Program 7.12: **To find the sum of even numbers upto 20**

```cpp
#include <iostream>
using namespace std;
int main()
{
    int i, sum = 0;
    i = 2;
    while( i<= 20)
    {
        sum = sum + i;        Updating of loop control variable by
        i = i + 2;               adding 2 to the current value
    }
    cout<<"\nsum of even numbers up to 20 is: "<<sum;
    return 0;
}
```

The output of Program 7.12 is given below:

```
sum of even numbers up to 20 is: 110
```

Let us do

Modify Program 7.11 to display all odd numbers between 100 and 200.

Modify Program 7.12 to find the sum of the first N natural numbers.

If we put a semi colon (;) after the test expression of while statement, there will not be any syntax error. But the statements within the following pair of braces will not be considered as loop body. The worst situation is that, if the test expression is evaluated to be True, neither the code after the while loop will be executed nor the program will be terminated. It is a case of infinite loop.

7.2.2 `for` statement

The `for` loop is also an entry-controlled loop in C++. All the three loop elements (initialisation, test expression and update statement) are placed together in `for` statement. So it makes the program compact.

The syntax is:

```
for (initialisation; test expression; update statement)
{
    body-of-the-loop;
}
```

The execution of **for** loop is the same as that of **while** loop. The flowchart used for **while** can explain the working of for loop. Since the three elements come together this statement is more in situations where counting is involved. So the flowchart given in Figure 7.4 is commonly used to show the execution of for statement. At first, the
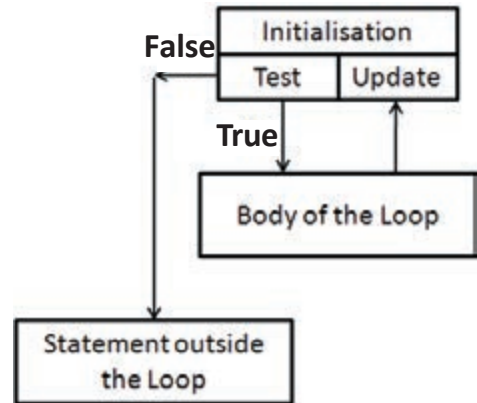
Fig 7.4: Execution of for *loop*

initialisation takes place and then the test expression is evaluated. If its result is True, body-of-the-loop is executed, otherwise the program control goes out of the for loop. After the execution of the loop body, update expression is executed and again test expression is evaluated. These three steps (test, body, update) are continued until the test expression is evaluated to False.

The loop segment used in Program 7.11 can be replaced with a for loop as follows:

```
for (n=1; n<=10; ++n)
    cout << n << "    ";
```

> *The steps 1 and 2 in the execution sequence of the* for *loop just mentioned before are given below. Write down the remaining steps.*
>
> **Let us do**
>
> *Step 1: N =1, Condition is True, 1 is displayed, N becomes 2*
>
> *Step 2: Condition is True, 2 is displayed, N becomes 3*
>
> *Step 3: ……………*

This code is executed in the same way as in the case of while loop.

Let us write a program using for loop to find the factorial of a number. Factorial of a number, say **N**, represented by **N!** is the product of the first **N** natural numbers. For example, factorial of 5 (5!) is calculated by $1 \times 2 \times 3 \times 4 \times 5 = 120$.

Program 7.13: To find the factorial of a number using `for` loop

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, fact=1;
    cout<<"Enter number to find the factorial: ";
    cin>>n;
    for (i=1; i<=n; ++i)          Initialisation; Test Expression; Updation
        fact = fact * i ;              Loop body
    cout << "Factorial of " << n << " is " << fact;
}
```

The following is a sample output of program 7.13

```
Enter number to find the factorial:   6
Factorial of 6 is 720
```

Another program is given below which gives the class average of scores obtained in Computer Applications. Program 7.14 accepts the value for **n** as the number of students, then reads the score of each student and prints the average score.

Program 7.14: To find the average score of n students

```
#include <iostream>
using namespace std;
int main()
{
    int i, sum, score, n;
    float avg;                            Initialisation contains
    cout << "How many students? ";              two expressions
    cin >> n ;
    for(i=1, sum=0;   i<=n; ++i)
    {
        cout << "Enter the score of student " << i <<": ";
        cin >> score;
        sum = sum + score;        Explicit type conversion
    }
    avg = (float)sum / n;
    cout << "Class Average is: " << avg;
}
```

The following is a sample output of Program 7.14 for 5 students

```
How many students? 5
Enter the score of student 1: 45
Enter the score of student 2: 50
Enter the score of student 3: 52
Enter the score of student 4: 34
Enter the score of student 5: 55
Class Average is: 47.2
```

While using `for` loops certain points are to be noted. The four code segments explain these special cases. Assume that all the variables used in the codes are declared with int data type.

Code segment 1:
```
for (n=1; n<5; n++);
        cout<<n;
```

A semicolon (;) appears after the parentheses of for statement. It is not a syntax error. Can you predict the output? If it is 5, you are correct. This loop has no body. But its process will be completed as usual. The initialisation assigns 1 to **n**, condition is evaluated to True, since there is no body update taking place and the process continues till **n** becomes 5. At that point, condition is evaluated to be False and the program control comes out of the loop. The execution of the output statement displays 5 on the screen

Code segment 2:
```
for (n=1; n<5; )
        cout<<n;
```

In this code, update expression is not present. It does not make any syntax error in the code. But on execution, the loop will never be terminated. The number 1 will be displayed infinitely. We call this an infinite loop.

Code segment 3:
```
for (; n<5; n++)
        cout<<n;
```

The output of this code cannot be predicted. Since there is no initialisation, the control variable **n** gets some integer value. If it is smaller than 5, the body will be executed until the condition becomes False. If the default value of **n** is greater than or equal to 5, the loop will be terminated without executing the loop body.

Code segment 4:
```
for (n=1; ; n++)
        cout<<n;
```

The test expression is missing in this code. C++ takes this absence as True and obviously the loop becomes an infinite loop. The four code segments given reveal that all the elements of a **for** loop are optional. But this is not the case for **while** and **do...while** statements. Test expression is compulsory for these two loops. Other elements are optional, but be cautious about the output.

In Program 7.14, the initialisation contains two expressions `i=1` and `sum=0` separated by comma (`,`). The initialisation part may contain more than one expression, but they should be separated by comma. Both the variables `i` and `sum` get their first values 1 and 0 respectively. Then, the test expression `i<=n` is evaluated to be True and body of the loop is executed. After the execution of the body of the loop the update expression `++i` is executed. Again the test expression `i<=n` is evaluated, and body of the loop is executed since the condition is True. This process continues till the test expression returns False. It has occurred in the sample output when the value of `i` becomes 6.

Write a program to display the multiplication table of a given number. Assume that the number will be the input to the variable n. The body of the loop is given below:

Let us do

```
cout<<i<<" x "<<n<<" = "<< i * n <<"\n";
```

Give the output also.

7.2.3 `do...while` statement

In the case of `for` loop and `while` loop, the test expression is evaluated before executing the body of the loop. If the test expression evaluates to False for the first time itself, the body is never executed. But in some situations, it is necessary to execute the loop body at least once, without considering the result of the test expression. In that case the **do...while** loop is the best choice. Its syntax is :

```
initialisation of loop control variable;
do
{
    body of the loop;
    Updation of loop control variable;
} while(test expression);
```

Figure 7.5 shows the order of execution of this loop. Here, the test expression is evaluated only after executing body of the loop. So `do...while` loop is an **exit controlled loop**. If the test expression evaluates to False, the loop will be terminated. Otherwise, the execution process will be continued. It means that in `do...while` loop the body will be executed at least once irrespective of the result of the condition.
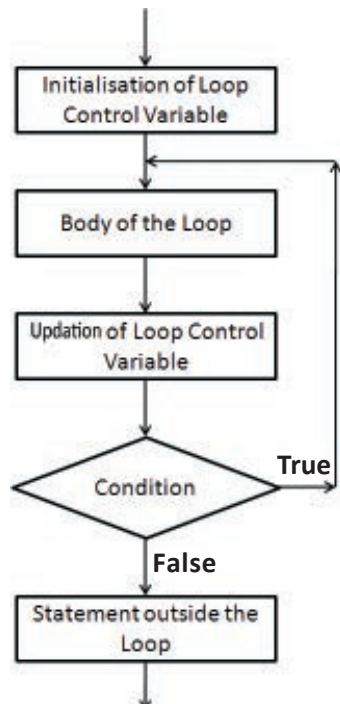
Fig. 7.5 : Flowchart of `do...while` *loop*

Let us consider the following program segment to illustrate the execution of a `do...while` loop.

```
                                      Initialisation before the loop
    int k=1;
    do
    {
         cout << k << '\t';          Body of loop
         ++k;                        Updation inside the loop body
    } while(k<=3);
                                      Test expression at the end
```

At first, the value 1 is assigned to the variable **k**. Then body of the loop is executed and the value of **k** is printed as 1. After that the **k** is incremented by 1 (now k=2). Then it checks the condition k<=3. Since it is found True the body of the loop is executed to print the value of **k**, i.e. 2 on the screen. Again the updation process is carried out, which makes value of **k** as 3 and the condition k<=3 is checked again. As it is True, the body of the loop is executed to print the value 3. The variable **k** is again updated to 4 and now the condition is evaluated to be False. It causes the program control to come out of the loop and executes the next statement after the loop body. Thus the output of the code will be:

  1   2   3

Now let us see how this loop differs from the other two. Imagine that the initial value of **k** is 5. What will happen? The body of the loop is executed and the value of **k** will be printed on the screen as 5. After that the variable **k** will be updated by incrementing it by 1 and **k** becomes 6. On checking the condition k<=3, the test expression is evaluated to False and the control comes out of the loop. This clearly shows that in `do...while` loop there is no restriction to enter the loop body for the first time. So if we want the body to be executed based on the True value of the condition, use `while` or `for` loops.

Let us see an interactive program in the sense that some part of the code will be executed on user's choice. The simplest form of such programs provides facility to accept user's response for executing a code segment in the program repeatedly. Program 7.15 illustrates the use of `do...while` loop to write an interactive program to find the area of rectangles by accepting the length and breadth of each rectangle from the user.

Program 7.15: **To find the area of rectangles**

```cpp
#include <iostream>
using namespace std;
int main()
{
    float length, breadth, area;
    char ch;
    do
    {
        cout << "Enter length and breadth: ";
        cin >> length >> breadth;
        area = length * breadth;
        cout << "Area = " << area;
        cout << "Any more rectangle (Y/N)? ";
        cin >> ch;
    } while (ch == 'Y' || ch == 'y');
    return 0;
}
```

A sample output of Program 7.15 is given below:

```
Enter length and breadth: 3.5    7        User input
Area = 24.5
Any more rectangle (Y/N)? Y        User input
Enter length and breadth: 6    4.5
Area = 27
Any more rectangle (Y/N)? N        User input
```

We have discussed all the three looping statements of C++. Table 7.2 shows a comparison between these statements.

for loop	**while loop**	**do...while loop**
Entry controlled loop	Entry controlled loop	Exit controlled loop
Initialisation along with loop definition	Initialisation before loop definition	Initialisation before loop definition
No guarantee to execute the loop body at least once	No guarantee to execute the loop body at least once	Will execute the loop body at least once even though the condition is False

Table 7.2 : Comparison between the looping statements of C++

Let us sum up

The statements providing facilities for taking decisions or for performing repetitive actions in a program are known as control statements. The control statements are the backbones of a computer program. In this chapter we covered the different types of control statements such as selection statements - (if, if…else, if…else if, switch) and iteration statements or loop statements (for, while, do…while). All these control statements will help us in writing complex C++ programs. These statements are essential for varuiys programming operations to be performmed.

Learning outcomes

After the completion of this chapter the learner will be able to

- use control statements in C++ for problem solving.
- identify the situation where control statements are used in a program.
- use correct control statements suitable for the situations.
- categorise different types of control statements.
- write C++ programs using control statements.

Lab activity

1. Write a program to input a digit and print it in words.
2. Write a program to print first 'n' odd numbers.
3. Write a program to print the sum of squares of first n odd numbers.
4. Write a program to print every integer between 1 and n which are exactly divisible by m.

Sample questions

Very short answer type

1. Write the significance of break statement in switch statement. What is the effect of absence of break in a switch statement?
2. What will be the output of the following code fragment?
   ```
   for(i=1;i<=10;++i)  ;
   cout<<i+5;
   ```
3. Write a program using for loop that will print the numbers between 1000 and 2000 which are divisible by 132.

4. Rewrite the following statement using **while** and **do...while** loops.

```
for (i=1; i<=10; i++) cout<<i;
```

5. How many times the following loop will execute?

```
int   s=0, i=0;
while(i++<5)
      s+=i;
```

Short answer type

1. Consider two program fragments given below.

// version 1	//version 2
cin>>mark;	cin>>mark;
if (mark >= 90)	if (mark>=90)
cout<<"A+";	cout<<"A+";
if (mark >= 80 && mark <90)	else if (mark>=80 && mark <90)
cout<<"A";	cout<<"A";
if (mark >= 70 && mark <80)	else if (mark>=70 && mark <80)
cout<<"B+";	cout<<"B+";
if (mark >= 60 && mark <70)	else if (mark>=60 && mark <70)
cout<<"B";	cout<<"B";

   discuss the advantages of version 2 over version 1.

2. Briefly explain the working of a for loop along with its syntax. Give an example of for loop to support your answer.

3. Compare and discuss the suitability of three loops in different situations.

4. What is wrong with the following while statement if the value of $z = 3$.

```
while(z>=0)
        sum+=z;
```

5. Consider the following if else if statement. Rewrite it with switch command.

