

1

Anatomy of a Digital Computer

1.1 Introduction

We should keep in mind that a computer is a programmable machine. The two main characteristics of a computer are: (i) it responds to a specific set of instructions in a well-defined manner. (ii) It can execute a pre-recorded list of instructions (a program). Modern computers are electronic and digital. The actual machinery - wires, transistors and circuits is called hardware; the instructions and data are called software. All general purpose computers require the following hardware components:

- (i) Central Processing Unit (CPU): The 'brain' of the computer, the component that actually executes instructions.
- (ii) Memory : It enables a computer to store, at least temporarily, data and programs.
- (iii) Input device : Usually a keyboard or mouse is used to read data and programs into the computer.
- (iv) Output device: A display screen, printer, etc. that lets you see what the computer has accomplished.
- (v) Mass storage device: It allows a computer to permanently store large amounts of data. Common mass storage devices include disk drive and tape drive.

In addition to these components, many others make it possible for the basic components of a computer to work together efficiently.

1.2 Objectives

After going through this lesson, you will be in a position to:

- identify the basic components of a computer and their working
- explain the importance of various units of a computer
- learn how these units work together to accomplish a given job

1.3 Functions and Components of a Computer

To function properly, the computer needs both hardware and software. Hardware consists of the mechanical and electronic devices, which we can see and touch. The software consists of programs, the operating system and the data that reside in the memory and storage devices.

A computer does mainly the following four functions:

- Receive input —Accept data/information from outside through various input devices like the keyboard, mouse, scanner, etc.
- Process information—Perform arithmetic or logical operations on data/information.
- Produce output—Communicate information to the outside world through output devices like monitor, printer, etc.
- Store information—Store the information in storage devices like hard disk, floppy disks, CD, etc.

These four basic functions are responsible for everything that computers do. The hardware components of the computer specialize in any one of these functions.

Computer hardware falls into two categories: processing hardware and the peripheral devices. The Processing hardware consists of the Central Processing Unit (CPU), and as its name implies, is where the data processing is done. Peripheral devices allow people to interact with the CPU. Together, they make it possible to use the computer for a variety of tasks. Various parts of the computer have been shown in Fig. 1.1

Fig. 1.1 Various Parts of a Computer

Central Processing Unit (CPU)

This part of the computer that executes program instructions is known as the processor or Central Processing Unit (CPU). In a microcomputer, the CPU is based on a single electronic component, the microprocessor chip, within the system unit or system cabinet. The system unit also includes circuit boards, memory chips, ports and other components. A microcomputer's system cabinet will also house disk drives, hard disks, etc., but these are considered separate from the CPU.

The CPU has two parts —The Control Unit (CU) and the Arithmetic Logic Unit (ALU). In a microcomputer, both are on a single microprocessor chip.

Control Unit (CU)

The control unit tells the rest of the computer system how to carry out a program's instructions. It directs the movement of electronic signals between memory - which temporarily holds data, instructions and processes information - and the ALU. It also directs these control signals between the CPU and input/output devices.

Arithmetic - Logic Unit (ALU)

Arithmetic Logic Unit, usually called the ALU, performs two types of operations - arithmetical and logical. Arithmetical operations are the fundamental

mathematical operations consisting of addition, subtraction, multiplication and division. Logical operations consist of comparisons. That is two pieces of data are compared to see whether one is equal to, less than, or greater than the other.

Memory

Memory - also known as the primary storage or main memory - is a part of the microcomputer that holds data and instructions.

Part of the contents of the memory is held only temporarily, that is, it is stored only as long as the microcomputer is turned on. When you turn the machine off, the contents are lost. The capacity of the memory to hold data and program instructions varies in different computers. The original IBM PC could hold approximately several thousand characters of data or instructions only. But modern microcomputers can hold millions or even billions of characters in their memory.

Registers

Computers also have several additional storage locations called registers. These appear in the Control Unit and ALU and make processing more efficient. Registers are a sort of special hi-speed storage areas that hold data and instructions temporarily during processing. They are parts of the Control Unit and ALU rather than the memory. Their contents can, therefore be handled much faster than the contents of the memory.

Addresses

To locate the characters of data or instructions in the main memory, the computer stores them in locations known as addresses. A unique number designates each address. Addresses can be compared to post office mailboxes. Their numbers remain the same, but contents continuously change.

How the CPU and Memory work

The working of the CPU and memory is shown in Fig. 1.2. Various steps involved for multiplying two numbers is shown in that figure and explained below.

1. The control unit recognizes that the program has been loaded into the memory. It begins to execute the first step in the program.
-

2. The program tells the user, "Enter 1st Number".
 3. The user types the number 100 on the keyboard. An electronic signal is sent to the CPU.
 4. The control unit recognizes this signal and routes the signal to an address in memory - say address 7.
 5. After completing the above instruction, the next instruction tells the user, "Enter 2nd Number."
 6. The user types the number 4 on the keyboard. An electronic signal is sent to the CPU.
 7. The control unit recognizes this signal and routes it to memory address 8.
 8. The next program instruction is executed - "Multiply 1st and 2nd Numbers."
 9. To execute this instruction, the control unit informs the ALU that two numbers are coming and the ALU is to multiply them. The control unit next sends to the ALU a copy of the contents of address 7 (100) and address 8(4).
 10. ALU performs the multiplication : $100 \times 4 = 400$
 11. The control unit sends a copy of the multiplied result (400:) back to memory to store it in address 9.
 12. The next program instruction is executed : "Print the Result."
 13. To execute this instruction, the control unit sends the contents of the address 9 (400) to the monitor.
 14. Monitor displays the value 400.
 15. Final instruction is executed: "End". The program is complete.
-

Fig. 1.2 Working of the CPU and Memory

Fig. 1.2 is a simplified version of the actual processing activity. The actual working of the CPU and memory is much more complicated. For example, there are actually many more memory addresses - thousands to millions that cannot be shown here. Moreover, the addresses are in a form the computer can interpret - electronic signals rather than the numbers and letters shown here.