```
if (a==1)
      cout << "One";
else if (a==0)
      cout << "Zero";
else
      cout << "Not a binary digit";
```

6. Write the importance of a loop control variable. Briefly explain the different parts of a loop.

Long answer type

1. Explain different types of decision statements in C++.

2. Explain different types of iteration statements available in C++ with syntax and examples.

8

Computer Networks

Have you surfed the Internet to search your examination result or to know whether you got admission to the Plus One course in a school of your choice? Have you visited an ATM counter to draw money? Have you transferred songs, images or movie clips from your computer to a cell phone or booked a train ticket using Internet? If your answer to any of these questions is 'yes', you have accessed the services of a computer network. In this chapter, we will learn more about the working of networks and their advantages. We will also discuss different media and devices, different types of networks and the rules to be followed in data communication using these devices.

8.1 Computer network

Computer network is a group of computers and other computing hardware devices (such as printers, scanners, modems, CD drives, etc.) connected to each other electronically through a communication medium. They can communicate with each other, exchange commands, share data, hardware and other resources. Computers on a network may be linked through cables, telephone lines, radio waves, satellites or infrared light beams.

8.1.1 Need for network

Internet is a good example for a computer network. It is impossible to imagine a world without e-mails, online newspapers, blogs, chat and other services offered through the Internet. Apart from these, there are many other advantages in using networked computers instead of stand-alone computers. Some of them are listed below.

- Resource sharing
- Price-performance ratio
- Communication
- Reliability
- Scalability

Resource sharing: The sharing of available hardware and software resources in a computer network is called **resource sharing**. For example, the contents of a DVD placed in a DVD drive of one computer can be read in another computer. Similarly, other hardware resources like hard disk, printer, scanner, etc. and software resources like application software, anti-virus tools, etc. can also be shared through computer networks.

Price-performance ratio: One can easily share the resources available in one computer with other computers. The cost of purchasing licensed software for each computer can be reduced by purchasing network versions of such software. This will least affect the performance of such resources and lead to considerable savings in cost.

Communication: Computer network helps user to communicate with any other user of the network through its services like e-mail, chatting, video conferencing etc. For example, one can send or receive messages within no time irrespective of the distance.

Reliability: It is possible to replicate or backup data/information in multiple computers using the network. For example, the C++ files, photos or songs saved in one computer can also be saved in other computers in the same network. These can be retrieved from other computers in which they are saved in case of disasters (malfunctioning of computers, accidental deletion of files, etc.)

Scalability: Computing capacity can be increased or decreased easily by adding or removing computers to the network. In addition to this, the storage capacity of networks can also be increased by including more storage devices to the network.

8.1.2 Some key terms

Some of the key terms related to computer network are explained below:

Bandwidth : Bandwidth measures the amount of data that can be sent over a specific connection in a given amount of time. Imagine you are in a highway or a public road. The bigger the road, the more will be the number of vehicles that can travel on it. Moreover, the traffic here is faster than on a narrow road. On a narrow road, the traffic is likely to be congested. We can say that the bandwidth of a bigger road is higher than a narrow road.

Bandwidth describes the maximum data-transfer rate between computers in a network. In digital systems, bandwidth is measured in bits per second (bps). If the bandwidth is more, data travels faster and hence large amounts of data can be transferred within a particular time span across the network. For example, an Internet connection via cable modem may provide 25 Mbps of bandwidth.

Noise : Noise is unwanted electrical or electromagnetic energy that lowers the quality of data signals. It occurs from nearby radio transmitters, motors or other cables. The transfer of all types of data including texts, programs, images and audio over a network is adversely affected by noise.

Node : Any device (computer, scanner, printer, etc.) which is directly connected to a computer network is called a node. For example, computers linked to the computer network in the school are nodes. When we connect the Internet to our computer, our computer becomes a node of the Internet.

Make a list of the hardware and software resources shared in your school network.

Let us do

8.2 Data communication system

In a computer network, computing devices are connected in various ways, to communicate and share resources. **Data communication** is the exchange of digital data between any two devices through a medium of transmission. Figure 8.1 show the representation of a general data communication system. The following five basic elements are necessary for building any data communication system.
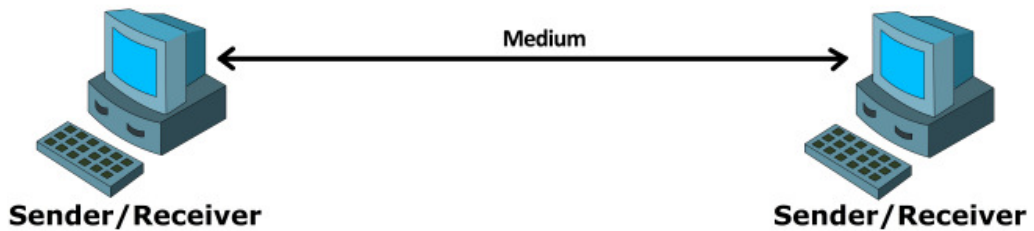
Fig. 8.1 : Data communication system

Message : It is the information to be communicated. Major forms of information include text, picture, audio, video, etc.

Sender : The computer or device that is used for sending messages is called the sender, source or transmitter.

Receiver : The computer or device that receives the messages is called the receiver.

Medium : It is the physical path through which a message travels from the sender to the receiver. It refers to the way in which nodes are connected.

Protocol : The rules under which message transmission takes place between the sender and the receiver is called a protocol.

8.3 Communication medium

Data communication is possible only if there is a medium through which data can travel from one device to another. The medium for data transmission over a computer network is called **communication channel** or **communication medium**. The communication medium between computers in a network are of two types: guided and unguided. In guided or wired medium, physical wires or cables are used and in unguided or wireless medium radio waves, microwaves or infrared signals are used for data transmission.

8.3.1 Guided medium (Wired)

The coaxial cable, twisted pair cable (Ethernet cable) and optical fibre cable are the different types of cables used to transfer data through computer networks.

a. Twisted pair cable (Ethernet cable)

This is the most widely used media in small computer networks. It consists of four twisted pairs which are enclosed in an outer shield. These pairs are colour coded. Twisted pair cables are of two types: (i) Unshielded Twisted Pair (UTP) cables and (ii) Shielded Twisted Pair (STP) cables.

Unshielded Twisted Pair (UTP) cable : As its name suggests, the individual pairs in UTP cables are not shielded. Figure 8.2 shows the components of a UTP cable.
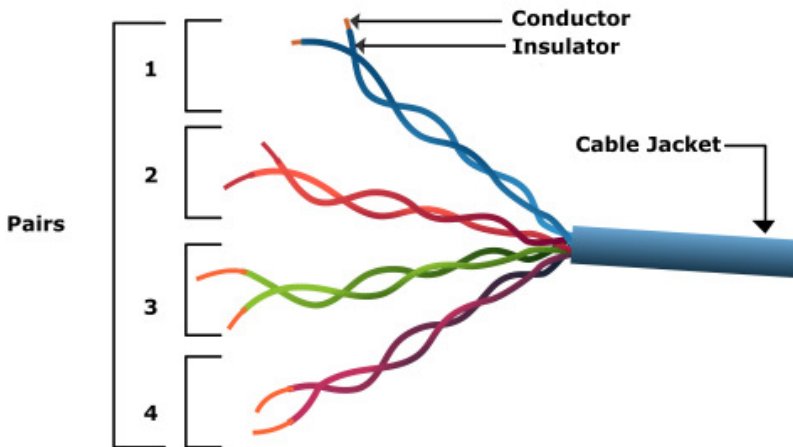
Fig. 8.2 : UTP cable

Characteristics of UTP cable

- Low cost cable available for setting up small networks.
- Thin and flexible cable.
- Ease of installation.
- Carries data upto a length of 100 m at a stretch.

Shielded Twisted Pair (STP) cable : It is the same cable as the UTP, but with each pair shielded individually. An outer shield then covers all the pairs like in UTP.

Characteristics of STP cable

- Shielding in STP offers better immunity against noise.
- It is more expensive than UTP cable.
- Compared to UTP cable, STP cable is difficult to install.

An RJ-45 connecter is used to connect UTP/ STP twisted pair cable to a computer. Figure 8.3 shows the STP cable and RJ-45 connector.

Fig. 8.3 : STP cable, RJ-45 connecter

b. Coaxial cable

A coaxial cable consists of an inner conductor surrounded by a tubular insulating layer which is further covered by a tubular conducting shield. It has an outer insulation to

protect the cable too. Figure 8.4 shows the construction of a coaxial cable.

Characteristics of coaxial cable

Fig. 8.4 : Coaxial Cable

- Carries data to longer distances (185 m - 500 m) at a stretch.

- High bandwidth.

- Less electromagnetic interference due to the outer shield.

- Thicker than twisted pair.

- Less flexible than twisted pair.

- Difficult to install than twisted pair cable.
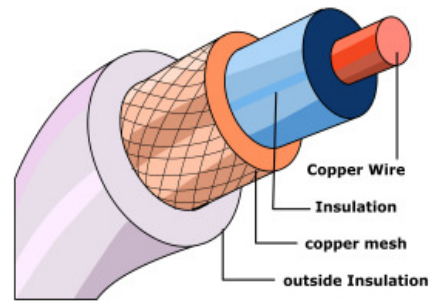
c. Optical fibre cable

Optical fibres are long thin glass fibres through which data is transmitted as light signals. Data travels as fast as light and can be transmitted to far off distances. Figure 8.5 shows the major parts of an optical fibre cable.

Optical fibre has the following parts:

- Core - the thin glass rod at the centre through which the light travels.

Fig. 8.5 : Optical Fibre

- Cladding - the outer optical material surrounding the core that reflects the light back into the core.

- Coating - the plastic coating that protects the cable from damage and moisture.
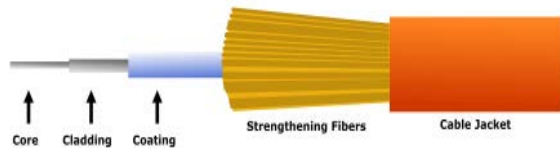
These optical fibres are arranged in bundles of hundreds and thousands and are protected by the outer covering of the cable called jacket.

At the source end, the optical transmitter converts electrical signals into optical signals (modulation) using semiconductor devices such as light-emitting diodes (LEDs) or laser diodes. At the destination end, the optical receiver, consisting of a photo detector, converts light back to electric signals (demodulation) using the photoelectric effect. The speed of transmission and the distance of signals are higher for laser diodes than for LEDs.

Characteristics of optical fibre cable

- High bandwidth for voice, video and data applications
- Carries data over a very long distance at a stretch.
- Not susceptible to electromagnetic fields, as it uses light for data transmission.
- The most expensive and the most efficient communication media available for computer networks.
- Installation and maintenance are difficult and complex.

8.3.2 Unguided medium (Wireless)

Electromagnetic waves are used for wireless communication on computer networks. Frequencies of waves are measured in Hertz (Hz). Based on their frequencies, electromagnetic waves are categorised into various types as shown in Figure 8.6. From this category we can infer that only radio waves, microwaves and infrared rays are used for wireless communication.
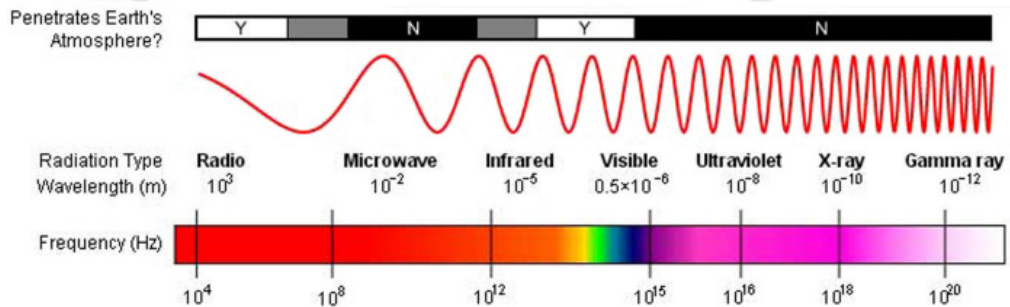
Fig. 8.6 : Electromagnetic Spectrum

a. Radio waves

Radio waves have a frequency range of 3 KHz to 3 GHz. Radio waves can be used for short and long distance communication. These waves are easy to generate and can go over the walls of a building easily. That is why radio waves are widely used for communication-both indoors and outdoors. Cordless phones, AM and FM radio broadcast and mobile phones make use of radio wave transmission.
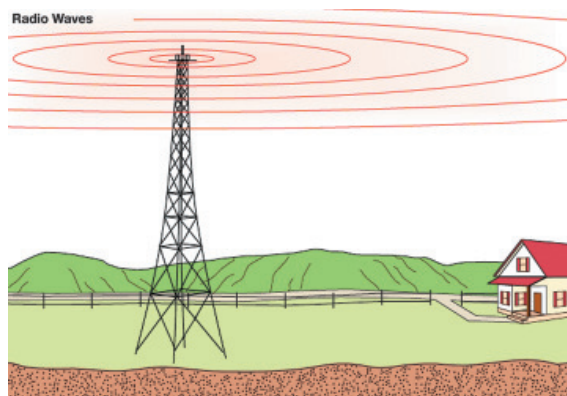
Fig. 8.7 : Radio wave transmission

Characteristics of radio wave transmission

- Waves are transmitted in all directions, therefore transmitting and receiving antennas need not be aligned face to face.

- Relatively inexpensive than wired media.

- Can penetrate through most objects.

- Transmission can be affected by motors or other electrical equipment.

- Less secure mode of transmission.

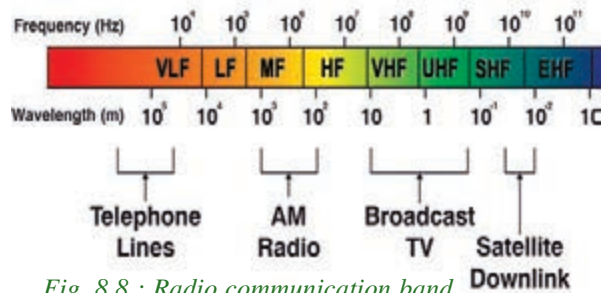- Permissions from authorities concerned are required for the use of radio wave transmission.

Fig. 8.8 : Radio communication band

b. Micro waves

Micro waves have a frequency range of 300 MHz (0.3 GHz) to 300 GHz. Microwaves travel in straight lines and cannot penetrate any solid object. Therefore, high towers are built and microwave antennas are fixed on their top for long distance microwave communication. As these waves travel in straight lines the antennas used for transmitting and receiving messages have to be aligned with each other. The distance between two microwave towers depends on many factors including frequency of the waves being used and heights of the towers. Figure 8.9 shows the components of a microwave transmission system.
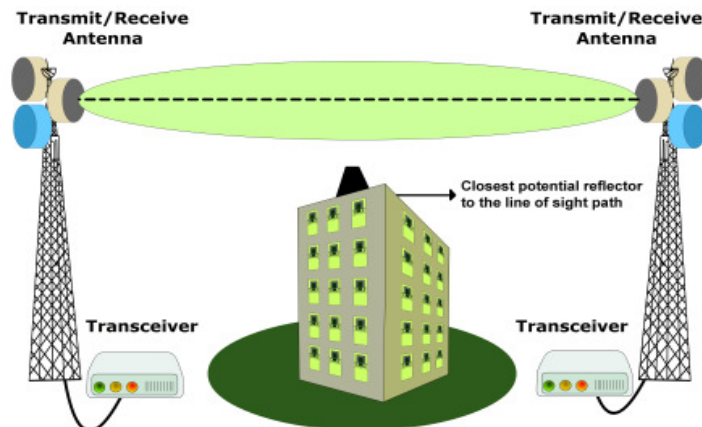
Fig. 8.9 : Microwave transmission

Characteristics of micro wave transmission

- Relatively inexpensive than wired media.

- Offers ease of communication over difficult terrain.

- The transmission is in straight lines. Therefore, the transmitting and receiving antennas need to be properly aligned (line of sight transmission).

c.   Infrared waves

Infrared waves have a frequency range of 300 GHz to 400 THz. These waves are used for short range communication (approx. 5 m) in a variety of wireless communications, monitoring and control applications.

Home entertainment remote control devices, cordless mouse and intrusion detectors are some of the devices that utilise infrared communication (refer Figure 8.10).

Fig. 8.10 : Infrared transmission

Characteristics of infrared wave transmission

- A line of sight transmission; hence information passed to one device is not leaked.

- Only two devices can communicate at a time.

- The waves cannot cross solid objects. (You may stand between the remote control and your television set and check whether the remote control works.)

- The longer the distance the weaker the performance.

8.3.3 Wireless communication technologies using radio waves

a.   Bluetooth

Bluetooth technology uses radio waves in the frequency range of 2.402 GHz to 2.480 GHz. This technology is used for short range communication (approx. 10 m) in a variety of devices for wireless communication. Cell phones, laptops, mouse, keyboard, tablets, head sets, cameras etc. are some of the devices that utilise bluetooth communication (refer Figure 8.11).

Fig. 8.11 : Bluetooth transmission

Characteristics of bluetooth transmission

- Line of sight between communicating devices is not required.
- Bluetooth can connect upto eight devices simultaneously.
- Slow data transfer rate (upto 1 Mbps).

b.   Wi-Fi

Wi-Fi network makes use of radio waves to transmit information across a network like cell phones, televisions and radios. The radio waves used in Wi-Fi ranges from a frequency of 2.4 GHz to 5 GHz. Communication across a wireless network is two-way radio communication. The wireless adapter in a computer translates data into radio signal and transmits it using an antenna. A wireless router receives the signal and decodes it. Once decoded, the data will be sent to the Internet or network through a wired Ethernet /wireless connection. Similarly, the data received from the Internet/ network will also pass through the router and coded into radio signals that will be received by the wireless adapter in a computer as indicated in Figure 8.12. Nowadays, this technology is widely used to share Internet connection with laptops or desktops.

Fig. 8.12 : Wi-Fi transmission

Characteristics of Wi-Fi transmission

- Line of sight between communicating devices is not required
- Data transmission speed is up to 54 Mbps
- Wi-Fi can connect more number of devices simultaneously
- Used for communication upto 375 ft (114 m)

c.   Wi-MAX

Worldwide Interoperability for Microwave Access (Wi-MAX) combines the benefits of broadband and wireless. Wi-MAX has a the frequency range of 2 GHz to 11 GHz.

Wi-MAX can provide high-speed wireless Internet access over very long distances (a whole city). Wi-MAX equipment exists in two basic forms - base stations, installed by service providers to deploy the technology in a coverage area, and receivers, installed by clients. Figure 8.13 shows the basic components of a WiMAX transmission.

Fig. 8.13 : Wi-Max transmission

Characteristics of Wi-MAX transmission

- Hundreds of users can connect to a single station.
- Provides higher speed connection upto 70 Mbps over an area of 45 Kilometres.
- Line of sight between communicating devices is not required.
- Weather conditions like rain, storm etc. could interrupt the signal.
- Very high power consumption.
- High costs of installation and operation.

d.   Satellite link

Long distance wireless communication systems use satellite links for transmitting signals. Usually, a signal travels in a straight line and is not able to bend around the globe to reach a destination far away. Signals can be sent to geostationary satellites in space and then redirected to another satellite or directly to a far away destination. A geostationary satellite orbits the earth in the same direction and amount of time it takes to revolve the earth once is equal to the time taken for earth's one rotation. From the earth, therefore, the satellite appears to be stationary, always above the same area of the earth. These satellites carry electronic devices called transponders for receiving, amplifying, and re-broadcasting signals to the earth.

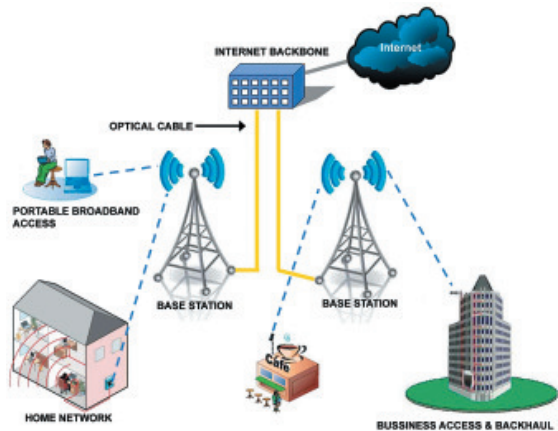Transmission of signals from the earth to a satellite is called **uplink** and from a satellite

Fig. 8.14 : Satellite link
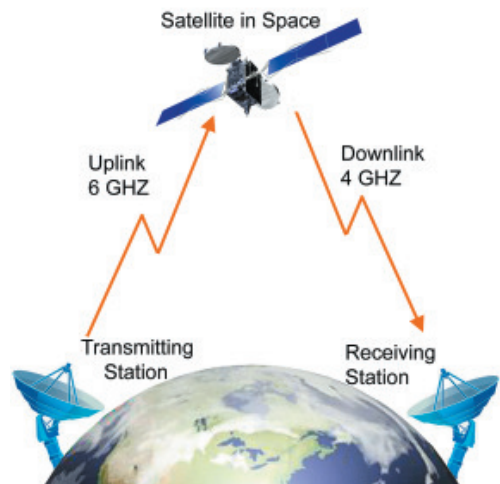
to the earth is called **downlink**. There are multiple micro wave frequency bands which are used for satellites links. Frequency used for uplink varies from 1.6 GHz to 30.0 GHz and that for downlink varies from 1.5 GHz to 20.0 GHz. Downlink frequency is always lower than the uplink frequency.

The satellite system is very expensive, but its coverage area is very large. Communication satellites are normally owned by governments or by government approved organisations of various countries.

Characteristics of transmission using satellite link

- Satellites cover a large area of the earth.
- This system is expensive.
- Requires legal permission and authorisation.

Check yourself

1. Name the basic elements needed for a data communication system.
2. Define resource sharing.
3. Name two classifications of communication channels between computers in a network.
4. Name the connecter used to connect UTP/STP cable to a computer.
5. The cable media that use light to transmit data signals to very long distances is _____.
6. AM and FM radio broadcast and mobile phones make use of _____ medium for transmission.
7. A short range communication technology that does not require line of sight between communicating devices is _____.
8. A communication system that is very expensive, but has a large coverage area when compared to other wireless communication systems is _____.

8.4 Data communication devices

A data communication device provides an interface between computer and the communication channel. These devices are used to transmit, receive, amplify and route data signals across a network through various communication media.

8.4.1 Network Interface Card (NIC)

Network Interface Card (NIC) is a device that enables a computer to connect to a network and communicate. It provides hardware interface between a computer and a

network. It can be a separate circuit board that is installed in a computer or a circuit already integrated with the motherboard. NIC can prepare, send, receive and control data on the network. It breaks up data into manageable units, translates the protocols of the computer to that of the communication medium and supplies address recognition capabilities.

Fig. 8.15 (a) : NIC card   *Fig. 8.15 (b) : Wireless NIC card*

Figure 8.15 (a, b) shows the NIC card and wireless NIC card. Some NIC cards have wired connections (Ethernet), while others are wireless (Wi-Fi). Ethernet NICs include jacks for network cables, while Wi-Fi NICs contain built-in transmitters / receivers (transceivers) and an antenna. NICs can transfer data at a speed of 1 Gbps.

8.4.2 Hub

A hub is a device used in a wired network to connect computers/devices of the same network. It is a small, simple, passive and inexpensive device (refer Figure 8.16). Computers/devices are connected to ports of the hub using Ethernet cable. When NIC of one computer sends data packets to hub, the hub transmits the packets to all other computers connected to it. Each computer is responsible for determining its data packets. The computer for which the data packets are intended accepts it. Other computers on the network discards these data packets. The main disadvantage of hub is that it increases the network traffic and reduces the effective bandwidth, as it transmits data packets to all devices connected to it.

Fig. 8.16 : Hub

8.4.3 Switch

A switch is an intelligent device that connects several computers to form a network. It is a higher performance alternative to a hub. It looks exactly like a hub. Switches are capable of determining the destination and redirect the data only to the intended node. Switch performs this by storing the addresses of all the devices connected to it in a table. When a data packet is send by one device, the switch reads the destination address on the packet and transmits the packet to the destination device with the help of the table. A switch performs better than a hub on busy networks, since it generates less network traffic in delivering messages.

8.4.4 Repeater

A repeater is a device that regenerates incoming electrical, wireless or optical signals through a communication medium (refer Figure 8.17). Data transmissions through wired or wireless medium can travel only a limited distance as the quality of the signal degrades due to noise. Repeater receives incoming data signals, amplifies the signals to their original strength and retransmits them to the destination.

Fig. 8.17 : Wireless repeater

8.4.5 Bridge

A bridge is a device used to segmentise a network. An existing network can be split into different segments and can be interconnected using a bridge. This reduces the amount of traffic on a network. When a data packet reaches the bridge, it inspects the incoming packet's address and finds out to which side of the bridge it is addressed (to nodes on the same side or the other side). Only those packets addressed to the nodes on the other side, will be allowed to pass the bridge. Others will be discarded. The packet that passes the bridge will be broadcast to all nodes on the other side and is only accepted by the intended destination node. Figure 8.18 shows the function of a bridge.
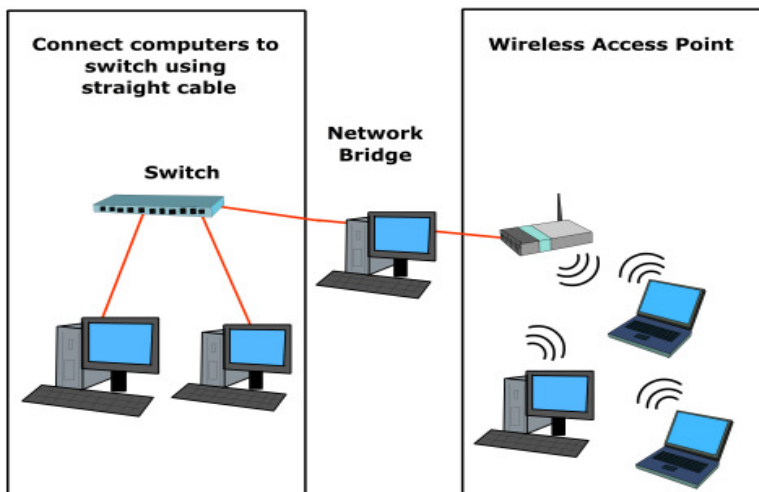
Fig. 8.18 : Bridge

8.4.6 Router

A router is a device that can interconnect two networks of the same type using the same protocol. It can find the optimal path for data packets to travel and reduce the amount of traffic on a network. Even though its operations are similar to a bridge, it is more intelligent. The router can check the device address and the network address and can

use algorithms to find the best path for packets to reach the destination. Figure 8.19 shows the role of a router.
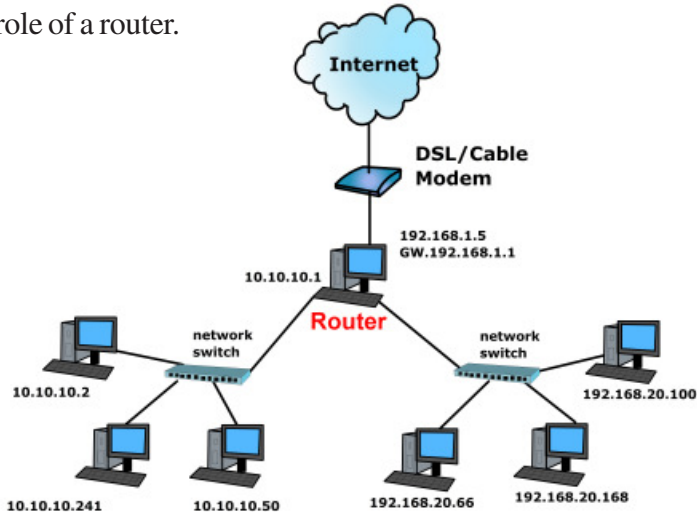

Fig. 8.19 : Router

8.4.7 Gateway

A gateway is a device that can interconnect two different networks having different protocols (refer Figure 8.20). It can translate one protocol to another protocol. It is a network point that acts as an entrance to another network. Its operations are similar to that of a router. It can check the device address and the network address and can use algorithms to find the best path for packets to reach the destination. Further, while interconnecting two networks with different protocols, there must be some mutual understanding between the networks. A gateway is capable of understanding the address structure used in different networks and seamlessly translate the data packet between these networks.
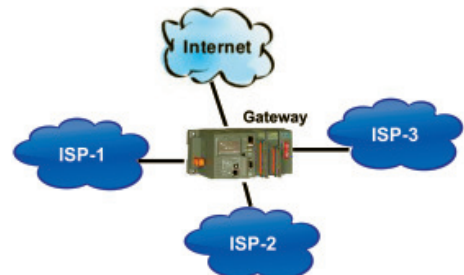
Fig. 8.20 : Gateway

8.5 Data terminal equipments

A data terminal equipment (DTE) is a device that controls data flowing to or from a computer. It is connected to the transmission medium at the end of a telecommunication link. Here we discuss the most commonly used DTEs - modem and multiplexer.

8.5.1 Modem

A modem is an electronic device used for communication between computers through telephone lines (refer Figure 8.21). The name is formed from the first three letters of the two words modulator and demodulator. It converts digital signals received from a computer to analog signals for telephone lines. It also converts the analog signals received
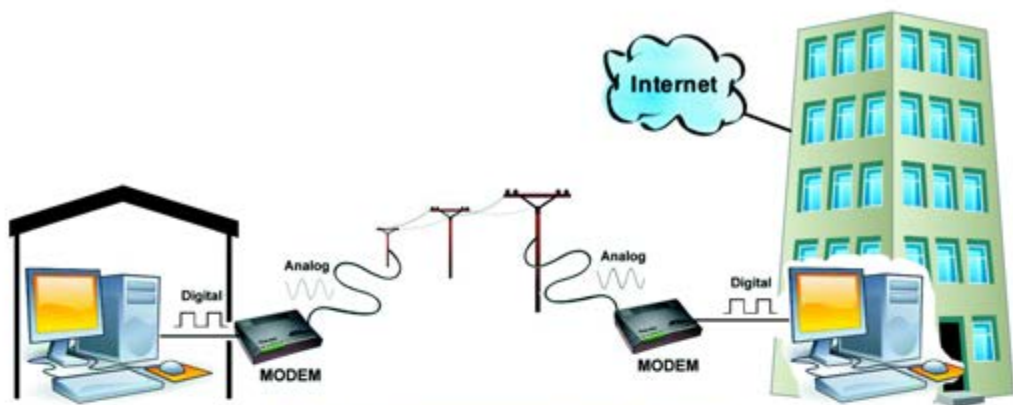
Fig. 8.21 : Communication using Modem

back from telephone lines to digital signals for the computer. The speed of the modem determines how fast it can send and receive information through telephone lines. Speed of modem is measured in bits per second (bps).

8.5.2 Multiplexer/Demultiplexer

Have you ever wondered how 200 or more TV channels are transmitted through a single cable in a television network? It is called multiplexing. Similar is the case with data transmission over networks. Multiplexing is sending multiple signals on a physical medium at the same time in the form of a single, complex signal and then recovering the separate signals at the receiving end. Multiplexing divides the physical medium into logical segments called frequency channels. Multiplexer combines (multiplexes) the inputs from different sources and sends them through different channels of a medium. The combined data travels over the medium simultaneously. At the destination, a demultiplexer separates (demultiplexes) the signal and sends them to destinations. Figure 8.22 shows the function of a multiplexer and de-multiplexer.
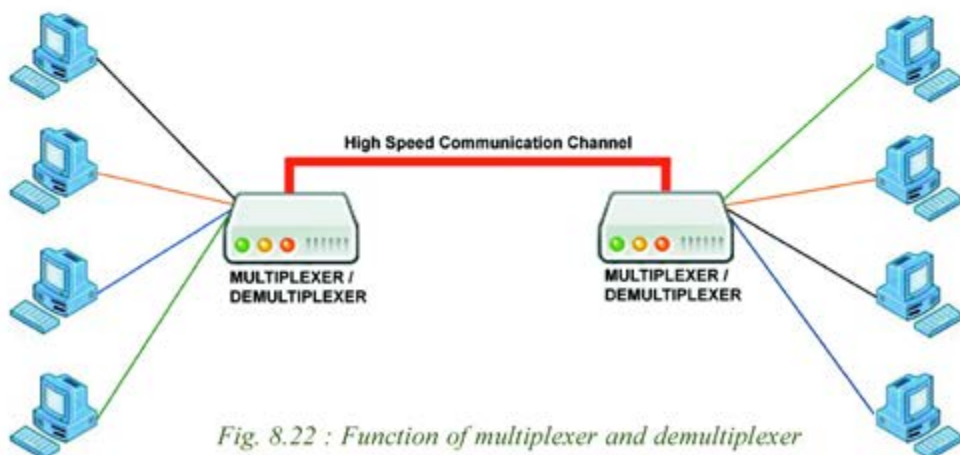

Fig. 8.22 : Function of multiplexer and demultiplexer

Check yourself

1. Compare hub and switch.
2. What is the use of a repeater?
3. The devices used to interconnect two networks of same type is _____.
4. Differentiate between router and bridge.
5. A device that can interconnect two different networks having different protocols is _____.
6. An electronic device used for communication between computers through telephone lines is _____.

8.6 Network topologies

Imagine that we have ten computers and we need to interconnect them to form a network. What are the ways by which we can interconnect them?

Using available media and based on certain conditions, there are different ways of interconnecting the nodes. The way in which the nodes are physically interconnected to form a network is called a **Topology.** Major topologies are bus, star, ring and mesh.

8.6.1 Bus topology

In bus topology (refer Figure 8.23) all the nodes are connected to a main cable called bus. If a node has to send data to another node, it sends data to the bus. The signal travels through the entire length of the bus. All nodes check the bus, and only the node for which data is addressed accepts it. A small device called terminator is attached at each end of the bus. When
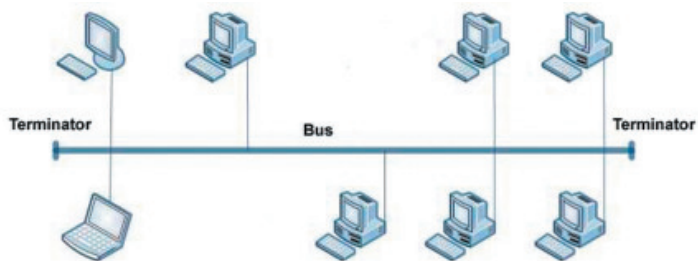

Fig. 8.23 : Bus topology

the signal reaches the end of the bus, the terminator absorbs the signal and removes it from the bus. Now the bus is free to carry another signal. This prevents the reflection of a signal back on the cable and hence eliminates the chances of signal interference. The process of transmitting data from one node to all other nodes is called broadcasting.

Characteristics of bus topology

- Easy to install.
- Requires less cable length and hence it is cost effective.
- Failure of a node does not affect the network.
- Failure of cable (bus) or terminator leads to a break down of the entire network.
- Fault diagnosis is difficult.
- Only one node can transmit data at a time.

8.6.2 Star topology

In star topology each node is directly connected to a hub/switch as shown in Figure 8.24. If any node has to send some information to any other node, it sends the signal to the hub/switch. This signal is then broadcasted (in case of a hub) to all the nodes but is accepted only by the intended node. In the case of a switch, the signal is sent only to the intended node.
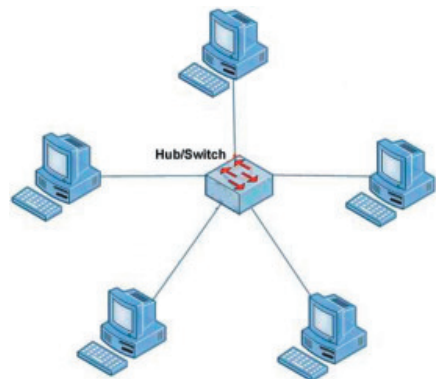

Fig. 8.24 : Star topology

Characteristics of star topology

- More efficient compared to bus topology.
- Easy to install.
- Easy to diagnose faults.
- Easy to expand depending on the specifications of central hub/switch.
- Failure of hub/switch leads to failure of entire network.
- Requires more cable length compared to bus topology.

8.6.3 Ring topology

In ring topology, all nodes are connected using a cable that loops in a ring or circle. A ring topology is in the form of a circle that has no start and no end (refer Figure 8.25). Terminators are not necessary in a ring topology. Data travels only in one direction in a ring. While they are passed from one node to the next, each node regenerates the

signal. The node for which the signal is intended reads the signal. After travelling through each node, the signal reaches back to the sending node from where it is removed.

Characteristics of ring topology

- No signal amplification is required as each node amplifies the signal.

- Requires less cable length and hence is cost effective.

- If one node fails, entire network will fail.

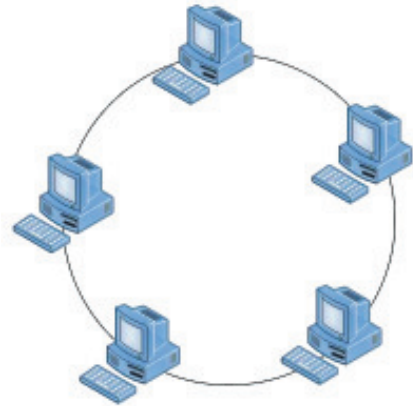- Addition of nodes to the network is difficult.

Fig. 8.25 : Ring topology

8.6.4 Mesh topology

In mesh topology, every node is connected to other nodes. So there will be more than one path between two nodes as shown in Figure 8.26. If one path fails, the data will take another path and reach the destination.

Characteristics of mesh topology

- Network will not fail even if one path between the nodes fails.

- Expensive because of the extra cables needed.
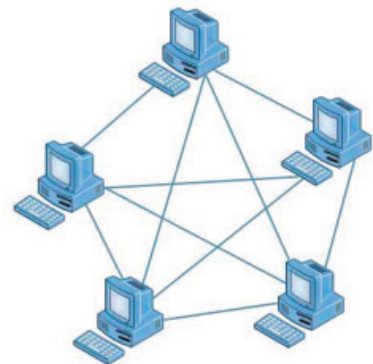
- Very complex and difficult to manage.

Fig. 8.26 : Meshopology

> *Identify the network topology used in your school lab.*
>
> **Let us do**

8.7 Types of networks

A computer network may span any amount of geographical area. It can be on a table, in a room, in a building, in a city, in a country, across continents or around the world. On the basis of the area covered, computer networks are classified as:

- PAN - Personal Area Network

- LAN - Local Area Network

- MAN - Metropolitan Area Network

- WAN - Wide Area Network

8.7.1 Personal Area Network (PAN)

PAN is a network of communicating devices (computer, mobile, tablet, printer, etc.) in the proximity of an individual. It can cover an area of a radius with few meters (refer Figure 8.27).

When we transfer songs from one cell phone to another or from a PC to an MP3 player, a PAN is set up between the two. PAN can be set up using guided media (USB) or unguided media (Bluetooth, infrared).

Fig. 8.27 : Personal Area Network

8.7.2 Local Area Network (LAN)

LAN is a network of computing and communicating devices in a room, building, or campus. It can cover an area of radius with a few meters to a few Kilometers. A networked office building, school or home usually contains a single LAN, though sometimes one building can contain a few small LANs (Like some schools have independent LANs in each computer lab) as shown in Figure 8.28. Occasionally a LAN can span a group of nearby buildings. In addition to operating in a limited space, a LAN is owned, controlled and managed by a single person or an organisation.

LAN can be set up using wired media (UTP cables, coaxial cables, etc.) or wireless media (infrared, radio waves, etc.). If a LAN is setup using unguided media, it is known as WLAN (Wireless LAN).
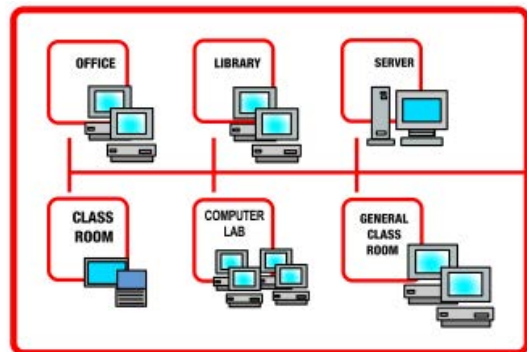

Fig. 8.28 : Local Area Network

8.7.3 Metropolitan Area Network (MAN)

MAN is a network of computing and communicating devices within a city. It can cover an area of a few Kilometers to a few hundred Kilometers radius. MAN is usually formed by interconnecting a number of LANs and individual computers. All types of communication media (guided and unguided) are used to set up a MAN. MAN is

typically owned and operated by a single entity such as a government body or a large corporation (refer Figure 8.29).
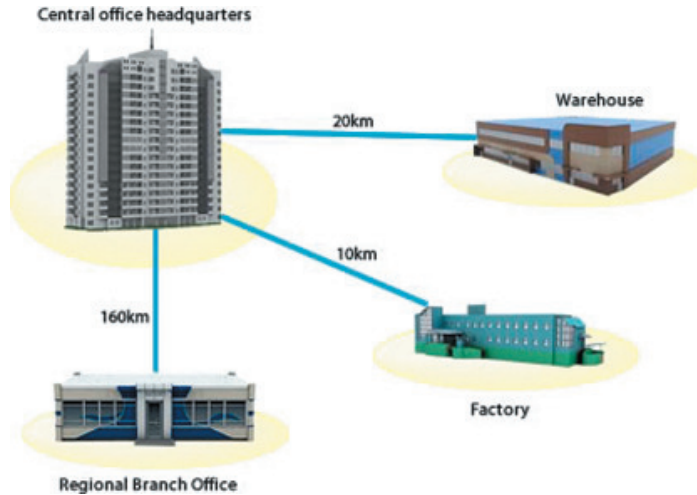

Fig. 8.29 : Metropolitan Area Network

8.7.4 Wide Area Network (WAN)

WAN is a network of computing and communicating devices crossing the limits of a city, country or continent. It can cover an area of over hundreds of Kilometers in radius. WAN usually contain a number of interconnected individual computers, LANs, MANs and maybe other WANs. All types of communication media (guided and unguided) are used to set up a WAN as shown in Figure 8.30. The best known example of a WAN is the Internet. Internet is considered as the largest WAN in the world. A network of ATMs, banks, government offices, international organisations, offices, etc. spread over a country, continent or covering many continents are examples of WAN.
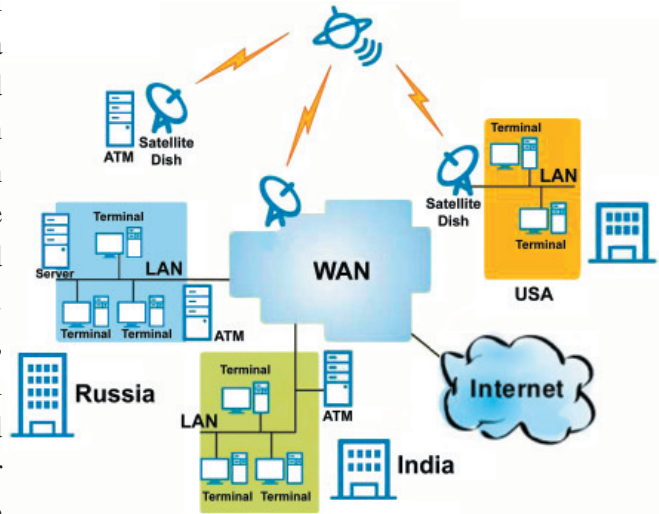

Fig. 8.30 : Wide Area Network

Table 8.1 summerises the characteristics of PAN, LAN, MAN and WAN

Parameter	PAN	LAN	MAN	WAN
Area covered	Small area (Up to 10 m radius)	A few meters to a few Kilometers (Up to 10 Km radius)	A city and its vicinity (Up to 100 Km radius)	Entire country, continent, or globe
Transmission speed	High speed	High speed	Moderate speed	Low speed
Networking cost	Negligible	Inexpensive	Moderately expensive	Expensive

Table 8.1 : Characteristics summary of PAN, LAN, MAN, WAN

8.8 Logical classification of networks

This classification is based on the role of computers in the network and division falls into two categories: peer-to-peer and client-server.

8.8.1 Peer-to-Peer

A peer-to-peer network has no dedicated servers. Here a number of computers are connected together for the purpose of sharing information or devices. All the computers are considered equal. Any computer can act as a client or as a server at any instance. This network is ideal for small networks where there is no need for dedicated servers, like home network or small business establishments or shops.

8.8.2 Client-Server

The client-server concept is the driving force behind most of the networks. It is similar to going to a restaurant, reading the menu, calling the waiter (server) and then ordering one's preference from the menu. If the ordered item is available in the restaurant at that time, it is 'served' to whom the order was placed (client), else the request is refused.

In a network, the client-server architecture consists of high-end computer (called server) serving lower configuration machines called clients. A server provides clients with specific services (responses) upon client's request. The services include sharing of data, software and hardware resources. Figure 8.31 shows the general client-server architecture.

Client-server architecture is an example for centralised software management. When software is loaded on the server and shared among the clients, changes made to the software in the server will reflect in the clients also. So there is no need to spend time and energy for installing updates and tracking files independently on the clients.

Fig. 8.31 : Client - Server

Classifications for servers are

a) **File server** - A computer that stores and manages files for multiple users on a network.

b) **Web server** - A computer dedicated to responding to requests for web pages.

c) **Print server** - Redirects print jobs from clients to specific printers.

d) **Database server** - Allows authorised clients to view, modify and/or delete data in a common database.

Check yourself

1. In bus topology, when the signal reaches the end of the bus, _____absorbs the signal and removes it from the bus.

2. In which topology is every node connected to other nodes?

3. Categorise and classify the different types of networks given below.
   ATM network, Cable television network, Network within the school, Network at home using bluetooth,Telephone network, Railway network.

4. What is PAN?

8.9 Identification of computers over a network

Imagine that you are in India and you wish to write a letter to your friend in America. What would you do? You write a letter, put it in an envelop, write your friend's address on it and write your address on the back. When the letter is posted in a post office in India, it is stamped with a unique seal and date. After going through a feasible route the letter reaches the post office in America, where it is stamped again with a unique seal and date. Then, the postman makes sure that it reaches the specified addressee. In a network, data is sent in the form of packets in a similar way.

Once a network has been set up, the nodes can communicate among themselves. But for proper communication, the nodes should be uniquely identifiable. If node X sends some information to node Y on a network, then it is mandatory that nodes X and Y are uniquely identifiable on the network. Let us see how this is achieved.

8.9.1 Media Access Control (MAC) address

A Media Access Control (MAC) address is a universally unique address (12 digit hexadecimal number) assigned to each NIC (Network Interface Card) by its

manufacturer. This address is known as the MAC address. It means that a machine with an NIC can be identified uniquely through the MAC address of its NIC. MAC address of an NIC is permanent and never changes.

MAC addresses are 12-digit hexadecimal (or 48 bit Binary) numbers. By convention, MAC addresses are usually written in one of the following two formats:

MM : MM : MM : SS : SS : SS   or   MM – MM – MM – SS – SS – SS

The first half (MM:MM:MM) of a MAC address contains the ID number of the adapter manufacturer. The second half (SS:SS:SS) of a MAC address represents the serial number assigned to the adapter (NIC) by its manufacturer. For example, in the following MAC address,

<div align="center">

00:A0:C9 : 14:C8:35

Fig: 8.32 : MAC Id

</div>

the prefix 00:A0:C9 indicates that the manufacturer is Intel Corporation. The last three numbers 14:C8:35 are given by the manufacturer (Intel in this example) to this NIC.

8.9.2 Internet Protocol (IP) Address

An IP address is a unique 4 part numeric address assigned to each node on a network, for their unique identification. IP address is assigned to each machine by the network administrator or the Internet Service Provider. An IP address is a group of four bytes (or 32 bits) each of which can be a number from 0 to 255.

To make it easier for us to remember, IP addresses are normally expressed in decimal format as a "dotted decimal number" as indicated in Figure 8.33.

On a network, the IP address of a machine is used to identify it. IP protocol identifies a machine with its IP address to route the packets.

10010100	01001110	11111010	00001100

148 . 78 . 250 . 12

Fig. 8.33 : IP address

There are two versions of IP addresses: version 4 (IPv4) and version 6 (IPv6). IPv4 uses 32 bits and IPv6 uses 128 bits for an IP address. Using IPv4 only 2^{32} (approximately 4 billion) distinct devices can be addressed.

As the number of devices which need to be networked (mobile phones, home appliances, personal communication devices, etc.) is increasing at a very fast pace, IPv4 addresses are being exhausted. To address this problem IPv6 was developed and it is now being deployed. Using IPv6, 2^{128} (approximately 4 billion × 4 billion × 4 billion × 4 billion) distinct devices can be addressed.

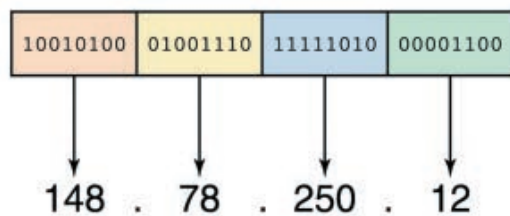*Identify the IP and MAC Id of each networked machine in your school and prepare a table as follows. (Use ipconfig /all at command prompt).*

Sl No	Computer Name	IP	MAC
1.			
2.			
3.			

8.10 Network protocols

A network protocol is the special set of rules to be followed in a network when devices in the network exchange data with each other. Each protocol specifies its own rules for formatting data, compressing data, error checking, identifying and making connections and making sure that data packets reach its destination.

Several computer network protocols have been developed for specific purposes and environments. Some commonly used protocols are TCP/IP, SPx/IPx etc.

TCP/IP

TCP/IP, Transmission Control Protocol/Internet Protocol, is a suite of communications protocols used to interconnect network devices on the local networks and the Internet. TCP/IP defines rules for how electronic devices (like computers) should be connected to the Internet and how data should be transmitted between them.

When data is to be sent from one computer to another over Internet, it is first broken into smaller packets by TCP and then sent. When these packets are received by the receiving computer, TCP checks packets for errors. If errors are found, TCP submits requests for retransmission; else packets are assembled into the original message according to the rules specified in TCP protocol. Figure 8.34 shows the steps involved
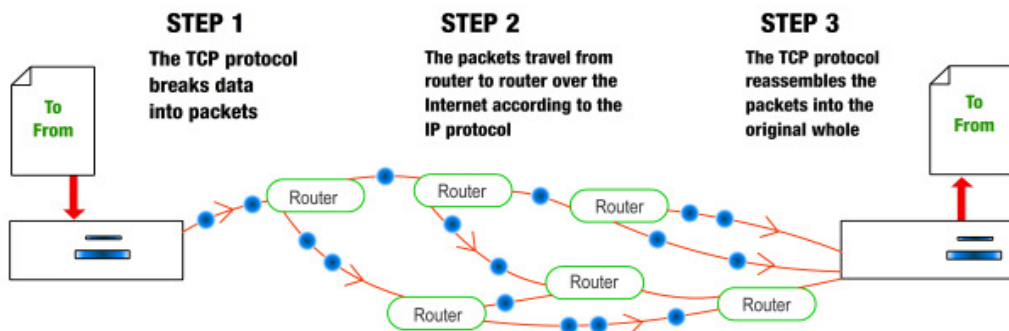

Fig: 8.34 : How TCP/IP works

in the working of TCP/IP protocol. Delivery of each of these packets to the right destinations is done by Internet protocol (IP). Even though different packets of the same message may be routed differently, they will reach the same destination and get reassembled there. HTTP, FTP and DNS are three sub protocols of TCP/IP protocol suite.

HTTP