Now we shall discuss in details about various components of a computer system.

1.4 Input devices

An input device presents data to the processing unit in a machine-readable form. Although the keyboard is a common input device for a small computer, a system may also support various other input devices such as Optical Character Recognition (OCR), Magnetic Ink Character Recognition (MICR), mark sense reader, etc.

1.4.1 Key board

The keyboard is very much like a standard typewriter keyboard with a few additional keys. The basic QWERTY layout of characters is maintained to make it easy for the trained typist to use the system. The additional keys are included to perform certain special functions such as loading a program, edition a text, etc. These are known as function keys that vary in number from system to system.

1.4.2 Optical Character Recognition

Often abbreviated as OCR, optical character recognition involves reading text from paper and translating the images into a form that the computer can manipulate. An OCR system enables you to take a book or a magazine article and feed it directly into an electronic computer file.

1.4.3 Magnetic Ink Character Recognition (MICR)

An MICR can identify characters printed with a special ink that contain particles of magnetic material. This device particularly finds applications in banking industry. Since the MICR system can recognise only certain character styles, the characters have to be accurately formed.

1.4.4 Optical Mark Recognition (OMR)

Optical mark recognition, also called mark sense reader, is a technology where an OMR device senses the presence or absence of a mark, such as pencil mark. OMR is widely used in tests such as aptitude tests.

1.4.5 Bar Code Reader

These devices are generally available in super markets, bookshops, etc. Bar-code readers are photoelectric scanners that read the bar codes or vertical zebra striped marks, printed on product containers. Supermarkets use a bar code system called the Universal Product Code (UPC). The bar code identifies the product to the supermarket's computer which has a description and the latest price of the product. The computer automatically tells the Point of Sales (POS) terminal what the price is.

1.4.6 Digitizing Tablet

This is an input device that enables you to enter drawings and sketches into a computer. A digitizing tablet consists of an electronic tablet and a cursor or pen. A cursor (also called a puck) is similar to a mouse, except that it has a window with cross hairs for pinpoint placement, and it can have as many as 16 buttons. A pen (also called a stylus) looks like a simple ballpoint pen but uses an electronic head instead of ink. The tablet contains electronic field that enables it to detect movement of the cursor or pen and translate the movements into digital signals that it sends to the computer. Digitizing tables are also called digitizers, graphics tables, touch tables or simply tables.

1.4.7 Scanner

Scanner is an input device that can read text or illustrations printed on paper and translate the information into a form that the computer can use. A scanner works by digitizing an image - dividing it into a grid of boxes and representing each box with either a zero or a one, depending on whether the box is filled in. The resulting matrix of bits, called a bit map, can then be stored in a file, displayed on a screen and manipulated by programs. Optical scanners do not distinguish text from illustrations, they represent all images as bit maps. Therefore, you cannot directly edit text that has been scanned. To edit text read by an optical scanner, you need an optical character recognition (OCR) system to translate the image into ASCII characters. Most optical scanners sold today come with OCR packages.

1.4.8 Mouse

Mouse is a device that controls the movement of the cursor or pointer on a display screen. It is a small object you can roll along a hard and flat surface. As you move the mouse, the pointer on the display screen moves in the same direction. Mouse contains at least one button and sometimes as many as three, which have different functions depending on what program is running.

1.4.9 Light Pen

Light pen is an input device that utilizes a light-sensitive detector to select objects on a display screen.

1.4.10 Speech input devices

Speech or voice input devices convert a person's speech into digital form. These input devices, when combined with appropriate software, form voice recognition systems. These systems enable users to operate microcomputers using voice commands.

Intext Questions 1.1

1. What are the four basic functions performed by the computer ?
 2. What is a scanner ? Explain briefly.
 3. Choose the correct answer :
 - (a) The task of performing arithmetic and logical operations is performed by.
(i) ALU (ii) Editor (iii) storage (iv) output
 - (b) The ALU and CU are jointly known as
(i) RAM (ii) ROM (iii) CPU (iv) none of the above
 - (c) The process of producing results from the data for getting useful information is called.
(i) output (ii) input (iii) processing (iv) storage
-

1.5 Memory Unit

There are two kinds of computer memory: primary and secondary. Primary memory is accessible directly by the processing unit. RAM is an example of primary memory. As soon as the computer is switched off, the contents of the primary memory is lost. You can store and retrieve data much faster with primary memory compared to secondary memory. Secondary memory such as floppy disk, magnetic disk, etc., is located outside the computer. Primary memory is more expensive than secondary memory.

Computer memory is used to store two things: i) instructions to execute a program and ii) data. When the computer is doing any job, the data that have to be processed are stored in the primary memory. This data may come from an input device like keyboard or from a secondary storage device like a floppy disk.

As program or the set of instructions is kept in primary memory, the computer is able to follow instantly the set of instructions. For example, when you book ticket from railway reservation counter, the computer has to follow the same steps : take the request, check the availability of seats, calculate fare, wait for money to be paid, store the reservation and get the ticket printed out. The program containing these steps is kept in memory of the computer and is followed for each request.