HTTP stands for Hypertext Transfer Protocol. It is a standard protocol for transferring requests from client-side and to receive responses from the server-side. The HTTP client (browser) sends a HTTP request to the HTTP server (web server) and server responds with a HTTP response. This pair of request and response is called an HTTP session (refer Figure 8.35).

The response from the server can be static such as a file already stored on the server, or dynamic, such as, the result of executing a piece of code by the server as per the request from the client.

Fig. 8.35 : An HTTP session

The two important characteristics of HTTP are

- HTTP is transmission medium independent.
- HTTP is stateless (The server and client are aware of each other only during a request or response. Afterwards, each forgets the other).

FTP

FTP stands for File Transfer Protocol. It is a standard for exchanging of data and program files across a network. FTP is the easiest way to transfer files between computers via the Internet. It uses TCP and IP to perform uploading and downloading. A FTP client program installed in the computer can help in the uploading (sending files to another computer) and downloading (receiving files from another computer) of files.

FTP uses client-server architecture in servers with security features, username and password protection for file transfer. An FTP client program (Filezilla, Cute FTP, etc) installed in the computer can help in the easy uploading and downloading of files.

DNS

DNS stands for Domain Name System. DNS returns the IP address of the domain name, that we type in our web browser's address bar. (like mobile phone automatically dialing the phone number when we select a name from contact list).

The DNS system has its own network. DNS implements a database to store domain names and IP address information of all web sites on the Internet. DNS assumes that IP

addresses do not change (statically assigned). If one DNS server does not know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.

Find and prepare notes on five protocols other than TCP/IP, HTTP, FTP, DNS.

Let us do

8.11 Uniform Resource Locator (URL)

URL stands for Uniform Resource Locator. URL is a formatted text string used by web browsers, email clients and other software to identify a network resource on the Internet. Every resource on the Internet has a unique URL. Network resources are files that can be plain web pages, other text documents, graphics, programs, etc.

URL consists of letters, numbers and punctuations. URL string can be divided into three parts.

a)   Network protocol (also called the scheme)

b)   Domain name (Host name or address)

c)   File name

For example, the URL http://www.dhsekerala.gov.in/index.html has three parts as shown in Figure 8.36. The detailed description of these three parts are given below:
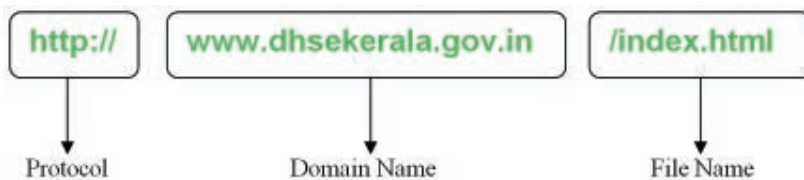
http://	www.dhsekerala.gov.in	/index.html
Protocol	Domain Name	File Name

Fig. 8.36 : URL of DHSE web site

a. Protocol

The protocol enables the browser to know what protocol is used to access the information specified in the domain.

b. Domain name

Domain name is the name assigned to a server through the Domain Name System (DNS). Domain names are used in URLs to identify the particular web server. Domain names provide Internet users with a short name that is easy to remember. Whenever we have to communicate with a computer on Internet, we can do so by using its IP address. But it is practically impossible for a person to remember the IP addresses of all the computers, one may have to communicate with. Therefore, a system has been

developed which assigns names to computers (web servers) and maintains a database of these names and their corresponding IP addresses. These names are called domain names.Examples of some domain names are dhsekerala.gov.in, keralaresults.nic.in, google.com, gmail.com, etc.

A domain name usually has more than one part: top level domain name or primary domain name and sub-domain name(s). For example, in the domain name above, 'in' is the primary domain name; 'gov' is the sub-domain of in and 'dhsekerala' is the sub-domain of 'gov'. There are only a limited number of top level domains and these are divided into two categories: Generic Domain Names and Country-Specific Domain Names. Examples of generic and country specific domain names are given in Table 8.2.

Generic Domain Names		Country Specific Domain Names	
·com	Commercial business	·in	India
·edu	Educational institutions	·au	Australia
·gov	Government agencies	·ca	Canada
·mil	Military	·ch	China
·net	Network organizations	.jp	Japan
·org	Organizations (nonprofit)	.us	United States of America

Table. 8.2 : Generic and country specific domain names

c. File name

It is the file to be opened. In the example given in Figure 8.35, 'index.html' is the file that is to be accessed from the web server specified by the domain name.

> *Make a list of valid URL, containing two examples for each generic domain name and country specific domain name. Also note down the file name opened by default (for file look at the URL in address bar after the site is opened).*
>
> **Let us do**

Let us sum up

We learned about computer networks, the essential technology of the century, in this chapter. Importance of network was discussed by highlighting the various advantages it provides. We discussed the various communication media and their pros and cons. The devices used at various situations while forming a network was also discussed. Before discussing the types of network, we learned the different ways a network could be formed by discussing various topologies. We then discussed protocol and how TCP/IP send/receive data across the network. Methods to uniquely identify a node in the network were introduced and finally we concluded discussing URL.

Learning outcomes

After the completion of this chapter the learner will be able to

- identify and choose a communication medium.
- compare different types of network.
- explain the logical classification of networks.
- identify how data is send across networks.
- design small networks.
- explain how a node is identified in a network.
- identify the various parts of a URL.

Sample questions

Very short answer type

1. The transmission media which carry information in the form of light signals is called
   _____.
   a. Coaxial   b. Twisted   c. WiFi   d. Optical Fiber

2. Different networks with different protocols are connected by a device called _____.
   a. Router   b. Bridge   c. Switch   d. Gateway

3. In which topology, the failure of any one computer will affect the network operation?

4. To transmit signals from multiple devices through a single communication channel
   simultaneously, we use _____ device.
   a. Modem   b. Switch   c. Router   d. Multiplexer

5. Satellite links are generally used for
   a. PANs   b. LANs   c. MANs   d. All of the above

Short answer type

1. Define bandwidth.
2. Switch and Hub are two devices associated with networking. Differentiate them.
3. What is an IP address? Give an example of IP address.
4. What is TCP/IP? What is its importance?
5. What is Bluetooth?
6. Distinguish between router and gateway.
7. Explain the need for establishing computer networks.
8. What are the uses of computer networks?
9. What is the limitation of microwave transmission? How is it eliminated?
10. Briefly describe the characteristics of Wi-Fi.

11. An International School is planning to connect all computers, spread over distance of 45 meters. Suggest an economical cable type having high-speed data transfer, which can be used to connect these computers.

12. Suppose that you are the administrator of network lab in one Institution. Your manager directed you to replace 10 Mbps switch by 10 Mbps Ethernet hub for better service. Will you agree with this decision? Justify your answer.

13. You need to transfer a biodata file stored in your computer to your friend's computer that is 10 kms away using telephone network
    a. Name the device used for this at both ends.
    b. Explain how the file is send and received inside the device, once a connection between two computers is established.

14. When is a repeater used in a computer network?

15. Compare infrared and Bluetooth transmission.

16. Identify and explain the device used for connecting a computer to a telephone network.

17. Briefly explain LAN topologies.

18. Briefly describe TCP/IP protocol.

19. What is a MAC address? What is the difference between a MAC address and an IP address?

Long answer type

1. How are computer networks classified, based on size?

2. Compare different LAN topologies.

3. Explain various types of guided communication channels.

4. Compare different unguided media.

5. Define the term protocol. Briefly describe any two communication protocols.

6. Briefly describe the various communication devices used in computer networks.

7. Which is/are communication channel(s) suitable in each of the following situations?
    a. Setting up a LAN.
    b. Transfer of data from a laptop to a mobile phone.
    c. Transfer of data from one mobile phone to another.
    d. Creating a remote control that can control multiple devices in a home.
    e. Very fast communication between two offices in two different countries.
    f. Communication in a hilly area.
    g. Communication within a city and its vicinity where cost of cabling is too high.

9

Key concepts

- **History of the Internet**
- **Connecting the computer to the Internet**
- **Types of connectivity**
  - o Dial up
  - o Wired broadband
  - o Wireless broadband
  - o Internet access sharing methods
- **Services on Internet**
  - o www
  - o Search engines
  - o E-mail
  - o Social media
- **Cyber security**
  - o Computer virus
  - o Worm
  - o Trojan horse
  - o Spams
  - o Hacking
  - o Phishing
  - o Denial of service attack
  - o Man-in-the-middle attack
- **Preventing network attacks**
  - o Firewall, antivirus scanners, cookies
- **Guidelines for using computers over Internet**

Internet

In the previous chapter we saw that the Internet is the largest computer network in the world. We use the Internet to check SSLC results, to submit applications for higher secondary school admissions and check its status, to get information about various types of scholarships and to submit applications for receiving them, etc. Can you imagine life without the Internet today? It would be difficult for us to manage all the above tasks without it. Internet has definitely made life easier for us. It has influenced our daily life to a great extent. Because of its wide popularity and increase in use even the television sets with facilities for Internet connectivity have come up in markets.

People generally use the Internet to search information, access e-mails, make bill payments, for online shopping, online banking, to connect with people in social networking sites, etc. The reach of Internet is very vast and it helps in reducing cost and time. Issues like online intrusion to privacy, online fraud, cyber-attacks, etc. are becoming common now. Apart from the Internet and its access methods, let us discuss the various services provided by the Internet like search engines, e-mail, social media and about the threats and preventive measures while using Internet in this chapter.

9.1 History of the Internet

The Internet started as a small network through a project by the United States Department of Defence by the name ARPANET (Advanced Research Projects Agency Network). During 1970s this military network was connected to the computers of Universities and companies that worked for the Department of Defence. In 1984 the military network split from ARPANET to form MILNET to be used by the American military only. ARPANET which used TCP/IP protocol for communication was thereafter used for scientific research and information sharing. Later, several other networks merged with ARPANET to form a large network. ARPANET is considered as the first wide area network (WAN).Vinton Gray Cerf who was instrumental in the development of Internet and TCP/IP protocol, is considered as the father of Internet.

Vinton Gray Cerf (1943 -) popularly called Vint Cerf, an American computer scientist, is widely known as 'Father of the Internet'. He was instrumental in the initial development of Internet along with his fellow American computer scientist Bob Kahn. He worked for the United States Department of Defence Advanced Research Projects Agency (DARPA) and had a key role in the development of TCP/IP protocol. He was also involved in the formation of ICANN.
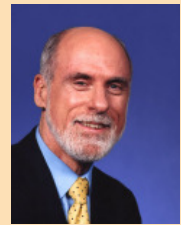
In 1989, Tim Berners Lee, a researcher, proposed the idea of World Wide Web (WWW). Tim Berners Lee and his team are credited with inventing Hyper Text Transfer Protocol (HTTP), HTML and the technology for a web server and a web browser. Using hyperlinks embedded in hypertext the web developers were able to connect web pages. They could design attractive webpages containing text, sound and graphics. This change witnessed a massive expansion of the Internet in the 1990s.

Fig. 9.1 : Tim Berners Lee (1955 -)

Various types of computers loaded with diverse operating systems in different organisations at geographically distant locations joined this network making it a global phenomenon.TCP/IP protocol is used as the communication protocol for Internet. Any computer that joins Internet should follow the TCP/IP protocol. In 1998, Internet Corporation for Assigned Names and Numbers (ICANN) was established. ICANN does not control the Internet content; rather it develops policies on the Internet's Uniform Resource Locators (URL). Today, Internet is the largest public network that connects billions of computers all over the world and provides several services like searching, e-mail, file transfer, social networking,

etc. The *Internet* is an interconnected system of computer networks that serves the users all over the world.

An *intranet* is considered as a private computer network similar to Internet that uses TCP/IP protocol to share information, software or services within an organisation. An intranet can host websites, provide e-mail service, file transfer and other services available on Internet.

When an intranet is made accessible to some computers that are not part of a company's private network it is called an *extranet*. A network that allows vendors and business partners to access a company resource can be considered as an example of extranet.

9.2 Connecting the computer to the Internet

As we know today, the Internet has become very popular and almost all organisations and people around the world are joining it. Earlier, people used the Internet to search for information and check e-mails only, but today it is used to book train tickets, recharge mobile phones, Internet banking and a lot more. Therefore almost all of us require an Internet connection in our computers or mobile devices.

The following are the hardware and software requirements for connecting a computer to the Internet:

- A computer with Network Interface Card (wired/wireless) facility and an operating system that supports TCP/IP protocol

- Modem

- Telephone connection

- An Internet account given by an Internet Service Provider (ISP)

- Software like browser, client application for e-mail, chat, etc.

Nowadays desktop computers or laptops are not the only devices that we use to connect to the Internet. People have also started using tablets, smart phones, etc. to browse the Internet. Some of these devices come with built-in modems, whereas others use a wireless dongle or wireless connection from a modem to access the Internet.

9.3 Types of connectivity

Today most websites use images and multimedia content to make webpages more attractive. Several websites provide videos that can be downloaded or viewed on the Internet. Instead of distributing software in CDs or other storage media, it is now distributed online by various vendors. The latest trend shows that software like word processors, spreadsheets, antivirus, etc. are used online on a rental basis instead of

installing it on each computer. In all these cases, a large volume of data is transferred online. Therefore the speed or data transfer rate of the Internet is an important aspect. *Data transfer rate* is the average number of bits transferred between devices in unit time.

1 kbps	= 1000 bits per second
1 Mbps	= 1000 kbps
1 Gbps	= 1000 Mbps

Difference between unit symbols b and B

b stands for bit

B stands for Byte

Difference between unit symbols k and K

k = 1000 = 10^3

K = 1024 = 2^{10}

Here 'k' is a decimal unit and 'K' is a binary unit of measurement. But for Mega, Giga and Tera, both decimal and binary units use 'M', 'G' and 'T' as symbols respectively. They are differentiated from the context in which they are used.

Note that data transfer rate is measured in decimal units and memory is measured in binary.

The main factor that decides Internet access speed is the type of connectivity we choose to link to the Internet. Internet connectivity is classified based on the speed of the connection and the technology used. They can be broadly classified as dial-up connectivity, wired broadband connectivity and wireless broadband connectivity. The data transfer rates of each type of connectivity may vary as techonology advances.

9.3.1 Dial-up connectivity

A dial-up connection uses the conventional telephone line and a dial-up modem to dial and connect to the server at the Internet Service Provider (ISP). Figure 9.2 shows the dial-up connectivity system. As the connection is made by dialing, it takes time to connect to the server at the ISP. This connection commonly uses a 56 kbps modem that can transmit data up to a maximum speed of 56 kbps. This slow connection is comparatively less costly when compared to other types of connections. Another disadvantage is that a dial-up connection requires exclusive use of the telephone line, i.e., while accessing Internet, we cannot make or receive telephone calls (voice calls). Nowadays, broadband connections that have a higher speed are replacing dial-up connections.
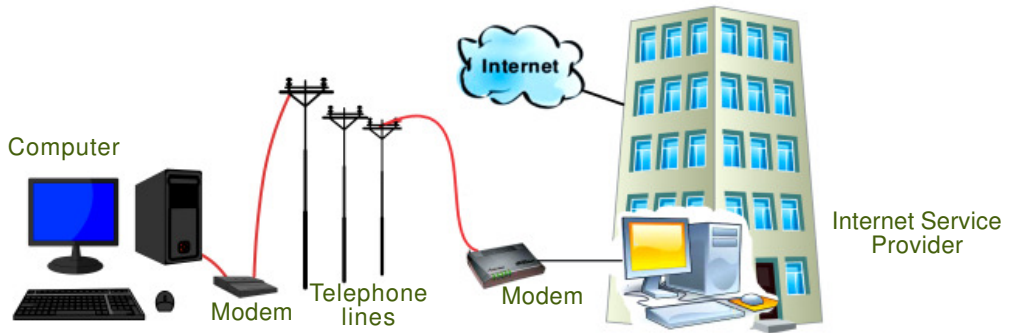
Fig. 9.2 : Dial-up connectivity

9.3.2 Wired broadband connectivity

The term broadband refers to a broad range of technologies that helps us to connect to the Internet at a higher data rate (speed). Wired broadband connections are 'always on' connections that do not need to be dialled and connected.

Fig. 9.3 : Broadband modem

Broadband connections use a broadband modem (refer Figure 9.3) and allow us to use the telephone even while using the Internet. Table 9.1 shows the comparison between dial-up and wired broadband connections.

Dial-up connection	Wired broadband connection
• Slow connection, speed upto 56 kbps	• High speed connection, speed usually higher than 256 kbps
• Requires dialing to connect to ISP	• Always on connection
• Uses telephone line exclusively	• Simultaneous use of voice and Internet
• Uses dial-up modem	• Uses broadband modem

Table 9.1 : Comparison between dial-up and wired broadband connections

Popular broadband technologies are Integrated Services Digital Network (ISDN), Cable Internet, Digital Subscriber Line (DSL), Leased Lines and Fiber to the Home (FTTH).

a. Integrated Services Digital Network (ISDN)

ISDN is a broadband service capable of transporting voice and digital data. Most ISDN lines offered by telephone companies give users two lines. The users can use one line for voice and the other for data, or they can use both lines for data. ISDN lines are capable of offering data transfer rates upto 2 Mbps.

b. Cable Internet

Cable Internet access provides Internet access using coaxial cables laid for television signal transmission to our homes. The service provider uses a cable modem at our

home to connect our computer to cable network. Cable TV systems are designed to carry large bandwidth and therefore cable Internet can provide speeds between 1 Mbps to 10 Mbps.

c. Digital Subscriber Line (DSL)

DSL is another broadband service that provides connection to the Internet through standard telephone lines. DSL allows the user to use copper telephone lines for both Internet communication and for making voice calls simultaneously. It is composed of several subcategories, the most common being Asymmetric Digital Subscriber Line (ADSL). ADSL is a communication technology that allows faster flow of information over a telephone line. The down stream speed of ADSL services typically ranges from 256 kbps to 24 Mbps. This connection requires an ADSL modem at our homes/offices. ADSL is the most popular broadband service available in India.

d. Leased Line

Leased lines are dedicated lines used to provide Internet facility to ISPs, business, and other large enterprises. An Internet leased line is a premium Internet connection that provides speed in the range from 2 Mbps to 100 Mbps and is comparatively costly. This is why leased lines are used only for connecting large campus of organisations like educational institutions to Internet.

e. Fibre To The Home (FTTH)

Fibre to the Home (FTTH) uses optical fibers for data transmission. Optical fibres are laid from the ISP to our homes. FTTH technology has been accepted worldwide to implement high speed Internet to the home. Since optical fibres are known to have high bandwidth and low error rates, they provide very high speed connectivity. A Network Termination Unit (NTU) is installed in our homes, which is connected to our computer through an FTTH modem.

9.3.3 Wireless broadband connectivity

Wireless broadband connectivity provides almost the same speed as that of a wired broadband connection. The popular wireless broadband accesses are Mobile Broadband, Wi-MAX, Satellite Broadband and Wi-Fi. Some of the wireless modems available for use to connect to Internet are shown in Figure 9.4.

Fig. 9.4 : Wireless broadband modems

a. Mobile broadband

Mobile broadband is wireless Internet access using mobile phone, USB wireless modem, tablet or other mobile devices. The modem is built into mobile phones, tablets, USB

dongles, etc. Mobile broadband offers the freedom to allow the user to access the Internet from anywhere on the move. This technology uses the cellular network of mobile phones for data transmission. The mobile technology for data transmission has been changing from 2nd Generation (2G) through 3rd Generation (3G) to the current 4th Generation (4G). The speed of data transmission increases with the progression of generations of mobile technology.

b. Wi-MAX

In the previous chapter we learned that Worldwide Interoperability for Microwave Access (Wi-MAX) is used as an alternative for wired broadband. Wi-MAX offers a Metropolitan Area Network which can provide wireless Internet upto a distance of 50 Km. Connectivity is provided using devices like Wi-MAX handsets, USB dongles, devices embedded in laptops, etc. that have a Wi-MAX modem integrated in it. This technology provides a maximum connection speed of upto 70 Mbps.

c. Satellite broadband

Satellite broadband technology is a method by which Internet connectivity is provided through a satellite. A Very Small Aperture Terminal (VSAT) dish antenna and a transceiver (transmitter and receiver) are required at the user's location. A modem at the user's end links the user's computer with the transceiver. Download speed is upto 1 Gbps for this technology. It is among the most expensive forms of broadband Internet access. They are used by banks, stock exchanges, governments, etc. and also for Internet access in remote areas.

9.3.4 Internet access sharing methods

An Internet connection can be shared among several computers using a LAN, Wi-Fi network or Li-Fi network.

a. Using LAN

The Internet connected to a computer in a Local Area Network (LAN) can be shared among other computers in the network. This can be done either using features available in the operating system or using any proxy server software available in the market. Sharing can also be done by connecting computers directly to the router using a cable.

b. Using Wi-Fi network

We have heard of Wi-Fi campuses in large educational institutions, coffee shops, shopping malls, hotels, etc. We also know that some of the broadband modems at our homes and schools offer Wi-Fi Internet access.Wi-Fi is a popular short distance data transmission technology that is used for network access, mostly Internet. Wi-Fi locations

Fig. 9.5 : Wi-Fi network

receive Internet connection through any one of the above mentioned wired or wireless broadband access methods, as discussed in the previous section. They provide us Internet connectivity through a Wi-Fi router or a wireless network access point. Such an access point, popularly called hotspot, has a range of about 100 meters indoors and a greater range outdoors. We access Internet in our Wi-Fi enabled devices like laptops, tablets, mobile phones, etc. through these hotspots. A drawback of Wi-Fi is that it is less secure than wired connections.

c. Using Li-Fi network

Li-Fi (Light Fidelity) is a fast optical version of Wi-Fi, which uses visible light for data transmission. The main component of this communication is a bright LED (Light Emitting Diode) lamp that can transmit data and a photo diode that serves as the receiver. LEDs can be switched on and off to generate a binary string of 1s and 0s. The flickering of this LED is so fast that the human eye cannot detect it. A data rate of over 100 Mbps is possible using this technique as light offers very high bandwidth. Another advantage is that since Li-Fi uses light, it can be used in aircrafts and hospitals where radio waves may cause interference. It can also be used underwater where Wi-Fi does not work. It provides greater security as light cannot penetrate walls when compared to Wi-Fi. One of the shortcomings of Li-Fi is that it works only in direct line-of-sight. In future this technology can be further developed to use light bulbs as a source of Internet.

Check yourself

1.   ARPANET stands for _____.
2.   Who proposed the idea of www?
3.   The protocol for Internet communication is _____.
4.   What do you mean by an 'always on' connection?
5.   A short distance wireless Internet access method is _____.

Prepare a comparison chart on the different methods of Internet connection.

Let us do

9.4 Services on Internet

The internet offers a variety of services. Services like WWW, e-mail, search engines, social media, etc. are widely used throughout the globe. In this section we shall discuss some of the services of Internet.

9.4.1 World Wide Web (WWW)

The World Wide Web (WWW) is a system of interlinked hypertext documents accessed via the Internet. It is a service on the Internet that uses Internet infrastructure. WWW is a huge client-server system consisting of millions of clients and servers connected together. Each server maintains a collection of documents and they can be accessed using a reference called Uniform Resource Locator (URL). These documents may contain, text, images, videos and other multimedia content. It may also contain hyperlinks to documents on different servers. Selecting a hyperlink results in a request to fetch that document/web page from the server and display it. The WWW works by establishing hypertext links between documents anywhere on the network. Clients can access the documents on the servers using software called browser. A browser is responsible for properly displaying the documents.

a. Browser

A web browser is a software that we use to retrieve or present information and to navigate through web pages in the World Wide Web. The document to be displayed is identified using a URL. A URL consists of its DNS name and the file name to be retrieved. It also specifies the protocol for transferring the document across the network. A browser is capable of displaying text, images, hypertext links, videos, sounds, scripts (program code inside a web page), etc. in a web document/page. Most of WWW documents are created using Hyper Text Markup Language (HTML) tags and are called web pages. The web browser interprets these tags and displays a formatted page. It allows us to navigate through web pages using the hyperlinks available in web pages. Some common browsers are Google Chrome, Internet Explorer, Mozilla Firefox, Opera, and Safari. Icons of some popular browsers are shown in Figure 9.6. Some of these browsers have a mobile version that can be used in mobile operating systems.

Google Chrome   Internet Explorer   Mozilla Firefox   Opera   Safari

Fig. 9.6 : Icons of popular browsers

b. Web browsing

All of us have visited web sites by entering the website address (URL) into web browsers and then using the hyperlinks in it to move through the web pages. Traversing through the web pages of World Wide Web is called web browsing. Major operations performed while web browsing are shown in Figure 9.7.
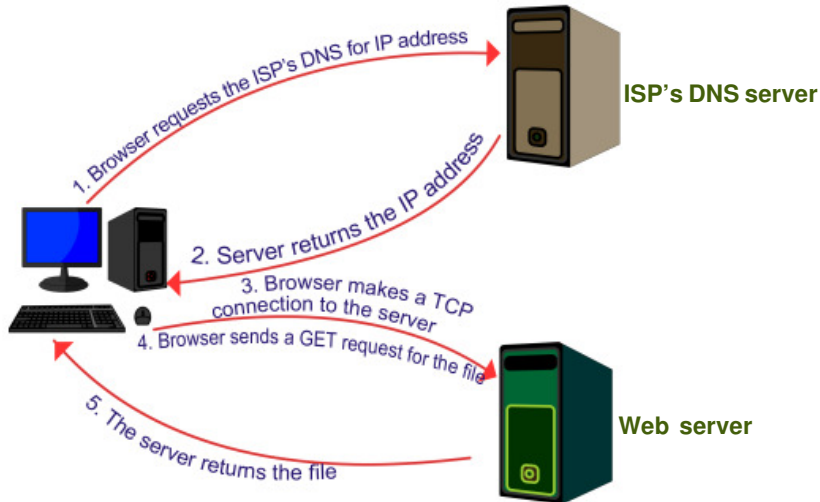

Fig. 9.7 : Web browsing

Suppose you wish to visit the website 'www.kerala.gov.in'. What will you do? You will enter this URL in the address box of the web browser and press **Enter** key. The steps a browser will follow to display a webpage may be summarised as follows.

1. The browser determines the URL (http://www.kerala.gov.in) entered.

2. The browser then sends a request to the DNS server of the user's ISP to get the IP address of the URL.

3. The ISP's DNS server replies with the IP address.

4. The browser then makes a TCP connection to the web server at the IP address (www.kerala.gov.in).

5. Then it sends a GET request for the required file (web page) to the web server.

6. The web server returns the web page.

7. The TCP connection is released.

8. The browser processes the contents of the webpage and displays it.

9.4.2 Search engines

There are millions of pages available on the Internet that contain information on a variety of topics. But it is very difficult to search for a topic in this large collection of web pages. Internet search engine websites are special programs that are designed to help people to find information available in World Wide Web. Search engine programs search documents available on World Wide Web for specified keywords and return a list of the documents/web pages matching the keywords.

Let us discuss the technology behind these websites. Search engine web sites use programs called web crawlers or spiders or robots to search the web. Web crawlers search the web pages stored in the different web servers and find possible keywords. The search engine website stores these keywords along with their URLs to form an index in the search engine's web servers. When we use the search engine website to search a particular topic (keyword), it does not search the World Wide Web. It only searches the index, which the web crawler programs have created in the search engine's web server for the topic/keyword. Search engines
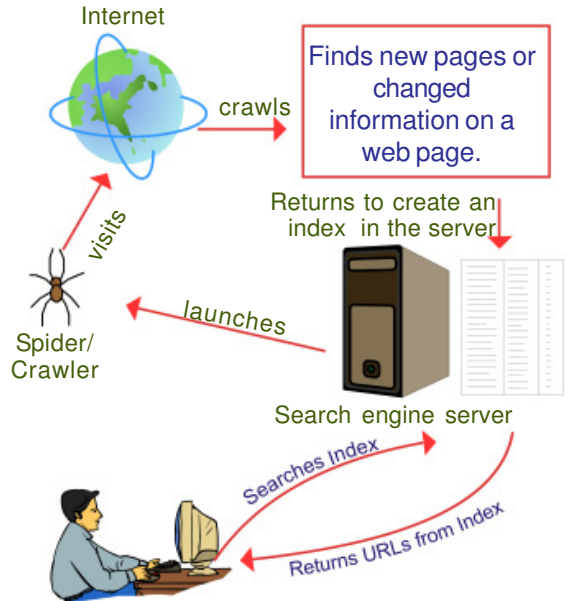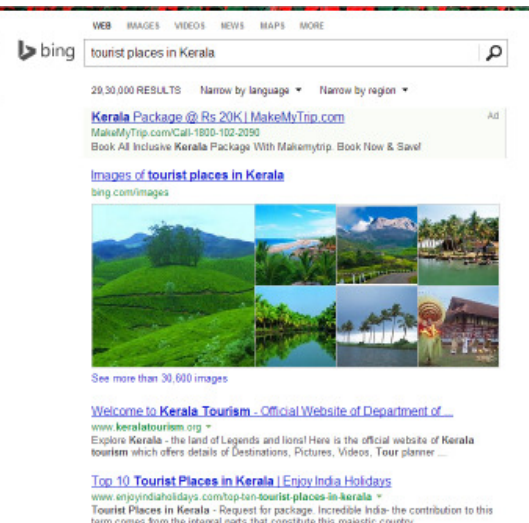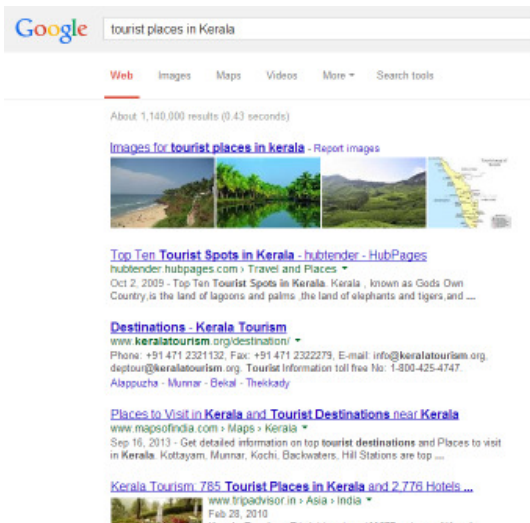

Fig. 9.8 : Working of a search engine

Fig. 9.9 : Search results of different search engines

select a list of URLs where the particular topic is found from the index and displays it as the result. Figure 9.8 shows the working of a search engine.

Some of the most popular web search engine sites are Google, Bing, Yahoo Search, Ask, etc. Figure 9.9 shows the search results of different search engines.

9.4.3 E-mail

E-mail enables us to contact any person in the world in a matter of seconds. Billions of e-mail messages are sent over the Internet every day. ***Electronic mail or e-mail*** is a method of exchanging digital messages between computers over Internet.

E-mail has become an extremely popular communication tool. The e-mail will be delivered almost instantly in the recipient's mail box (Inbox). Apart from text matter, we can send files, documents, pictures, etc. as attachment along with e-mail. The same e-mail can be sent to any number of people simultaneously. Figure 9.10 shows a sample e-mail message.
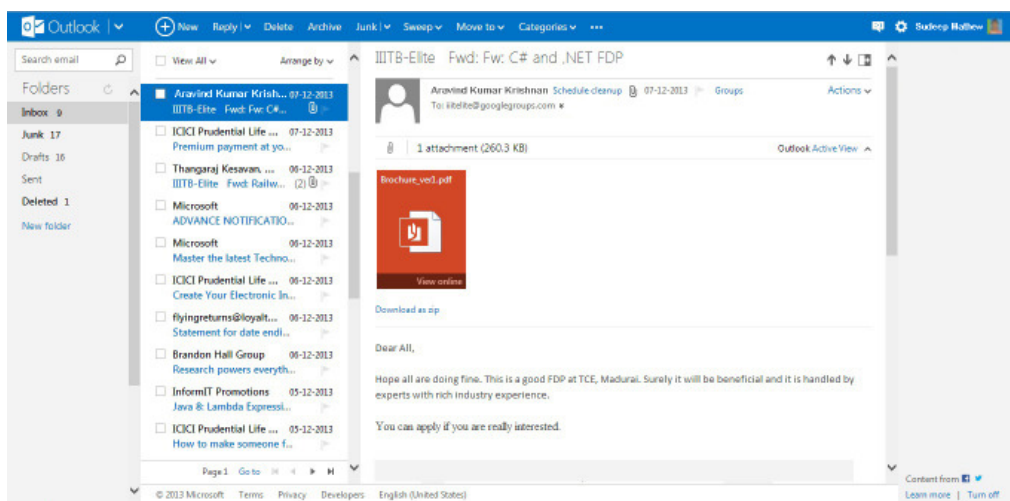

Fig. 9.10 : A sample e-mail message

Most of you will have an e-mail address. The structure of an e-mail address is: user name@domain name. An example of an e-mail address is

<p style="text-align:center">scertkerala@gmail.com</p>

An e-mail address consists of two parts separated by @ symbol. The first part *scertkerala* is the username that identifies the addressee and the second part *gmail.com* is the domain name of the e-mail server, i.e., the name of the e-mail service provider.

E-mails can be accessed using websites like *gmail.com*, *hotmail.com*, etc. that provide web applications consisting of functions to send, receive, forward, reply and organise emails . Such a facility is popular and is commonly referred to as web mail.

E-mails can also be accessed using e-mail client software that is installed in our computers. Such software uses our e-mail address and password to retrieve e-mails from the e-mail service provider's server and store it in our computer. An e-mail client allows to send, receive and organise e-mail. The messages sent when the computer is offline are stored in the program and send later when computer is online. For receiving messages, e-mail client applications usually use either the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP). The popular e-mail client applications are Microsoft Outlook and Mozilla Thunderbird.

a. Sections of an e-mail

A client software gives provisions to enter the following sections.

To (Recepient Address) – A box to provide the e-mail addresses of the primary recipients to whom the e-mail has to be sent.

Cc (Carbon copy) – Write the e-mail addresses of the secondary recipients to whom the message has to be sent.

Bcc (Blind carbon copy) – Write the e-mail addresses of the tertiary recipients who receive the message. When the message is received the primary and secondary recipients cannot see the email addresses of the tertiary recipients in the message. Depending on e-mail service used, the tertiary recipients may only see their own e-mail address in Bcc, or they may see the e-mail addresses of all recipients.

Subject – Provide a meaningful subject for your conversation here. This helps you to identify a conversation with a particular person when you search your e-mails later.

Content – Type your message here. Today most of the e-mail service providers offer features to create an attractive message by giving colours, changing font styles, size, etc.

Attachment facility allows us to send files like documents, pictures, etc. along with an e-mail. The 'Send' button is used to send the message to the recipients. 'Reply' button allows you to send a reply back to the sender of the message received. 'Forward' button helps you to send a message received by you to other people. Figure 9.11 shows the major sections of an e-mail.
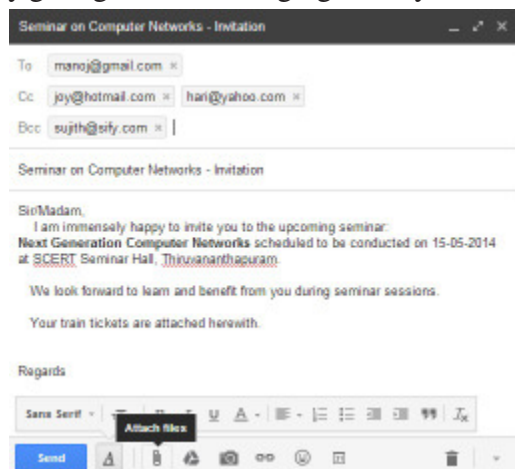

Fig. 9.11 : Composing an e-mail

b. Working of e-mail

Have you ever wondered how e-mail is sent from your computer to a friend on the other side of the world? When an e-mail is sent from your computer using web mail or e-mail client software, it reaches the e-mail server of our e-mail service provider. From there the message is routed from sender's e-mail server all the way to the recipient's e-mail server. The recipient's e-mail server then delivers the e-mail to the recipient's mail box (inbox), which stores the e-mail and waits for the user to read it. Simple Mail Transfer Protocol (SMTP) is used for e-mail transmission across Internet. Figure 9.12 shows the working of e-mail.
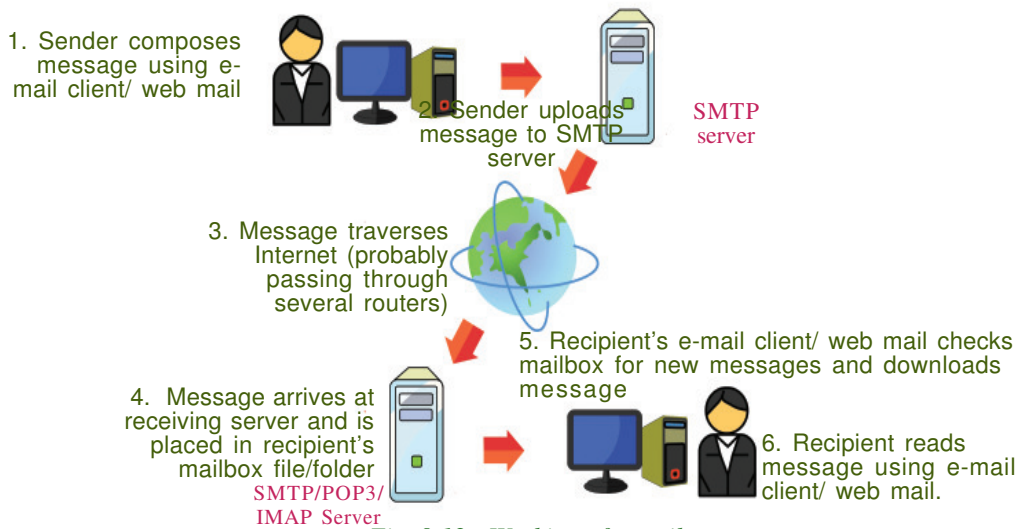
1. Sender composes message using e-mail client/ web mail

2. Sender uploads message to SMTP server

SMTP server

3. Message traverses Internet (probably passing through several routers)

4. Message arrives at receiving server and is placed in recipient's mailbox file/folder

SMTP/POP3/ IMAP Server

5. Recipient's e-mail client/ web mail checks mailbox for new messages and downloads message

6. Recipient reads message using e-mail client/ web mail.

Fig. 9.12 : Working of e-mail

c. Advantages of using e-mail

The benefits of using e-mail facility are listed below.

*   **Speed**: An e-mail is delivered instantly to any location across the globe. We can send the same e-mail to multiple users simultaneously.

*   **Easy to use**: We can send and receive e-mails, organise our daily conversations and save them easily on our computer.

*   **Provision of attachments**: The attachment feature allows to send pictures, files, documents, etc. along with e-mail.

*   **Environment friendly**: E-mails do not use paper and save a lot of trees from being cut down.

*   **Reply to an e-mail**: When we need to reply to an e-mail, we can use the provision of attaching previous e-mails as reference. It helps to refresh the recipient about the subject.

- **Cost-effective**: When compared to fax or conventional mail, e-mail is less expensive.
- **Available anywhere anytime**: Messages can be read at user's convenience. Access to mail box is available anytime.

The e-mail service, though beneficial in our daily life, can be misused in different ways as listed below.

- **E-mails may carry viruses**: Viruses send along with e-mail can harm our computer system. Viruses can also illegally access our e-mail address book and spread virus infected messages to all email addresses in it.
- **Junk mails**: Checking and deleting unwanted mails consume a lot of time.

Internet of Things (IoT)

Can you imagine a fridge which checks its egg tray and reminds you to buy eggs in your mobile phone or orders the nearby grocery store to supply eggs to your home; an air conditioner that can be switched on or off using your mobile phone; or a car that automatically reminds you about filling fuel as you approach a fuel pump? This is being made possible using Internet of Things (IoT). IoT is the concept of connecting all devices like mobile phones, fridges, cars, air conditioners, lamps, wearable devices, etc. to the Internet. Each device is provided with a unique IP address which identifies it and allows it to transfer data over Internet without human intervention. The huge increase in the number of IP addresses due to the implementation of IPv6 supports the introduction of this technology. The IoT can be used to monitor health of patients and inform the doctor about an urgency, applied to things which help us reduce wastage like power, water, etc. and improve the way we work and live.

9.4.4 Social media

All of us are familiar with wikipedia, the free encyclopedia in Internet. We have also heard about people responding to social issues through facebook, twitter, etc. Also we know that people use youtube to share videos and for promotion of products or business. All of these are part of social media which is changing the way we communicate, entertain and live. Social media refers to the use of mobile and web-based technologies through which individuals and communities can create, share, discuss and modify content.

In social media, interactions among people happen in virtual communities and networks over Internet. These digital technologies influence the formation and activities of civil communities to a great extent.

a. Classification of social media

The various types of social media that exist on the Internet are: Internet forums, social blogs, microblogs, wikis, social networks, content communities and a lot more. Figure 9.13 displays logos of popular social media websites. Here we discuss the most popular classifications of social media.

1. Internet forums

An Internet forum is an online discussion web site where people can engage in conversations in the form of posted messages. Each Internet forum will have sub forums which may have several topics. Each discussion on a topic is

Fig. 9.13 : Logo of popular social media websites

called a thread. People can login and start a thread or respond to discussion in a thread. Some forums allow anonymous login also. Discussions can be about programming, social/political issues, fashion, etc. These discussions help us to learn and find solutions to problems. Ubuntu Forum – a community that provides help on Ubuntu is a popular forum.

2. Social blogs

A blog (web log) is a discussion or informational website consisting of entries or posts displayed in the reverse chronological order i.e., the most recent post appears first. Some blogs provide comments on a particular subject; others function as personal online diaries and some others as online brand advertising for a particular individual or company. Initially blogs were created by a single user only. But now there are multi-author blogs that are professionally edited. Blogger.com and Wordpress.com are popular sites that offer blogging facility.

3. Microblogs

Microblogs allow users to exchange short sentences, individual images or video links. People use microblogs to share what they observe in their surroundings – information about events and opinions about topics from a wide range of fields. Microblogging offers a communication mode that is spontaneous and can influence public opinion. Twitter.com is a popular microblogging site.

4. Wikis

Wikis allow people to add content or edit existing information in a web page, to form a community document. Wiki is a type of content management system. Editing done by