But inside the computer, the steps followed are quite different from what we see on the monitor or screen. In computer's memory both programs and data are stored in the binary form. You have already been introduced with decimal number system, that is the numbers 1 to 9 and 0. The binary system has only two values 0 and 1. These are called bits. As human beings we all understand decimal system but the computer can only understand binary system. It is because a large number of integrated circuits inside the computer can be considered as switches, which can be made ON, or OFF. If a switch is ON it is considered 1 and if it is OFF it is 0. A number of switches in different states will give you a message like this: 110101.....10. So the computer takes input in the form of 0 and 1 and gives output in the same form, i.e., 0 and 1 only. Is it not absurd if the computer gives outputs as 0's & 1's only? But you do not have to worry about. Every number in binary system can be converted to decimal system and vice versa; for example, 1010 meaning decimal 10. Therefore, it is the computer that takes information or data in decimal form from you, converts it into binary form, processes it

producing output in binary form and finally again converts the output to decimal form.

The primary memory as you know in the computer is in the form of IC's (Integrated Circuits). These circuits are called Random Access Memory (RAM). Each of RAM's location stores one byte of information. (One byte is equal to 8 bits). A bit is an acronym for binary digit, which stands for one binary piece of information. This can be either 0 or 1. You will know more about RAM later. The Primary or internal storage unit is made up of several small storage locations called cells. Each of these cells can store a fixed number of bits called word length.

Each cell has a unique number assigned to it called the address of the cell and it is used to identify the cells. The address starts at 0 and goes up to (N-1). You should know that the memory is like a large cabinet containing as many drawers as there are addresses in memory. Each drawer contains a word and the address is written on outside of the drawer.

1.5.1 Capacity of Primary Memory

You know that each cell of memory contains one character or 1 byte of data. So the capacity is defined in terms of byte or words. Thus 64 kilobyte (KB) memory is capable of storing $64 \times 1024 = 32,768$ bytes. (1 kilobyte is equal to 1024 bytes). A memory size ranges from few kilobytes in small systems to several thousands kilobytes in large mainframe and super computer. In your personal computer you will find memory capacity in the range of 8 MB, 16 MB and even 128 MB (MB = Million bytes). The following terms related to memory of computer are discussed below:

- (a) **Random Access Memory (RAM)** : The primary storage is referred to as Random Access Memory (RAM) because it is possible to randomly select and use any location of the memory directly to store and retrieve data. It takes same time to any address of the memory as the first address. It is also called read/write memory. The storage of data and instructions inside the primary storage is temporary. It disappears from RAM as soon as the power to the computer is switched off. The memories, which lose their content on failure of power supply, are known as volatile memories. So now we can say that RAM is volatile memory.
-

- (b) **Read Only Memory (ROM)** : There is another memory in computer, which is called Read Only Memory (ROM). Again it is the ICs inside the PC that form the ROM. The storage of program and data in the ROM is permanent. The ROM stores some standard processing programs supplied by the manufacturers to operate the personal computer. The ROM can only be read by the CPU but it cannot be changed. The basic input/output program is stored in the ROM that examines and initializes various equipment attached to the PC when the switch is turned ON. The memories, which do not lose their content on failure of power supply, are known as non-volatile memories. ROM is non-volatile memory.
 - (c) **PROM** : There is another type of primary memory in computer, which is called Programmable Read Only Memory (PROM). You know that it is not possible to modify or erase programs stored in ROM, but it is possible for you to store your program in PROM chip. Once the programs are written it cannot be changed and remain intact even if power is switched off. Therefore, programs or instructions written in PROM or ROM cannot be erased or changed.
 - (d) **EPROM** : This stands for Erasable Programmable Read Only Memory, which overcomes the problem of PROM and ROM. EPROM chip can be programmed time and again by erasing the information stored earlier in it. Information stored in EPROM can be erased by exposing it to ultraviolet light. This memory can be reprogrammed using a special programming facility. When the EPROM is in use, information can only be read.
 - (e) **EAPROM** : This stands for Electrically Alterable Programmable Read Only Memory. This concept is same as that of EPROM. The only difference is that the memory can be altered using electrical signals. The whole of the memory need not be erased.
 - (f) **Cache Memory** : The speed of CPU is extremely high compared to the access time of main memory. Therefore, the performance of CPU decreases due to the slow speed of main memory. To decrease the mismatch in operating speed, a small memory chip is attached between CPU and Main memory whose access time is very close to the processing speed of CPU. It is called CACHE memory. CACHE memories are
-

accessed much faster than conventional RAM. It is used to store programs or data currently being executed or temporary data frequently used by the CPU. It is also very expensive to have bigger size of cache memory. Its size is therefore, normally kept small.

- (g) **Registers** : The CPU processes data and instruction with high speed. There is also movement of data between various units of the computer. It is necessary to transfer the processed data with high speed. So the computer uses a number of special memory units called registers. They are not part of the main memory but they store data or information temporarily and pass it on as directed by the control unit.

Intext Questions 1.2

1. Distinguish between bit and byte.
 2. Define volatile and non-volatile memory.
 3. Write True or False:
 - (a) There are two kinds of computer memory: primary and secondary.
 - (b) The computer can understand decimal system also.
 - (c) The storage of program and data in the RAM is permanent.
 - (d) PROM is secondary memory.
 - (e) The memories which do not lose their content on failure of power supply are known as non-volatile memories.
-

1.6 Secondary Storage

You are now clear that the operating speed of primary memory or main memory should be as fast as possible to cope up with the CPU speed. These high-speed storage devices are very expensive and hence the cost per bit of storage is also very high. Again, the storage capacity of the main memory is also very limited. Often it is necessary to store hundreds of millions of bytes of data for the CPU to process. Therefore, additional memory is required in all the computer systems. This memory is called auxiliary memory or secondary storage.

In this type of memory the cost per bit of storage is low. However, the operating speed is slower than that of the primary memory. Huge volume of data are stored here on permanent basis and transferred to the primary storage as and when required. Most widely used secondary storage devices are magnetic tapes, magnetic disks and floppy disks.

1.6.1 Magnetic Tape

Magnetic tapes are used for large computers like mainframe computers where large volume of data is stored for a longer time. In PCs also you can use tapes in the form of cassettes. The cost of storing data in tapes is inexpensive. Tapes consist of magnetic materials that store data permanently. It can be 12.5 mm to 25 mm wide plastic film-type and 500 meter to 1200 meter long which is coated with magnetic material. The deck is connected to the central processor and information is fed into or read from the tape through the processor. It is similar to cassette tape recorder.

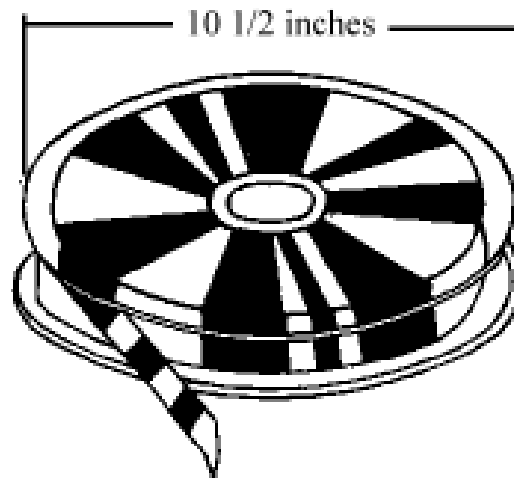


Fig.1.3 Magnetic Tape

Advantages of Magnetic Tape

- **Compact** : A 10-inch diameter reel of tape is 2400 feet long and is able to hold 800, 1600 or 6250 characters in each inch of its length. The maximum capacity of such type is 180 million characters. Thus data are stored much more compact on tape
-

- **Economical** : The cost of storing characters on tape is very less as compared to other storage devices.
- **Fast** : Copying of data is easier and fast.
- **Long term Storage and Re-usability** : Magnetic tapes can be used for long term storage and a tape can be used repeatedly with out loss of data.

1.6.2 Magnetic Disk

You might have seen the gramophone record, which is circular like a disk and coated with magnetic material. Magnetic disks used in computer are made on the same principle. It rotates with very high speed inside the disk drive. Data are stored on both the surface of the disk. Magnetic disks are most popular for direct access storage. Each disk consists of a number of invisible concentric circles called tracks. Information is recorded on tracks of a disk surface in the form of tiny magnetic spots. The presence of a magnetic spot represents one bit (1) and its absence represents zero bit (0). The information stored in a disk can be read many times without affecting the stored data. So the reading operation is non-destructive. But if you want to write a new data, then the existing data is erased from the disk and new data is recorded.

1.6.3 Floppy Disk

It is similar to magnetic disk discussed above. It is 3.5 inch in diameter. The capacity of a 3.5 inch floppy is 1.44 mega bytes. It is cheaper than any other storage devices and is portable. The floppy is a low cost device particularly suitable for personal computer system.



Fig. 1.4 Floppy Diskette

1.6.4 Optical Disk

With every application and software there is greater demand for memory capacity. It is the necessity to store large volume of data that has led to the development of optical disk storage medium. Optical disks can be divided into the following categories.

- (i) **Compact Disk/Read Only Memory (CD-ROM):** CD-ROM disks are made of reflective metals. CD-ROM is written during the process of manufacturing by high power laser beam. Here the storage density is very high, storage cost is very low and access time is relatively fast. Each disk is approximately 4 1/2 inches in diameter and can hold over 600 MB of data. As the CD-ROM can be read only we cannot write or make changes into the data contained in it.
- (ii) **Write Once Read Many (WORM):** The inconvenience that we can not write any thing in to a CD-ROM is avoided in WORM. A WORM allows the user to write data permanently on the disk. Once the data is written it can never be erased without physically damaging the disk. Here data can be recorded from keyboard, video scanner, OCR equipment and other devices. The advantage of WORM is that it can store vast amount of data amounting to gigabytes. Any document in a WORM can be accessed very fast.
- (iii) **Erasable Optical Disk :** These are optical disks where data can be written, erased and re-written. This also applies a laser beam to write and re-write the data. These disks may be used as alternatives to traditional disks. Erasable optical disks are based on a technology known as magnetic optical (MO). To write a data bit on the erasable optical disk the MO drive's laser beam heats a tiny, precisely defined point on the disk's surface and magnetises it.

1.7 Arithmetic Logic Unit (ALU)

This unit is used to perform all the arithmetic and logic operations, such as addition, multiplication, comparison, etc. For example, consider the addition of two numbers A and B. The control unit will select the number A from its location in the memory and load it into the arithmetic logic unit. The result will

then be stored in the memory or retained in the arithmetic unit for further calculations.

1.8 Control Unit (CU)

This unit coordinates the activities of all the other units in the system. Its main functions are:

- (i) To control the transfer of data and information between various units;
- (ii) To initiate appropriate functions by the arithmetic logic unit.

The user's program provides the basic control instructions. Conceptually, the control unit fetches instructions from the memory, decodes them and directs the various units to perform the specified functions.

1.9 Output Devices

Output devices receive information from the CPU and present it to the user in the desired form. Output devices include display screen, loudspeakers, printers, plotters, etc.

1.9.1 Display Screen

When a program is keyed in, the screen (which is similar to a television screen) displays the characters. The user can read the program line by line and make corrections before it is stored or printed on a printer. It is also possible to bring to the screen a portion of the program stored in the external storage for editing.