users is very closely monitored by other editors and therefore incorrect information, advertising, etc. are removed immediately. wikipedia.org – the free online encyclopedia is the most popular wiki on web.

Wikipedia is a free online encyclopedia to which anyone can add content and edit. Wikipedia was formally launched on 15th January 2001 by Jimmy Wales and Larry Sanger using the concept and technology of a wiki. Wikipedia consists of over 3 crore articles in around 300 languages. The english edition alone includes around 44 lakh articles and is one of the most visited websites on Internet. Articles on topics range from very broad to highly specific. Each article consists of a number of links to Wikipedia itself and other external resources. Since users are able to create and edit articles, the quality of the content in the articles depends on the person who contributes and edits it. The Malayalam edition of Wikipedia is available at ml.wikipedia.org.

5. *Social networks*

Social networking sites allow people to build personal web pages and then connect with friends to communicate and share content. We can share text, pictures, videos, etc. and comment to the posts. A social networking site can be for general topics or for a specific area like professional networking. Public opinion is greatly influenced by the discussions and posts in these websites. Popular social networking sites are facebook.com and linkedin.com.

6. *Content communities*

Content communities are websites that organise and share contents like photos, videos, etc. Youtube.com is a popular video sharing site and flickr.com shares pictures.

Most of today's social media websites offer more than one type of service, i.e., social networking and microblogging; blogging and internet forum; etc. Studies have revealed that social media is now recognised as a social influencer.

b. Advantages of social media

- **Bring people together:** Social networking allows people to find long-lost childhood friends and make new ones.

- **Plan and organise events**: These sites help users to organise and participate in events.

- **Business promotion**: Social media offers opportunities for businesses to connect with customers, implement marketing campaigns, manage reputation, etc.

- **Social skills**: These sites allow people to express their views over a particular issue and become an agent for social change.

c. Limitations in use of social media

- **Intrusion to privacy**: The personal information of users can be used for illegal activities. Information like the e-mail address, name, location and age can be used to commit online crimes.

- **Addiction**: Addiction to these sites wastes our valuable time. It will negatively affect our mental states and may lead to depression and tension. It can reduce the productivity of workers in an organisation. Students may lose concentration and this in turn may affect their studies.

- **Spread rumours**: Social media will spread the news very quickly. It can facilitate or worsen a crisis by spreading negative information or misinformation at an incredible speed.

d. Social media interaction – Best practices

- Avoid unnecessary uploading of personal data like e-mail address, telephone number, address, pictures and videos.

- Setting time schedule for using these sites can save wastage of time.

- In social media websites like wikis and blogs, photo and video sharing are public. What you contribute is available for all to see. Be aware of what you post online. Avoid posting content you may regret later.

- Set your privacy levels in such a way that you know exactly who can see your posts and who can share them. The three basic privacy levels in social media are private, friends and public.

Let us do

- *Prepare a chart on the different social networking websites and their uses.*

- *Create a blog of your class and update the activities like achievements in sports, arts, class tests, assignments, etc.*

- *Conduct a survey in your school to find the most popular Internet browser. Also prepare a chart based on the collected data.*

Check yourself

1. Give an example for an e-mail address.
2. Which of the following is not a search engine?
   (a) Google   (b) Bing   (c) Facebook   (d) Ask
3. Name the protocol used for e-mail transmission across Internet.
4. What is a blog?
5. Name two services over Internet.
6. Each document on the web is referred using _____.

9.5 Cyber security

Today, we know that people use Internet to transfer personal and confidential information or make payments, organisations like banks perform all their financial transactions using their computer network, railways do business – selling tickets, information on running trains, etc. using the railway's computer network. Can you imagine the volume of financial loss and other issues that may occur if these computer networks are not available, even for a short time?

Security to computer networks is vital because important data can be lost and privacy can be violated. Further, work or business can be interrupted for several hours or even days if a network comes under attack. With the arrival of the Internet, security has become a major concern as people started using Internet as a tool for communication and doing business. Every organisation should monitor its network for possible intrusion and other attacks. Here we discuss the common threats that affect a computer network.

9.5.1 Computer virus

A computer virus is a program that attaches itself to another program or file enabling it to spread from one computer to another without our knowledge and interferes with the normal operation of a computer. A virus might corrupt or delete data on our computer, replicate itself and spread to other computers or even erase everything in the hard disk. Almost all viruses are attached to executable files. A virus may exist on a computer, but it cannot infect the computer unless this malicious program is run or opened. Viruses spread when the file they are attached to, is transferred from one computer to another using a portable storage media (USB drives, portable hard disks, etc.), file sharing, or through e-mail attachments. Viruses have become a huge problem on the Internet and have caused damage worth billions.

9.5.2 Worm

A computer worm is a stand alone malware (malicious software) program that replicates itself in order to spread to other computers. Worms spread from computer to computer on its own. Unlike a virus, it does not need to attach itself to a program to propagate. A worm takes advantage of the data transport features of the computer system to travel without help. Worms always slow data traffic on the network by consuming bandwidth, whereas viruses almost always corrupt or modify files on a computer. The most destructive effect that a worm can cause is through e-mails. A worm can send a copy of itself to every address in an e-mail address book. Then, the worm sends itself to everyone listed in each of the receiver's address book and so on.

I LOVE YOU worm

This worm affected computers in 2000 by overwriting most of the files. Users received this worm as an e-mail with a subject line "ILOVEYOU" and with a file attachment LOVE-LETTER-FOR-YOU.TXT.vbs. Those who clicked the attachment got their computers affected by the worm and lost their files.

9.5.3 Trojan horse

A Trojan horse, will appear to be a useful software but will actually do damage once installed or run on the computer. Users are typically tricked into loading and executing it on their systems. When a Trojan is activated on a computer, they can cause serious damage by deleting files and destroying information on the system. Some Trojans create a backdoor on the computer. This gives malicious users access to confidential or personal information in the computer through the network. Unlike viruses and worms, Trojans do not reproduce by infecting files nor do they self-replicate.

Ie0199.exe Trojan

E-mail users received a message that offered a free upgrade to Internet Explorer that contained an executable file Ie0199.exe as attachment. This e-mail instructed the user to download and install this program for the upgrade. The users who followed these instructions got their files infected.

Trojan War

In Greek mythology, the Trojan War was waged against the city of Troy by the Greeks after Prince Paris of Troy stole away the Greek Queen Helen. The Greeks fought a battle with the city of Troy for ten long years. The Greek soldiers got fed up and wanted to return to their homes. Then, Athena, the Goddess of war gave the Greeks an idea to end the war.

According to the plan, they built a big hollow wooden horse. The hollow horse was filled with soldiers and they left it as a gift for the Trojans. All other soldiers pretended to abandon their camp. Trojans thought that they had won the war. They pulled the huge horse to their city. They started celebrations of their victory. In the night when everyone was asleep, the Greek soldiers opened the horse and came out. They killed the sleeping soldiers of Troy and rescued Queen Helen.
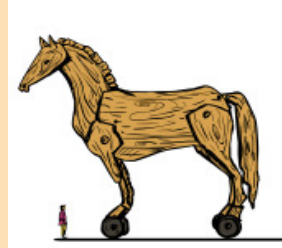
9.5.4 Spams

Spams or junk mails are unsolicited e-mails sent indiscriminately to persons to promote a product or service. Spammers collect e-mail addresses from chat rooms, websites, customer lists, newsgroups, etc. Clicking on links in spams may send users to websites that host certain viruses. Today most e-mail service providers provide e-mail filters that can successfully separate genuine email from spams as indicated in Figure 9.14.

Fig. 9.14 : Collection of spams in the e-mail menu

9.5.5 Hacking

In computer networking, hacking is a technical effort to manipulate the normal behavior of network connections and connected systems. Hacking is performed both by computer security experts and by computer criminals. Computer experts perform hacking to test the security and find the vulnerabilities in computer networks and computer systems. Such computer experts are often called 'white hats' and such hacking is called ethical hacking.

Computer criminals break into secure networks to destroy data or make the network unusable for those who are authorised to use the network. They do this with the intent of stealing confidential data or destroying files. Such criminals are called 'black hats'.

There is another category of hackers called grey hat hackers, who fall between white and black hackers. They sometimes act illegally, though with good intentions, to identify the vulnerabilities. Grey hat hackers do this to achieve better security.

9.5.6 Phishing

Phishing is a type of identity theft that occurs online. Phishing is an attempt to acquire information such as usernames, passwords and credit card details by posing as the original website, mostly that of banks and other financial institutions. Phishing websites have URLs and home pages similar to their original ones. The act of creating such a misleading website is called spoofing. People are persuaded to visit these spoofed websites through e-mails. Users are tempted to type their usernames, passwords, credit card numbers, etc. in these web pages and lose them to these websites. These frauds use this information to steal money. Phishing is currently the most widespread financial threat on the Internet. The URL in Figure 9.15 indicates that it is a phishing website.
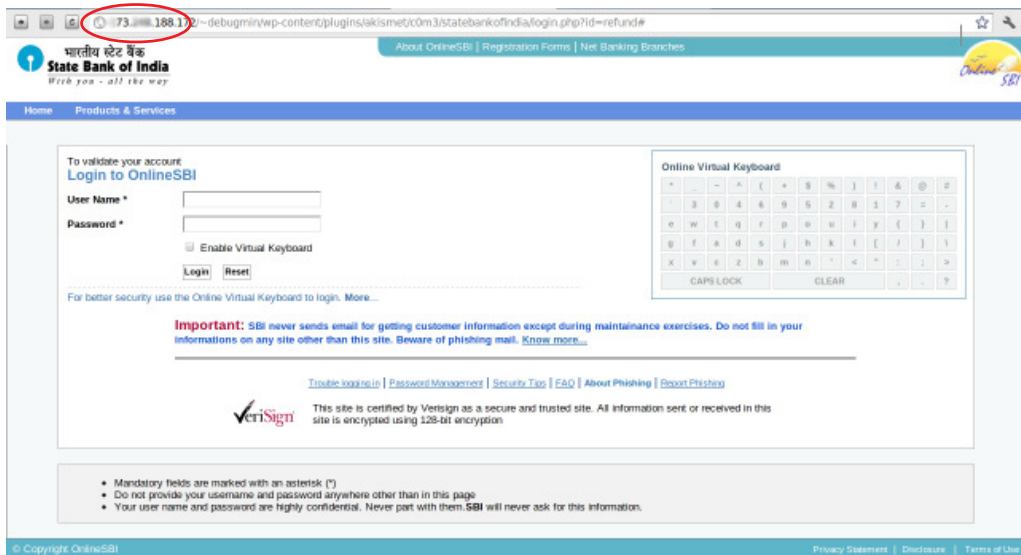

Fig. 9.15 : A phishing website

9.5.7 Denial of Service (DoS) attack

A Denial of Service(DoS) attack is usually aimed at a web server. Such an attack forces the server/computer to restart. An attack in which the attackers' goal is to shut down the target server rather than stealing data is called DoS attacks.This prevents genuine users of a service/website on the web server from using that service. This attack can be done using a single computer called Denial of Service (DoS) attack or using more than one computer called Distributed Denial of Service (DDoS) attack.

We have learned that when we type a website address in the browser and press the **Enter** key, the browser requests for that web page from the server. DoS attacks sends large number of such requests to the server until it collapses under the load and stops functioning. A DoS attack using a computer on a network slows down the network by flooding a server with a large number of requests. A DDoS attack uses multiple computers in the network that it has previously infected. These infected computers called 'zombies', work together and send out large quantities of fake messages/ requests to the target server. Figure 9.16 shows the Distributed Denial of Service attack. This increases the amount of data traffic to the

Fig. 9.16 : Distributed Denial of Service (DDoS) attacks

target server. This leads to server overload and the server is unable to provide services to its users. The target computer is thus forced to reset / restart leading to unavailability of its service for a period of time. A DoS attack interrupts network service for some period, but it does not cause severe damage to files as in the case of a virus attack.
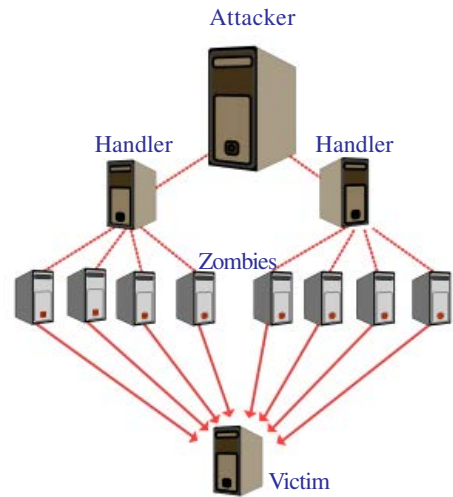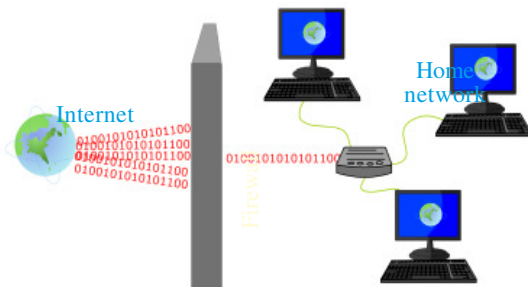
9.5.8 Man-in-the-Middle attacks

A man-in-the-middle attack refers to an attack in which an attacker secretly intercepts electronic messages between the sender and the receiver and then captures, inserts and modifies messages during message transmission. If sender transmits messages without appropriate security, the attacker may exploit the vulnerabilities in the network to capture and modify the messages and send the modified messages to the receiver. Since the network transmission still works properly, both the sender and receiver will find it difficult to notice that the messages have been trapped or modified by an intruder. If we use such a computer for online transactions, the man in the middle may capture our bank account number and password to steal money, leading to financial loss. Encrypted connections such as HTTPS (HTTP Secure), SFTP (Secure FTP), etc. should be used for secure transactions, so that intruders cannot modify the messages.

9.6 Preventing network attacks

Threats to computers and networks are a major issue as long as information is accessible and transferred across the Internet. Different defense and detection mechanisms are developed to deal with these attacks.

9.6.1 Firewall

A firewall is a system of computer hardware and software that provides security to the computer network in an organisation. A firewall controls the incoming and outgoing network traffic by analysing the data and determining whether they should be allowed through or not, based on a rule set. Firewalls deny malicious data from entering into the computer networks as shown in Figure 9.17.

Fig. 9.17 : Firewall

Sandboxing

Sandboxing is a technique through which programs that are suspected to be infected with a virus can be run. Through sandboxing such programs are run in a separate memory area and therefore cannot damage our operating system.

9.6.2 Antivirus scanners

Viruses, worms and Trojan horses are all examples of malicious software (malware). Antivirus tools are used to detect them and cure the infected system. Antivirus software scans files in the computer system for known viruses and removes them if found. The antivirus software uses virus definition files containing signatures (details) of viruses and other malware that are known. When an antivirus program scans a file and notices that the file matches a known piece of malware, the antivirus program stops the file from running, and puts it into 'quarantine'. Quarantine is a special area for storing files probably infected with viruses. These files can later be deleted or the virus can be removed. For effective use of antivirus software, virus definitions must be updated regularly.

9.6.3 Cookies

Cookies are small text files that are created when we use a browser to visit a website. Cookies keep track of our movements within the web site – remembers our user name, preferences, e-mail address, etc. Browsers store cookies for an interval of time, usually in a cookie folder on the client's computer. Cookies are text files and so they are not executable programs. Web sites use cookies mainly because they save time and make browsing efficient.

Cookies are treated as data and so it is not a virus, but it is always possible for a hacker to use it for malicious purposes. Cookies can be used to act as a spyware. There are

harmful cookies that are used by different websites to compromise our privacy. Such websites store a special cookie in our computer that will keep track of our activities like, websites visited, products purchased or the forms that are filled. Most browsers provide facilities to manage/delete harmful cookies. Frequent deletion of cookies helps to prevent illegal access and use of personal information.

I
STOP
BEFORE
I
DOWNLOAD

I
think
BEFORE I
click

MY PASSWORD MIXES
LeTters
#umbers
AND
$ymbol!#
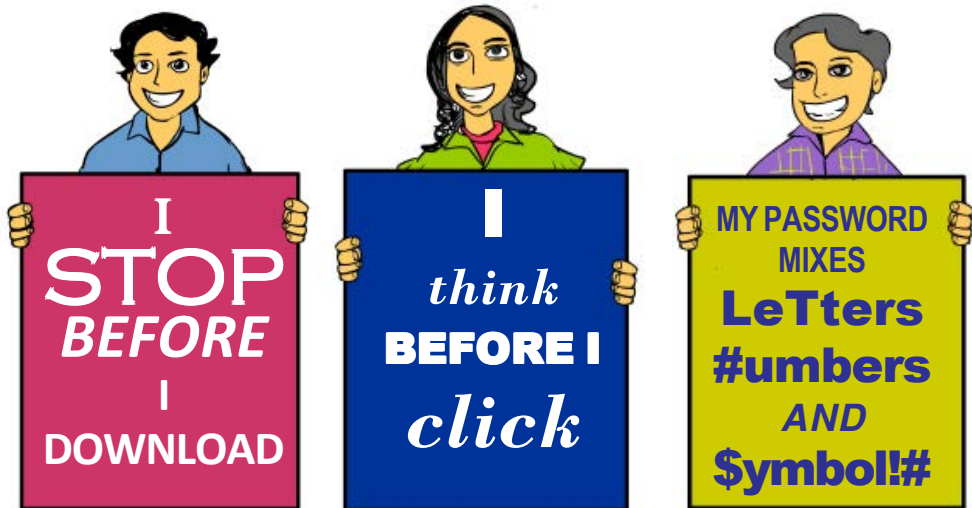
9.7 Guidelines for using computers over Internet

Following are the major guidelines for using computers over Internet.

• Most of the computer viruses are spread through e-mail attachments. Do not open any e-mail attachment that you are not sure about the sender.

• Download files only from reputed sources. Do not use/copy software that you cannot confirm the origin.

• Avoid clicking on pop-up advertisements. Close them instead.

• Use USB drives with caution. Plugging someone else's USB storage into your computer or plugging your own USB storage into a computer at an Internet cafe/ unsafe computer, can spread an infection through the USB storage.

• Make sure the firewall is set and turned on.

• Use strong passwords. Change passwords at regular intervals.

• Update the virus definitions of your antivirus program periodically online.

• Keep a regular backup of your important files (on DVD, another hard disk, etc.)

Guidelines for Setting up a Strong Password
- A password should have atleast 8 characters.
- A password should contain
    - Upper case letters
    - Lower case letters
    - Numbers
    - Symbols like @, #, $, etc.
- A password should not be personal information like name, date of birth, etc. or common words.
- Never disclose your password to others.
- Do not write it on a paper or store it in a file in your computer.
- Do not use the same password for all logins.
- Change password often.

- Be careful about giving personal data online. If you see e-mail message requests for personal data such as telephone number, address, credit card number, etc. from unknown persons, ignore it.

- Visit banks' websites by typing the URL into the address bar. Do not click on links within e-mails to go to bank websites. Banks or any of its representatives never sends you e-mail/SMS or phone calls to get your personal information, usernames or password. Never reveal your passwords or ATM card details to anyone.

- Check whether the website you are visiting is secure while performing financial transactions. The web address in the address bar should start with 'https://'. Also look for a lock icon on the browser's address bar.

- Keep a regular check on your online accounts. Regularly login to your online accounts, and check your statements. If you see any suspicious transaction, report them to your bank or credit card provider.
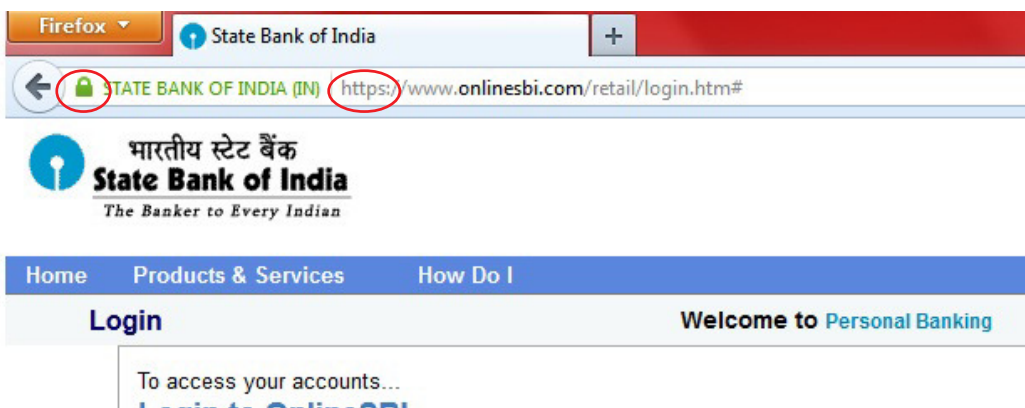

Fig 9.18 : Secure banking - lock icon and https

- *Conduct a group discussion on the topic "security threats / cyber attacks to your system", and draw a bar diagram based on information arrived in the event.*

- *List the names of various viruses and their features in the form of a chart.*

Check yourself

1. What is a virus?
2. What do you mean by phishing?
3. The small text files used by browsers to remember our email id's, user names, etc are known as _____ .
4. The act of breaking into secure networks to destroy data is called _____ hacking.
5. What is quarantine?

Let us sum up

The Internet, which was started as a defence project of the US government has become a part of our life. Today the Internet is accessed using mobile devices like mobile phones, tablets, etc. than using a desktop computer. Therefore speed of Internet access has become an important factor. New technologies connect to Internet focus on data transmission speed. Internet services like e-mail, social media, searching etc. have changed the way we communicate. Each of the above services has its own benefits and risks. Computer networks today play an important role in providing the above services. It has increased the risk factors for networks, like viruses, worms, Trojan horse, phishing, etc. Antivirus software, firewalls, etc. are used to protect computer networks from different kinds of attacks. The risks for a network attack can be reduced by following certain guidelines while using computers on Internet.

Learning outcomes

After the completion of this chapter the learner will be able to

- recognise the people behind the evolution of Internet.
- identify the hardware and software requirements for Internet connection.
- use the services available on Internet.
- classify the different types of social media.
- judge the risks while interacting with social media.
- recognise the threats to network security.

Sample questions

Very short answer type

1. Why is the invention of HTTP and HTML considered as an important land mark in the expansion of Internet?

2. Compare intranet and extranet.

3. Write short notes on

   a. Mobile broadband   b. Wi-MAX

4. Explain the terms web browser and web browsing.

5. Compare blogs and microblogs.

6. What are wikis?

7. What is firewall?

Short answer type

1. Your neighbour Ravi purchased a new PC for his personal use. Mention the components required to connect this PC to Internet.

2. What are the advantages of using broadband connection over a dial-up connection?

3. XYZ engineering college has advertised that its campus is Wi-Fi enabled. What is Wi-Fi? How is the Wi-Fi facility implemented in the campus?

4. Madhu needs to prepare a presentation. For this, he uses www.google.com to search for information. How does google display information when he types 'Phishing' in the search box and clicks search button?

5. Manoj's e-mail id is manoj@gmail.com. He sends an e-mail to Joseph whose e-mail id is joseph@yahoo.com. How is the mail sent from Manoj's computer to Joseph's computer?

6. How does a Trojan horse affect a computer?

7. Explain a few threats that affect a computer network.

Long answer type

1. Suppose you wish to visit the website of kerala school kalolsavam, www.schoolkalolsavam.in and you have entered the URL in the address bar. Write the steps that follow until the home page is displayed.

2. Write the disadvantages of social media. What are the different ways to avoid the disadvantages of social media?

3. Explain the various broadband technologies available for Internet access.

IT Applications

Information Technology (IT) is growing rapidly. It has influenced almost every sphere of life and has become an integral part of our daily life. Various IT applications available today help to make life easier. For example, we can pay various utility bills, reserve railway or airline tickets and do online shopping from anywhere, anytime. IT is also extensively used in the field of education. One of its recent contributions is e-Learning with virtual classrooms. The possibilities of IT are increasingly used in e-Governance, e-Business and in health care applications as well.

This chapter discusses some of these areas where IT is widely used today. In the case of online shopping, the required items can be ordered and the same will be delivered at our door step. Cash transactions can also be made online. The field of education has also advanced with the development of IT. Teachers and students need not be in the same room as in the conventional teaching system. They can share virtual class rooms and make use of e-Learning materials, sitting in dispersed locations. Healthcare is another major area which uses IT for its assistance. With the use of new computer aided medical imaging devices and related technologies, various diseases can be diagnosed at a very early stage. IT also plays a

key role in conducting research for the development of new drugs. We also discuss various ICT enabled services in this chapter.

Information and Communication Technology (ICT) is the term often used as an extended synonym for Information Technology (IT). ICT is more specific in integrating telecommunication and computers comprising of many technologies for capturing, storing, processing, interpreting and printing information.

10.1 e-Governance

Information Technology (IT) has contributed much to the smooth functioning of governments. It comprises many technologies for capturing, processing, interpreting, storing and transmitting information. IT helps governments to take quick and judicious decisions. It also increases transparency and accountability in all its services. Information and Communication Technology(ICT) aided Governance is generally known as e-Governance.

e-Governance is the application of ICT for delivering Government services to citizens in a convenient, efficient and transparent manner. The objective of e-Governance is to ensure that the services of the Government reach the public promptly and effectively. In Kerala, we have many government departments like the Motor Vehicle Department, Education, Revenue, etc. which have successfully implemented e-Governance. Let us now discuss various types of interactions in e-Governance.

10.1.1 Types of interactions in e-Governance

e-Governance facilitates interaction among different stakeholders in governance. Categories of these interactions are described as follows:

Government to Government (G2G) - It is the electronic sharing of data and/or information among government agencies, departments or organisations. The goal of G2G is to support e-Governance initiatives by improving communication, data access and data sharing.

Government to Citizens (G2C) - It creates an interface between the government and citizens. Here the citizens enjoy a large range of public services. It increases the availability and accessibility of public services. It also improves the quality of services. Its primary purpose is to make the government citizen-friendly.

Government to Business (G2B) - Here, e-Governance tools are used to aid the business community to interact with the government. The objective is to cut red-tapism, save time, reduce operational costs and create a more transparent business environment while dealing with the government.

Government to Employees (G2E) - Government, being the biggest employer has to interact with its employees on a regular basis. This interaction is a two-way process between the government and the employees. The policies and guidelines for implementing

various government programmes are made available to the employees as government orders or circulars through e-Governance portals. The salary and personal details of government employees are also managed through e-Governance services. Use of ICT tools helps in making these interactions fast and efficient.

10.1.2 e-Governance infrastructure

In India, the e-Governance infrastructure mainly consists of State Data Centers (SDC) for providing core infrastructure and storage, State Wide Area Network (SWAN) for connectivity and Common Service Centers (CSC) as service delivery points. The integration of these three components of e-Governance is shown in Figure 10.1.
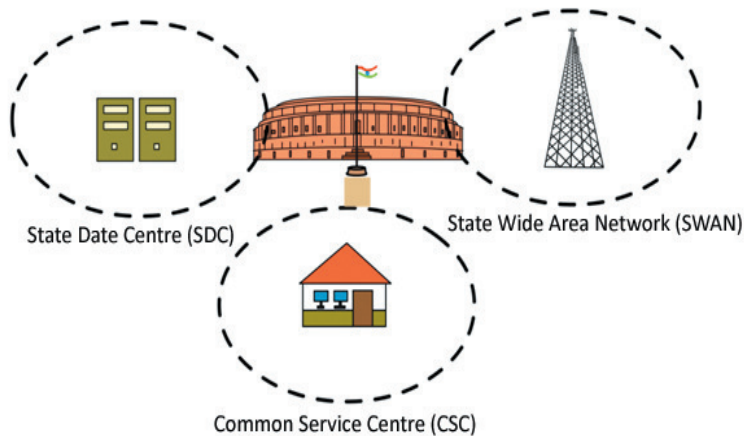

Fig 10.1 : e-Governance infrastructure

a. State Data Centre (SDC)

State Data Centre (SDC) is one of the important constituents of the core infrastructure for supporting e-Governance initiatives of National e-Governance Plan (NeGP). Under NeGP, it is proposed to create SDCs to combine services, applications and infrastructure and to provide efficient electronic delivery of G2G, G2C and G2B services. These services can be rendered by each state government through a common delivery platform. This platform is supported by a core connectivity infrastructure such as State Wide Area Network (SWAN) and Common Service Centre (CSC) connectivity that is extended up to the village level. State Data Centre provides several functionalities. These include keeping central data repository of the state, securing data storage, online delivery of services, citizen information/services portal, state intranet portal, disaster recovery, etc. SDCs also provide better operation and management control and minimize the overall cost of data management, resource management, deployment etc.

b. Kerala State Wide Area Network (KSWAN)

Kerala State Wide Area Network (KSWAN) has been set up as a backbone of the State Information Infrastructure (SII). It connects Thiruvananthapuram, Kochi and Kozhikode as its hubs and extends to all the 14 districts linking each of the 152 Block

Panchayats. The network also connects a large number of various offices of government departments. The infrastructure supports the integration of a large number of G2G, G2C services with the applications received from the State Data Centre.

c. Common Service Centre (CSC)

Common Service Centres (CSC) are the front-end delivery points of the government, private and social sector services for the rural citizens of India. A highlight of the CSCs is that it offers web-enabled e-Governance services in rural areas. It helps in utility payments such as electricity, telephone and water bills, submission of online applications and generating and distributing the certificates to the needy. Other services that could be offered through CSC are listed below:

- Agriculture services
- Education and training services
- Health services
- Rural banking and insurance services
- Entertainment services
- Commercial services

In Kerala *Akshaya* centres are working as Common Service Centres.

Akshaya centres

Akshaya centres were initially launched in the year 2002 in the Malappuram district in Kerala. Its intention was to impart e-Literacy to the citizens. Akshaya was conceived as a landmark ICT project by the Kerala State Information Technology Mission (KSITM) to bridge the digital divide and to bring the benefits of ICT to the entire population of the State. The services include e-grantz, e-filing, e-district, e-ticketing, submitting online application for ration card and electoral ID, Aadhaar enrolment, Aadhaar based services, insurance and banking services.

Akshaya was launched on 18th November 2002 by the former President of India, Dr. A. P. J. Abdul Kalam.

Akshaya works in the PPP (Private Public Partnership) Model. An Akshaya centre is owned by a private entrepreneur preferrably from the same panchayath where the Akshaya centre is located and its selection is done by local self governments. The income for the Akshaya Entrepreneurs is the nominal service charge collected either from the citizen or paid by the government for each transaction.

അക്ഷയ

അവസരങ്ങളുടെ ജാലകം

www.akshaya.kerala.gov.in

Aadhaar is a 12 digit individual identification number issued as part of e-Governance by the Unique Identification Authority of India (UIDAI) on behalf of the Government of India. This number will serve as a proof of identity and address, anywhere in India. Any individual, irrespective of age and gender, who is a resident in India, can enroll in Aadhaar. Each individual needs to enrol only once which is free of cost. During the enrolment, demographic details (name, age , gender and address) and biometric details (photo, 10-finger- print and 2-iris-images) of a resident are entered. Each Aadhaar number will be unique to an individual and will remain valid throughout life. Aadhaar number will help you provide access to services like banking, taking mobile phone connections and other Government and Non-Government services.

10.1.3 Benefits of e-Governance

Implementing e-Governance has many advantages. It informs the public about the functioning of the government and the policies they are trying to implement. Major benefits of e-Governance are listed below:

*   e-Governance *leads to automation of governement services*, ensuring the information regarding the activities of public welfare is easily available to all citizens.

*   e-Governance *strengthens the democracy* by ensuring greater participation of citizens at all levels of governance.

*   It ensures *more transparency in the functioning* and thus helps eliminate corruption.

*   It *makes every government department responsible* as they know that they are closely observed.

*   Proper implementation of e-Governance *saves unnecessary visits of the public to offices*. It saves time and money.

10.1.4 Challenges to e-Governance

Though e-Governance has many benefits, it faces some challenges too. Let us have a look at a few challenges in the implementation of e-Governance.

*   The existing digital divide is an obstacle in utilising the services of e-Governance. Those who live in remote areas with lower e-Literacy will face difficulty to access the services of e-Governance.

*   e-Governance applications being computer/web based, security measures are highly required since there is possibility of cyber-attack.

*   Usually a huge initial investment and planning are required for the proper implementation and maintenance of e-Governance services.

*   Many people are anxious about the sharing of their personal information, biometry, etc. to the agencies designated for data collection.

*   Integrity of various departments is very much essential for the efficiency and effectiveness of e-Governance.

10.1.5 Useful e-Governance websites

Table 10.1 shows the addresses of some websites that provide various services of the Government of Kerala. Figure 10.2 shows the home page of the Department of Higher Secondary Education, Kerala State and Figure 10.3 shows the home page of e-District portal.

Address of website	Purpose/Service
www.dhsekerala.gov.in	An official site of the Department of Higher Secondary Education, Government of Kerala that provides various facilities and services to students, teachers and school administrators.
www.edistrict.kerala.gov.in	It provides government services to the public.
www.incometaxindia.gov.in	It provides the services of Income Tax department.
www.keralamvd.gov.in	It is the official website of the Motor Vehicles Department of Kerala .
www.rti.gov.in	Right to Information Act 2005 mandates timely response to a citizen who requests for information from the Government or authorities under it.
www.itmission.kerala.gov.in	It is a Government of Kerala site which provides managerial support to various IT initiatives.
www.spark.gov.in	It is a web based G2E integrated solution for Service and Payroll Management.

Table 10.1 : Some e-Governance websites

Fig. 10.2 : The home page of DHSE website

www.dhsekerala.gov.in

As an official site of the Department of Higher Secondary Education, Government of Kerala, www.dhsekerala.gov.in provides various facilities and services to students, teachers and school administrators. The services include registration for the examinations, issue of hall tickets, publication of exam results, distribution of various circulars and notifications, etc. This site also provides facilities for downloading various forms and study materials including model question papers.
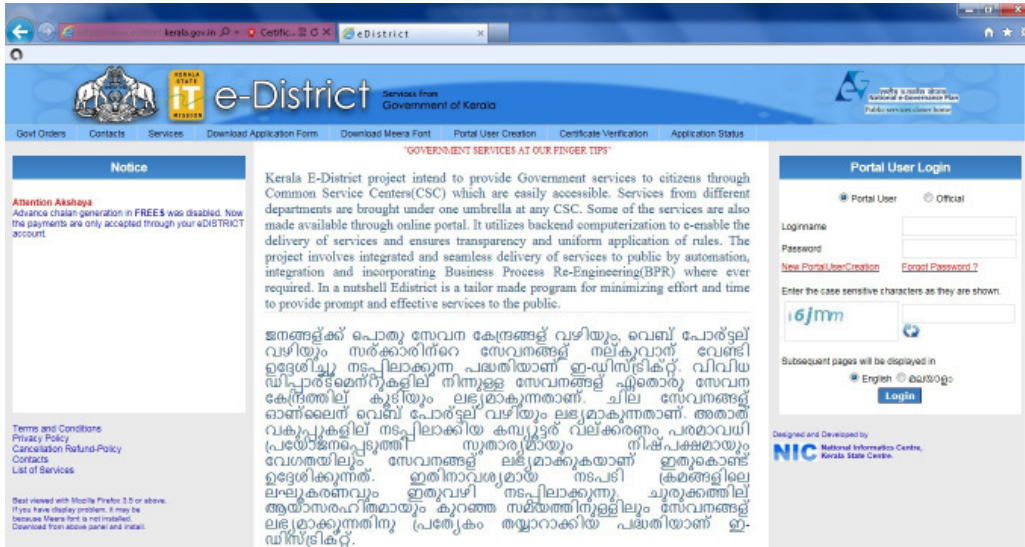

Fig. 10.3 : The home page of e-District portal.

e-District is a State Mission Mode Project under the National e-Governance plan. The project aims at supporting computerisation for services to the public.

In Kerala, issue of different certificates of the Department of Revenue is integrated with the e-District portal. It has simplified all complex procedures in government offices and made the citizens' life easy. It has been a difficult task for the citizen to obtain copies of public records, submit applications, meet officials, or to seek information regarding their day-to-day needs. This has caused the loss of his/her day's income, time as well as the cost of transportation. At government offices, sometimes, the relevant record, information, or official may not be available, resulting in repeated visits and additional expenses.

To avail a service in e-District, one has to apply online through the web portal or approach a nearby Akshaya centre with essential documents. Akshaya centre submits an online application on behalf of the citizen. This application through a work-flow reaches the village officer. After processing the application, the village officer forwards it to the next higher authority or approves it and issues the certificate using the digital signature. The availability of the certificate in the portal is informed with an SMS alert.

Check yourself

1. Name the application of Information and Communication Technology (ICT) for delivering government services to the citizens in a convenient, efficient and transparent manner.
2. Define the term e-Governance.
3. "e-Governance facilitates interaction between different stakeholders in governance". Say whether the statement is True or False.
4. Give an example for an e-Governance website.
5. What is KSWAN?

10.2 e-Business

We often visit shopping places and it takes a considerable amount of time and effort for shopping. What about the idea of shopping without leaving your home? It is one of the realities of the times. Almost all services and business are available online now. They are only a few clicks away.

e-Business is the sharing of business information, maintaining business relationships and conducting business transactions by means of the ICT application. Companies (business), public institutions (administration), as well as individuals (consumer) engage in e-Business. e-Business helps in effective e-marketing and increasing sales through effective use of e-Commerce services. It also reduces communication and travel cost through online meeting and shared workspace, etc.

10.2.1 e-Commerce and e-Business

Although the terms e-Commerce and e-Business are often used interchangeably there are differences. e-Commerce covers business transaction that involve exchange of money, whereas e-Business includes all aspects of running a business such as marketing, obtaining raw materials or goods, customer education, looking for suppliers etc. Thus e-Business is an extension of e-Commerce.

10.2.2 Electronic Payment System (EPS)

Can you think of living without money in your pocket? If somebody tells that he/she lived a week without touching currency notes, will you believe? Today we live in a world where almost every commodity has a price tag attached to it. Then how is it possible to buy something without giving money, the paper currency?

Electronic Payment System (EPS) is the solution. When we move on to electronic business, exchange of money also needs to be electronic. It is both convenient and secure if properly implemented. EPS plays an important role in e-Business.

A system of financial exchange between buyers and sellers in an online environment is called an **Electronic Payment System** (EPS). The financial exchange is facilitated by a digital financial instrument (such as credit/debit card, electronic cheque or digital cash) backed by a bank and/or an intermediary.

10.2.3 e-Banking

Banking has also undergone drastic changes with the advancements in Information Technology. Our traditional concept of banking has changed a lot. For example, now there is no need to visit bank for making financial transactions. We can do the transaction using online banking facilities even while travelling. Facilities such as ATM, debit cards, credit cards, Internet banking and core banking help in transforming traditional banking into e-Banking. **e-Banking** or electronic banking is defined as the automated delivery of banking services directly to customers through electronic channel. It can access data without geographical limitations.

10.2.4 Advantages of e-Business

e-Business provides many advantages to customers and firms. Let us discuss some of the major advantages of using e-Business applications.

- *It overcomes geographical limitations*. If you have a physical store, you are limited by the geographical area where you can provide service. But with e-Commerce, this limitation can be overcome.

- *e-Business reduces the operational cost*. An e-Commerce merchant does not need a prominent physical location; it reduces the operational cost. A portion of money thus saved can be passed on to the customers in the form of discounts.

- *It minimises travel time and cost*. Sometimes customers have to travel long distances to reach their preferred store. e-Business allows them to visit the same store virtually.

- *It remains open all the time*. e-Business application services are always open (24×7). From the merchant's point of view, it increases the number of service requests they receive. From the customer's point of view an 'always open' store is more convenient.

- *We can locate the product quicker from a wider range of choices*. On an e-Business website the consumers can have a wider range of choices of a product from various sellers. Customers can quickly locate their preferences from the given product lists. Some websites remember customer preferences and shopping lists to facilitate repeat purchase. The features like product characteristics and price comparisons are the other attractions of e-Business applications.