Screen sizes differ from system to system. The standard size is 24 lines by 80 characters. Most systems have provision for scrolling. This facilitates the user to move the text vertically or horizontally on the screens thus bringing to the screen the hidden text. Thus the user can scan through the entire file either to review or to select a particular portion. The cursor on the screen is controlled by the cursor keys on the keyboard,

1.9.2 Printer

Printer is a device that prints text or illustrations on paper and in many cases on

transparencies and other media. There are many different types of printers. In terms of the technology utilised, printer fall into the following categories.

(i) Ink-jet Printer

Ink-jet printers work by spraying ionized ink on a sheet of paper. Magnetized plates in the ink's path direct the ink onto the paper in the desired shapes. Ink-jet printers are capable of producing high quality print approaching to that produced by laser printers. A typical ink-jet printer provides a resolution of 300 dots per inch, although some newer models offer higher resolutions.

In general, the price of ink-jet printers is lower than that of laser printers. However, they are also considerably slower. Another drawback of ink-jet printers is that they require a special type of ink that is apt to smudge on inexpensive copier paper.

Because ink-jet printers require smaller mechanical parts than laser printers, they are specially popular as portable printers. In addition, colour ink-jet printers provide an inexpensive way to print full-colour documents.



Fig. 1.5 Ink-jet Printer

(ii) Laser Printer

Laser Printer utilizes a laser beam to produce an image on a drum. The light of the laser alters the electrical charge on the drum wherever it hits. The drum is then rolled through a reservoir of toner, which is picked up by the charged portions of the drum. Finally, the toner is transferred to the paper through a combination of heat and pressure. This is also the way copy machines work.

Because an entire page is transmitted to a drum before the toner is applied, laser printers are sometimes called page printers. There are two other types of page printers that fall under the category of laser printers even though they do not use lasers at all. One uses an array of LEDs to expose the drum, and the other uses LCDs. Once the drum is charged, however, they both operate like a real laser printer.

One of the chief characteristics of laser printers is their resolution - how many dots per inch (dpi) they lay down. The available resolutions range from 300 dpi at the low end to 1,200 dpi at the high end. By comparison, offset printing usually prints at 1,200 or 2,400 dpi. Some laser printers achieve higher resolutions with special techniques known generally as resolution enhancement.

In addition to text, laser printers are very adept at printing graphics. However, you need significant amounts of memory in the printer to print high-resolution graphics. To print a full-page graphic at 300 dpi, for example, you need at least 1 MB (megabyte) of printer RAM. For a 600-dpi graphic, you need at least 4 MB RAM.

The speed of laser printers ranges from about 4 to 20 pages of text per minute (ppm). A typical rate of 6 ppm is equivalent to about 40 characters per second (cps).

(iii) Line Printer

Line printers are high-speed printers capable of printing an entire line at one time. A fast line printer can print as many as 3,000 lines per minute. The disadvantages of line printers are that they can print only one font, they cannot print graphics, the print quality is low, and they are very noisy.

(iv) Thermal printer

Thermal printers are printers that produce images by pushing electrically heated pins against special heat-sensitive paper. Thermal printers are inexpensive and are used in most calculators and many fax machines. They produce low-quality print, and the paper tends to curl and fade after a few weeks or months.

1.9.3 Plotter

Plotter is a device that draws pictures on paper based on commands from a computer. Plotters differ from printers in that they draw lines using a pen. As a result, they can produce continuous lines, whereas printers can only simulate lines by printing a closely spaced series of dots. Multicolour plotters use different-coloured pens to draw different colours.



Fig. 1.6 Plotter

In general, plotters are considerably more expensive than printers. They are used in engineering applications where precision is mandatory.

1.9.4 Sound Cards & Speakers

An expansion board that enables a computer to manipulate and output sounds. Sound cards are necessary for nearly all CD-ROMs and have become commonplace on modern personal computers. Sound cards enable the computer to output sound through speakers connected to the board, to record sound input from a microphone connected to the computer, and manipulate sound stored on a disk.

Nearly all sound cards support MIDI, a standard for representing music electronically. In addition, most sound cards are Sound Blaster-compatible, which means that they can process commands written for a Sound Blaster card, the de facto standard for PC sound.



Fig. 1.7 Sound Card and Speakers

1.9.5 3D-Audio

3D audio is a technique for giving more depth to traditional stereo sound. Typically, 3D sound, or 3D audio, is produced by placing a device in a room with stereo speakers. The device dynamically analyses the sound coming from the speakers and sends feedback to the sound system so that it can readjust the sound to give the impression that the speakers are further apart.

3D audio devices are particularly popular for improving computer audio where the speakers tend to be small and close together. There are a number of 3D audio devices that can be attached to a computer's sound card.

Intext Questions 1.3

1. Define the following terms briefly:
 - (a) ALU
 - (b) Ink-jet Printer
 - (c) Plotter

 2. Fill in the blanks:
 - (a) unit coordinates the activities of all the other units in the system.
 - (b) The standard size of display screen is lines by characters.
 - (c) printer utilizes a laser beam to produce an image on a drum.
-

- (d) audio is a technique for giving more depth to traditional stereo sound.
-

1.10 What you have Learnt

In this lesson you learnt in detail about various components of a computer. You also learnt how various units of computer interact with each other to accomplish a given job.

1.11 Terminal Questions

1. What are the major components of a computer ?
2. What is CPU and how does it work? Explain briefly.
3. What is a plotter and how does it work?
4. Describe the various types of printers briefly.
5. Differentiate between the following:
 - (a) RAM and ROM
 - (b) Volatile and non-volatile memory.

1.12 Feedback to In-Text Questions

In-Text Questions 1.1

1. Four basic functions are:
 - (a) Receive input
 - (b) Process information
 - (c) Produce output
 - (d) Store information for future use
 2. Scanner is an input device that can read text or illustration printed on paper
-

and translate the information into a form that the computer can use

3. (a) ALU (b) CPU (c) processing

In-Text Question 1.2

1. A bit is an acronym for binary digit, which stands for one binary piece of information. This can be either 0 or 1. Byte consists of 8 bits. Each of RAM's location stores one byte of information.
2. The memories, which lose their content on failure of power supply, are known as volatile memories. RAM is a volatile memory. On the other hand ROM is a non-volatile memory because it does not lose its contents on failure of power supply.
3. (a) True (b) False (c) False (d) False (e) True

In-Text Questions 1.3

1. (a) ALU is used to perform all the arithmetic and logic operations such as addition, subtraction multiplication, comparison, etc.
(b) Ink-jet printers work by spraying ionized ink at a sheet of paper. These printers are capable of producing high quality print nearer to that produced by laser printers.
(c) Plotter is a device that draws pictures on paper based on commands from a computer.
 2. (a) Control (b) 24, 80 (c) Laser (d) 3D
-



2

Data Processing Concept

2.1 Introduction

Each organisation, regardless of its size or purpose, generates data to keep a record of events and transactions that take place within the business. Generating and organising this data in a useful way is called data processing. In this lesson, we shall discuss about various terms such as data, information, data processing and data processing system.

2.2 Objectives

After going through this lesson, you will be in a position to

- define the concepts of data, information and data processing
- explain various data processing activities
- utilise data processing cycle
- explain data elements, records, files and databases.

2.3 Data

The word "data" is the plural of datum, which means fact, observation, assumption or occurrence. More precisely, data are representations of facts pertaining to

people, things, ideas and events. Data are represented by symbols such as letters of the alphabets, numerals or other special symbols.

2.4 Data Processing

Data processing is the act of handling or manipulating data in some fashion. Regardless of the activities involved in it, processing tries to assign meaning to data. Thus, the ultimate goal of processing is to transform data into information.

Data processing is the process through which facts and figures are collected, assigned meaning, communicated to others and retained for future use. Hence we can define data processing as a series of actions or operations that converts data into useful information. We use the term 'data processing system' to include the resources that are used to accomplish the processing of data.

2.5 Information

Information, thus can be defined as “data that has been transformed into a meaningful and useful form for specific purposes”. In some cases data may not require any processing before constituting information. However, generally, data is not useful unless it is subjected to a process through which it is manipulated and organised, its contents analyzed and evaluated. Only then data becomes information.

There is no hard and fast rule for determining when data becomes information. A set of letters and numbers may be meaningful to one person, but may have no meaning to another. Information is identified and defined by its users.

For example, when you purchase something in a departmental store, a number of data items are put together, such as your name, address articles you bought, the number of items purchased, the price, the tax and the amount you paid. Separately, these are all data items but if you put these items together, they represent information about a business transaction.

2.6 Data Processing Activities

As discussed above, data processing consists of those activities which are necessary to transform data into information. Man has in course of time devised

certain tools to help him in processing data. These include manual tools such as pencil and paper, mechanical tools such as filing cabinets, electromechanical tools such as adding machines and typewriters, and electronic tools such as calculators and computers. Many people immediately associate data processing with computers. As stated above, a computer is not the only tool used for data processing, it can be done without computers also. However, computers have outperformed people for certain tasks. There are some other tasks for which computer is a poor substitute for human skill and intelligence.

Regardless to the type of equipment used, various functions and activities which need to be performed for data processing can be grouped under five basic categories as shown in Fig. 2.1

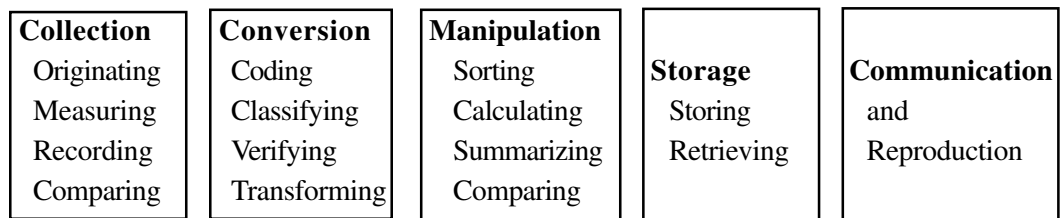


Fig. 2.1 Data Processing Activities

We will now describe each of these activities:

Deptt. 5 Employee Time Sheet for Week ending 21/07/2001							
Hours Worked							
Emp No.	Name	Mon	Tues	Wed	Thurs	Fri	Total Hours
110	Hariom	8	8	8	8	8	40
150	Dinesh	8	8	8	8	6	38
160	Rajesh	8	4	8	8	8	36
170	Pankaj	8	6	8	5	8	35

Fig. 2.2 The source document for a payroll application is the weekly time sheet

(a) Collection

Data originates in the form of events transaction or some observations. This data is then recorded in some usable form. Data may be initially recorded on paper source documents 2.2 and then converted into a machine usable form for processing. Alternatively, they may be recorded by a direct input device in a paperless, machine-readable form. Data collection is also termed as data capture.

(b) Conversion

Once the data is collected, it is converted from its source documents to a form that is more suitable for processing. The data is first codified by assigning identification codes. A code comprises of numbers, letters, special characters, or a combination of these. For example, an employee may be allotted a code as 52-53-162, his category as A class, etc. It is useful to codify data, when data requires classification. To classify means to categorize, i.e., data with similar characteristics are placed in similar categories or groups. For example, one may like to arrange accounts data according to account number or date. Hence a balance sheet can easily be prepared.