The term **M-Business** or mobile business covers all the business activities, processes and applications performed by means of mobile devices such as mobile phone, PDA, etc. M- Business can be regarded as a branch of e- Business.

e-tailing (or electronic retailing) is the selling of retail goods on the Internet. It is the most common form of business-to-consumer (B2C) transaction.

10.2.5 Challenges to e-Business

Though e-Business has enormous potential in the business world, it faces many challenges, particularly in developing countries like India. Major challenges are discussed below:

* A good percentage of the population is unaware of IT applications and its uses. Surprisingly, most of the regular Internet users also lack knowledge about online business and its possibilities.

* Most of the customers, especially to rural population, do not possess plastic money- credit card, debit card and net banking system, which is necessary for e-Business.

* If not used with caution, customers may lose valuable information like their credit card number, passwords, etc.

* Products like apparel, handicrafts, jewellery, etc are often purchased after examining physically. But in online shopping, customers don't have this '*touch and feel*' advantage.

* For the success of e-Business of any organisation or company, perfect and efficient shipment service is a necessity.

10.2.6 Useful e-Business websites

Some popular e-Business websites are given in Table 10.2. Figure 10.4 and 10.5 are the home pages of some e-Business websites

Address of website	Purpose/Service
www.irctc.co.in	Indian Railway Catering and Tourism Corporation Limited web site for reservation and cancellation of railway tickets and hotels for accommodation.
www.amazon.com	US based retailer with headquarters in Washington.
www.ebay.in	It is one of the largest online shopping sites in India.
www.licindia.com	Insurance company website.
www.airindia.com	Online site for booking Air India flight tickets.
www.keralartc.com	Online website for KSRTC bus ticket booking.
www.bookmyshow.com	Movies and theatre ticket booking website.

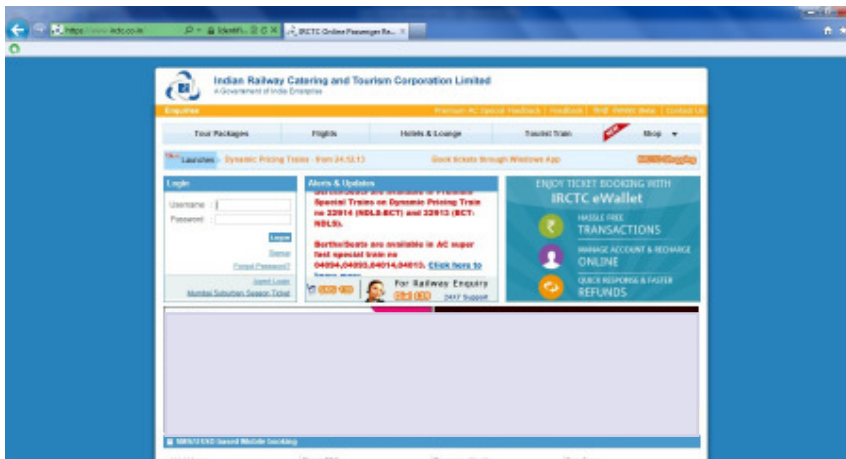*Table 10.2 : Some e-Business websites*

Fig. 10.4 : The home page of www.irctc.co.in

www.irctc.co.in

Indian Railway Catering and Tourism Corporation Limited (IRCTC) is a government of India enterprise. It handles the catering , tourism and online ticketing operations of the railways. IRCTC provides online booking facilities of railway tickets and offers other services like checking reservation status, train timing, reservation of hotels for the tourists across the country.
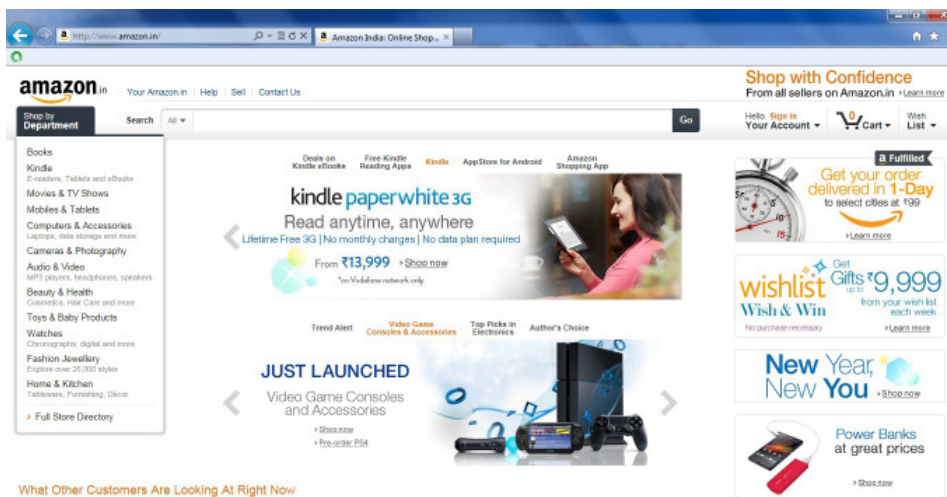

Fig 10.5 : The home page of amazon.in

www.amazon.com

amazon.com, Inc is a US based company with headquarters in Washington. It is the world's largest retailer. Initially started as an online bookstore it soon diversified itself into selling software, video games, electronics, furniture, food, toys, jewellery, etc. The company also produces consumer electronics. Amazon has separate retail websites for different countries. In 2013, Amazon launched its site for India, www.amazon.in.

1. The system of financial exchange between buyers and sellers in an online environment is known as _____.
2. Define e-Business.
3. Define e-Banking.
4. Check whether the following statement is true or false.
   " e-Business is an extension of e-Commerce".

10.3 e-Learning

We are familiar with the traditional classroom consisting of a teacher, students, teaching aids, etc. Think of a virtual class where you are attending classes as a student and a teacher is engaging class from a distant location, may be from a different country. Is this possible? Yes it is! Through e-Learning you can overcome many limitations of conventional teaching-learning process. The use of electronic media and ICT (Information and Communication Technologies) in education is termed **e-Learning**.

10.3.1 e-Learning tools

There are many tools for enhancing e-learning process. Some of the e-Learning tools are discussed below:

a. Electronic books reader (e-Books)

Think of going to school without bags packed with text books, but all necessary books are still with you in a small hand-held storage device! It is interesting, isn't it? e-Book reader is a device that can store lot of books in digital form. Portable computer devices that are loaded with digital book content via communication interfaces is called **electronic books reader**. It can open any book, any page quickly, without much effort. In principle, any web document can be downloaded from the Internet, sometimes by making online payment, and read whenever required. In some e-Book readers voice output is also available and the reader needs to just listen the audio of the text.

b. e-Text

Textual information available in electronic format is called **e-Text**. This text can be read and interacted with an electronic device like computer, e-Book reader, etc. e-Text can be converted to various formats to our liking using softwares. e-Text can be automatically read aloud with the help of a computer or an e-Text reader device. This is quite helpful for visually challenged people.

c. Online chat

It is a real-time exchange of text messages between two or more persons over the Internet. In the virtual class environment, online chatting is used to discuss the topics with teachers and other students. Chatting can be performed even with a low speed

Internet connection. Video chatting facility is also available. It however requires fairly high speed Internet connection and supporting devices such as web camera and speakers.

On line chat is a framework that provides interaction in a social environment. It helps communicating with people at different places. This facility can also be used to deliver and transfer live lectures given by the faculty. Students can login from the remote locations and attend the lecture.

d. e-Content

Nowadays lot of e-Learning materials are delivered in different multimedia formats like videos, presentations, graphics, animations, etc. The e-Contents once prepared can be broadcasted through television channels, webcasted or uploaded in websites. Uploaded content can be downloaded, viewed and saved for future reference. The multimedia contents have more scope to convey the subject matter effectively. It can show actual or simulated activities which are not possible in the traditional classroom based teaching.

e. Educational TV channels

There are many telecasting/webcasting channels which are dedicated for the e-Learning purpose. These channels broadcast recorded classes on various subjects, interviews with experts, lab experiments, etc. Some of these channels can be watched in the Internet also. Dooradarshan's 'VYAS' and Kerala Government's '**VICTERS**' channel are examples of educational television channels.

10.3.2 Advantages of e-Learning

e-Learning has lot of advantages. They are listed below:

- e-Learning has the ability to offer courses on variety of subjects to large number of students from distant location.
- In e-Learning cost for learning is much less. It saves journey time and money, instructor fees, etc.
- People with limited financial resources are very much supported by the lower cost of e-Learning.
- It provides facility to do online courses from various nationally or internationally reputed institutions.
- Time and place is not a constraint for e-Learning.

10.3.3 Challenges to e-Learning

Listed below are some of the challenges faced by e-Learning:

- Face to face contact between students and teachers is not possible.
- Proper interaction among teachers and students are often limited due to the lack of infrastructural facilities.
- Equipment and technology (computer and high speed Internet) requirement restrict adoption of e-Learning.
- Learners who require constant motivation may not be serviced adequately.
- Hands-on practicals in real laboratory scenario is also a constraint in e-Learning.

10.3.4 Useful e-Learning websites

The summary of useful e-Learning websites are shown in Table 10.3. Figure 10.6 shows the home page of IGNOU (Indira Gandhi National Open University)

Address of website	Purpose/Service
www.ignouonline.ac.in	Website of Indira Gandhi National Open University, one of the leading open universities offering various courses in the distance education mode.
www.nptel.iitm.ac.in	National Programme on Technology Enhanced Learning (NPTEL) provides e-Content through online Web and Video for courses in Engineering, Science and humanities streams.
www.ncert.nic.in	Website of National Council of Educational Research and Training. Includes many features such as providing study materials and reference materials in Hindi and English.
www.spoken_tutorial.org	It is an IIT Mumbai initiative. It provides software training through the spoken tutorials. All the courses are offerred totally free of cost as it is funded by the Government of India.
www.w3schools.com	W3Schools is a web developer information website, with tutorials and references relating to web development topics such as HTML, CSS, JavaScript, PHP, SQL, and JQuery.
www.gcflearnfree.org	GCFLearnFree.org is supported by the Goodwill Community Foundation. It provides easy-to-follow lessons in mathematics, science and technology and English to help anyone anywhere learn skills and gain knowledge.

Table 10.3 : Some e-Learning websites

Fig. 10.6 : The home page of IGNOU site

www.ignouonline.ac.in

IGNOU (Indira Gandhi National Open University) is an open university under Government of India. The website provides several features including eGyankosh, which is a digital repository available in text and video format. The site also provides webcasting facility linking to education channels - Gyandarshan, Gyanvani and EDUSAT. Virtual class provides links to all the online programmes conducted by the University.

Fig .10.7 : The home page of NPTEL website

www.nptel.iitm.ac.in

NPTEL is an abbreviation for National Programme on Technology Enhanced Learning which is an initiative by seven Indian Institutes of Technology (IIT) and Indian Institute of Science (IISc) for creating course content in engineering and science. Web based Course materials have been developed for different courses and are accessible freely through the website. Lecture contents are also made available in web site through video.

Check yourself

1. Real-time exchange of text messages between two or more persons over Internet is termed _____.
2. Pick the odd one out:
   (a) e-Book reader   (b) e-Text   (c) television channels   (d) e-business
3. Define e-Text.
4. Give an example for an e-Learning tool.

10.4 ICT applications in health care

Developments in ICT have contributed much in the field of medical practice and medical care. There is an enormous range of opportunities for significant cost reduction and service enhancement through IT applications in health care. Let us discuss some of the changes that ICT has brought about in the field of medicine and healthcare.

10.4.1 Medical equipments

Most of the modern medical equipment work with the support of computers. Scanners like CT and MRI depend on computers to process their data into a readable format. Micro-processors are used to control a variety of support devices. Computer guided lasers are used in surgery, even in the removal of brain tumours. Some hand-held devices are also available that help patients to lead normal life. A device for self-monitoring of sugar levels for people with diabetes is an example.

Fig. 10.8 : ICT in health care

10.4.2 Electronic Medical Record (EMR)

There has been a move to store all patients' records in digital format. This is known as Electronic Medical Record (EMR). An EMR is mostly used by doctors for diagnosis and treatment. There are many advantages in keeping patients record in electronic format.

Some of them are listed below:

- It is a more economical way of storing data.
- Access to medical history of patients will be easier and quicker.
- Transfer of records will be quicker between doctors and hospitals.
- Data can be examined to analyse the medical conditions of the patients.
- Prescription will be legible and are stored for later reference.

10.4.3 Web-based support/diagnosis

On the Internet you can find a wide variety of medical information, including information about available treatments/alternative therapies, access to medical research/journals, support groups for people suffering from particular conditions, etc .

Some websites also provide facility for symptom diagnosis - for people who wish to get primary awareness about their health problem without the help of a doctor. However there are some risks involved here. This can lead to incorrect diagnosis and cause false complacency or anxiety.
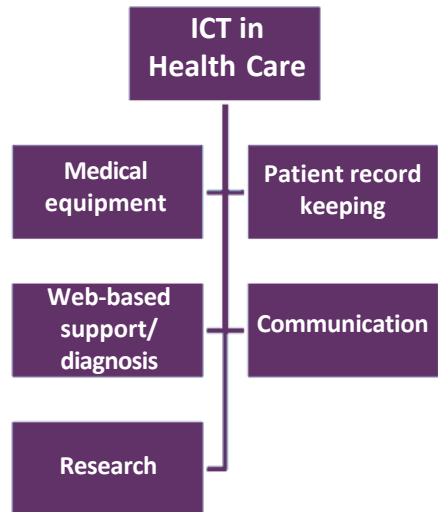
10.4.4 Telemedicine

Telemedicine has been used to share observations and prescriptions with the experts in the medical field. A surgeon in one part of the world can procure medical expertise from another who is far away, while performing surgery. It reduces cost and time. A patient in a critical condition can get attention from a surgeon who may not be physically available at the operation theatre. Wireless transmission helps medical personnel, transport workers and hospitals to keep in touch in emergency.

10.4.5 Research and development

Over recent years drug discovery has been a concern for many in the pharmaceutical industry. Escalating costs and increasing complexity suggest that traditional research and development methods are to be changed. Now the convergence of information and bio-technology is revolutionising drug discovery and design. Today, drugs meant for specific purposes can be developed with the use of advanced computers. Use of IT may radically alter the cost associated with the discovery of drug over the coming years. Thus IT helps in diagnosis, treatment of patients and better running of hospitals.

10.5 ICT enabled services

ICT enabled services, also called web enabled services cover the entire range of operations which make use of ICT for improving services provided by an organisation. These services provide a wide range of career options that include opportunities in call centres, BPOs, medical transcription, etc. Some of the common ICT enabled services are discussed in the following sections.

10.5.1 Business Process Outsourcing (BPO)

Outsourcing is a term used in the business field. It is a practice used by different companies to reduce costs by transferring portions of work to outside suppliers. It is the process of hiring another company to handle business activities for you. Business process outsourcing (BPO) is a subset of outsourcing. It involves contracting of the operations and responsibilities of specific business functions to a third-party service provider. Outsourcing sometimes involves transferring employees and asset from one firm to another.

Companies implement Business Process Outsourcing (BPO) because it increases their efficiency and saves cost. Processes such as human resource management, customer care services, payroll, etc. are some of the processes that are usually given to BPO.

10.5.2 Knowledge Process Outsourcing (KPO)

Knowledge Process Outsourcing (KPO) is a form of outsourcing. Here knowledge and information related work is carried out by a different company or subsidiary within the same organisation. It helps in saving costs and resources. Some of the core processes outsourced in KPO include: data search, data integration, market research, fraud analytics, etc. The KPO sector has experienced explosive growth in recent years. Industries that are increasingly using KPO include legal, financial services, management and consulting.

10.5.3 Call centre

Call centres are becoming a very important part of any business/service organisation. Call centres facilitate the public for the payment of bills, purchase of goods, etc. A **call centre** is a telephone service facility set up to handle a large number of both incoming and outgoing calls for an organisation.

Call centres are also called service centres, sales centres, contact centres, etc. Basic responsibility of a call centre is to handle the telephone calls for supporting various services provided by an organisation. Call centres are used by telemarketing companies, computer product help desks and any large organisation that uses the telephone to sell products and carryout services. Generally the calls can be classified as in-bound calls and out-bound calls. In-bound calls are usually meant to seek assistance, help, or to order, while out-bound calls are meant for sales promotion or other messages.

Some firms, however, specialise only in calls that are in-bound or out-bound. Most telephonic orders are handled by call centres and not by the manufacturers or suppliers of goods or services.

10.5.4 Teleconferencing

Teleconferencing is a meeting or conference held between two or more parties in remote locations, made possible by the use of IT infrastructure and services.

Teleconferencing is used to connect several different people from several different locations. It is used in educational, business and administrative fields. Saving time, reducing travel expense and including several people in a decision-making process are some of the advantages.

10.5.5 Video conferencing

Video conferencing is a type of teleconferencing wherein we can include the video of the parties involved in the conference. Videoconference allows people at remote locations join in a conference and share information. A video camera and a speakerphone are connected to a computer and the computers are connected to the network or Internet.

Video conferencing is a facility which enables participants in distant locations to take part in a conference by means of audio and video communication.

Let us sum up

Information Technology (IT) is a fast growing field. It is widely used in our day today activities. Business transactions, government activities, learning processes, research and medical field, etc. are some of the major areas where IT is substantially utilised. The explosion in the development of communication owes much to information technology. The use of IT helps people communicate with less effort and minimum cost. ICT is another term often used as an extended synonym for IT. Some of the ICT enabled services include BPO, KPO, Call centre, Teleconferencing and Video conferencing. Hence IT and ICT enabled services have become inseparable part of modern society.

Learning outcomes

After the completion of this chapter the learner will be able to

- list the facilities of e-Governance, e-Business, e-Learning.
- explain the tools used in e-Learning.
- list the ICT applications in health care.
- list various advantages and challenges faced in implementing e-Governance, e-Business, e-Learning applications.
- use some of the useful websites in each area.
- utilise various ICT enabled services.

Sample questions

Very short answer types

1. Name an electronic device using which we can easily read e-Text.

2. Write the full forms of BPO and KPO.

3. Name any two e-Learning tools.

Short answer types

1. List out different types of interactions in e-Governance.

2. Differentiate between BPO and KPO.

3. What are the advantages of e-Governance?

4. What are the duties of Akshaya?

5. Write down the function of Call centres.

Long answer types

1. What are the major challenges faced in the implementation of e-Learning ?
2. Compare the advantages and disadvantages of implementing e-Business?
3. Explain any three IT enabled services in detail.
4. Discuss in detail various uses of IT in health care field.
5. What is Common Service Centre (CSC)? List some of the services offered through CSC.

References

- Pradeep K. Sinha, Priti Sinha. *Computer Fundamentals* : BPB Publication

- V. Rajaraman (2010). *Fundamentals of Computers* : PHI Publication

- Thomas L. Floyd (2011). *Digital Fundamentals* : Pearson Education

- Craig Zacker, John Rourke (2008). *PC Hardware : The Complete Reference* : TMH Publication

- Abraham Silberschatz, Greg Gagne, Peter B. Galvin (2005). *Operating System Concepts* : John Wiley & Sons

- Herbert Schildt (2003). *C++ A beginners Guide* : McGraw-Hill Publication

- Bjarne Stroustrup (2013). *The C++ Programming Language* : Addison-Wesley Professional

- Robert Lafore (2009). *Object-Oriented Programming in C++* : Sams Publishing

- E. Balagurusamy (2008). *Object Oriented Programming with C++* : Tata McGraw-Hill Education

- Yashavant P Kanetkar(2000). *Let Us C++* : BPB Publication

- Andrew S. Tanenbaum, David J. Wetherall (2010). *Computer Networks* : Prentice Hall