After classification of data, it is verified or checked to ensure the accuracy before processing starts.

After verification, the data is transcribed from one data medium to another. For example, in case data processing is done using a computer, the data may be transformed from source documents to machine sensible form using magnetic tape or a disk.

(c) Manipulation

Once data is collected and converted, it is ready for the manipulation function which converts data into information. Manipulation consists of following activities:

Sorting

It involves the arrangement of data items in a desired sequence. Usually, it is easier to work with data if it is arranged in a logical sequence. Most

often, the data are arranged in alphabetical sequence. Sometimes sorting itself will transform data into information. For example, a simple act of sorting the names in alphabetical order gives meaning to a telephone directory. The directory will be practically worthless without sorting.

Business data processing extensively utilises sorting technique. Virtually all the records in business files are maintained in some logical sequence. Numeric sorting is common in computer-based processing systems because it is usually faster than alphabetical sorting.

Calculating

Arithmetic manipulation of data is called calculating. Items of recorded data can be added to one another, subtracted, divided or multiplied to create new data as shown in fig. 2.2(a). Calculation is an integral part of data processing. For example, in calculating an employee's pay, the hours worked multiplied by the hourly wage rate gives the gross pay. Based on total earning, income-tax deductions are computed and subtracted from gross-pay to arrive at net pay.

Weekly Payroll Summary Report 07/07/2005					
Dept.	Employee Number	Name	Hours Worked	Pay Rate	Gross Wages
2	170	Pankaj	34	10.00	340.00
	175	Rekha	32	9.00	288.00
	158	Param	20	5.00	100.00
	160	Rajesh	36	8.00	288.00
	165	Sanju	45	9.00	405.00
	159	Savini	25	3.00	75.00
Department 2 Total			192		1496.00
4	110	Hari Om	33	7.75	255.75
	115	Vimla	40	9.25	370.00
Department 4 Total			73		625.75
8	150	Dinesh	38	7.00	266.00
	155	Upma	40	9.45	390.00
	156	Arunima	32	8.50	272.00
	157	Ritwik	40	8.00	320.00
Department 8 Total			150		1,248.00
Company Total			415		3,369.75

Fig. 2.2(a) : Weekly Payroll Summary

Summarizing

To summarize is to condense or reduce masses of data to a more usable and concise form as shown in fig. 2.2(b). For example, you may summarize a lecture attended in a class by writing small notes in one or two pages. When the data involved is numbers, you summarize by counting or accumulating the totals of the data in a classification or by selecting strategic data from the mass of data being processed. For example, the summarizing activity may provide a general manager with sales-totals by major product line, the sales manager with sales totals by individual salesman as well as by the product line and a salesman with sales data by customer as well as by product line.

Department Payroll Summary for Week of 07-07-2005		
Department	Hours	Gross Wages
2	192	1,496.00
4	73	625.75
8	150	1,248.00
Company Total	415	3,369.75

Fig. 2.2(b) : Department Payroll Summary

Comparing

To compare data is to perform an evaluation in relation to some known measure. For example, business managers compare data to discover how well their companies are doing. They many compare current sales figures with those for last year to analyze the performance of the company in the current month.

(d) Managing the Output Results

Once data has been captured and manipulated following activities may be carried out :

Storing

To store is to hold data for continued or later use. Storage is essential for any organised method of processing and re-using data. The storage

mechanisms for data processing systems are file cabinets in a manual system, and electronic devices such as magnetic disks/magnetic tapes in case of computer based system. The storing activity involves storing data and information in organised manner in order to facilitate the retrieval activity. Of course, data should be stored only if the value of having them in future exceeds the storage cost.

Retrieving

To retrieve means to recover or find again the stored data or information. Retrieval techniques use data storage devices. Thus data, whether in file cabinets or in computers can be recalled for further processing. Retrieval and comparison of old data gives meaning to current information.

(e) Communication

Communication is the process of sharing information. Unless the information is made available to the users who need it, it is worthless. Thus, communication involves the transfer of data and information produced by the data processing system to the prospective users of such information or to another data processing system. As a result, reports and documents are prepared and delivered to the users. In electronic data processing, results are communicated through display units or terminals.

(f) Reproduction

To reproduce is to copy or duplicate data or information. This reproduction activity may be done by hand or by machine.

Intext Questions 2.1

1. Differentiate between the following :
 - (a) Data and Information
 - (b) Data processing and Data processing system
 2. Define the following terms briefly:
 - (a) Sorting
 - (b) Summarizing
-

2.7 The Data Processing Cycle

The data processing activities described above are common to all data processing systems from manual to electronic systems. These activities can be grouped in four functional categories, viz., data input, data processing, data output and storage, constituting what is known as a data processing cycle.

(i) Input

The term input refers to the activities required to record data and to make it available for processing. The input can also include the steps necessary to check, verify and validate data contents.

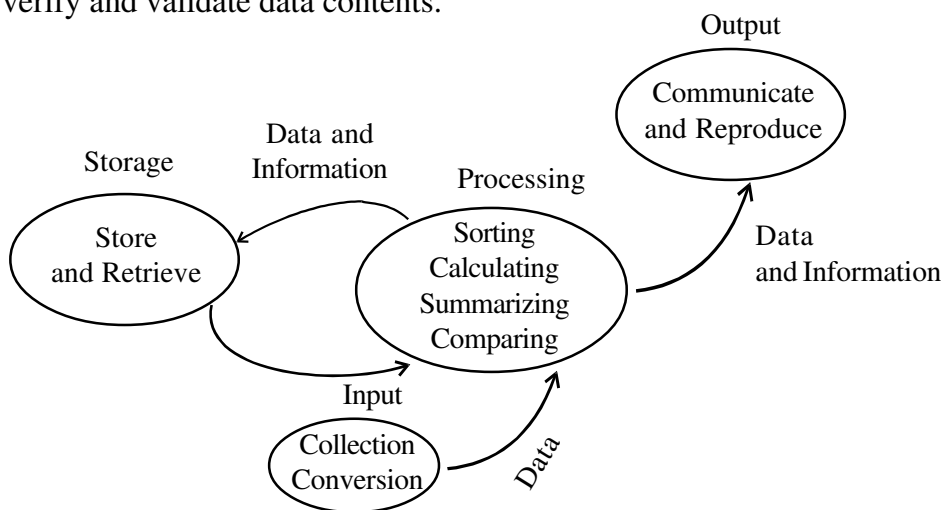


Fig. 2.3 Data Processing Cycle

(ii) Processing

The term processing denotes the actual data manipulation techniques such as classifying, sorting, calculating, summarizing, comparing, etc. that convert data into information.

(iii) Output

It is a communication function which transmits the information, generated after processing of data, to persons who need the information. Sometimes output also includes decoding activity which converts the electronically generated information into human-readable form.

(iv) Storage

It involves the filing of data and information for future use.

The above mentioned four basic functions are performed in a logical sequence as shown in Fig. 2.3 in all data processing systems.

2.8 Computer Processing Operations

A computer can perform only the following four operations which enable computers to carry out the various data processing activities we have just discussed.

(a) Input/Output operations

A computer can accept data (input) from and supply processed data (output) to a wide range of input/output devices. These devices such as keyboards, display screens, and printers make human-machine communication possible.

(b) Calculation and text manipulation Operations

Computer circuits perform calculations on numbers. They are also capable of manipulating numerics and other symbols used in text with equal efficiency.

(c) Logic/Comparison Operations

A computer also possesses the ability to perform logic operations. For example, if we compare two items represented by the symbols A and B, there are only three possible outcomes.

A is less than B ($A < B$); A is equal to B ($A = B$); or A is greater than B ($A > B$).

A computer can perform such comparisons and the, depending on the result, follow a predetermined path to complete its work. This ability to compare is an important property of computers.

(d) Storage and Retrieval Operations

Both data and program instructions are stored internally in a computer. Once they are stored in the internal memory, they can be called up quickly or retrieved, for further use.

2.9 Data Processing System

The activity of data processing can be viewed as a "system". According to James O'brien a system can be defined as "a group of interrelated components that seeks the attainment of a common goal by accepting inputs and producing outputs in an organised process". For example, a production system accepts raw material as input and produces finished goods as output.

Similarly, a data processing system can be viewed as a system that uses data as input and processes this data to produce information as output.



There are many kinds of data processing systems. A manual data processing system is one that utilizes tools like pens, and filing cabinets. A mechanical data processing system uses devices such as typewriters, calculating machines and book-keeping machines. Finally, electronic data processing uses computers to automatically process data.

2.10 Data Organisation

Having discussed the Data Processing Cycle (also called Information Processing Cycle) and the components of a computer, we will now describe how data is organised before processing on a computer. Data can be arranged in a variety of ways, but a hierarchical approach to organisation is generally recommended.

- **Data Item**

A data item is the smallest unit of information stored in computer file. It is a single element used to represent a fact such as an employee's name, item price, etc. In a payroll application, the employee number 170 is a data item. PANKAJ the name is a data item.

- **Field**

Data items are physically arranged as fields in a computer file. Their length may be fixed or variable. Since all individuals have 3 digit employee numbers, a 3-digit field is required to store the particular data.

Hence, it is a fixed field. In contrast, since customer's name vary considerably from one customer to another, a variable amount of space must be available to store this element. This can be called as variable field.

- **Record**

A record is a collection of related data items or fields. Each record normally corresponds to a specific unit of information. For example, various fields in the record, illustrated in Fig. 2.4 are employee number, employee's name, basic salary and house rent allowance. This is the data used to produce the payroll register report. The first record contains all the data concerning the employee PANKAJ. The second record contains all the data concerning the employee REKHA. Each subsequent record contains all the data for a given employee. It can be seen how each related item is grouped together to form a record.

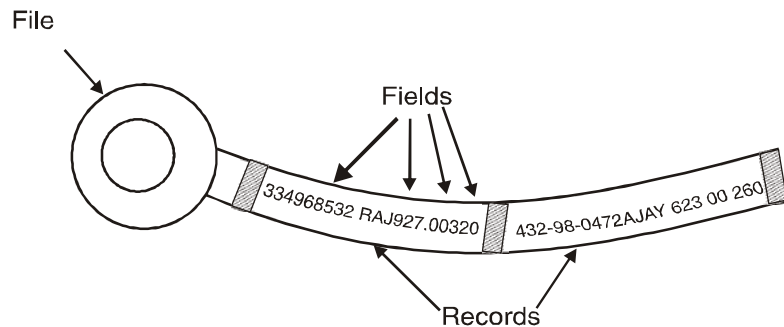


Fig. 2.4 Data Record

- **File**

The collection of records is called a file. A file contains all the related records for an application. Therefore, the payroll file shown in Fig. 2.5 contains all records required to produce the payroll register report. Files are stored on some medium, such as floppy disk, magnetic tape or magnetic disk.

- **Database**

The collection of related files is called a database. A database contains all the related files for a particular application.

2.11 Variable and Fixed Length Records

Records can be of fixed or variable length as depicted in Fig. 2.5

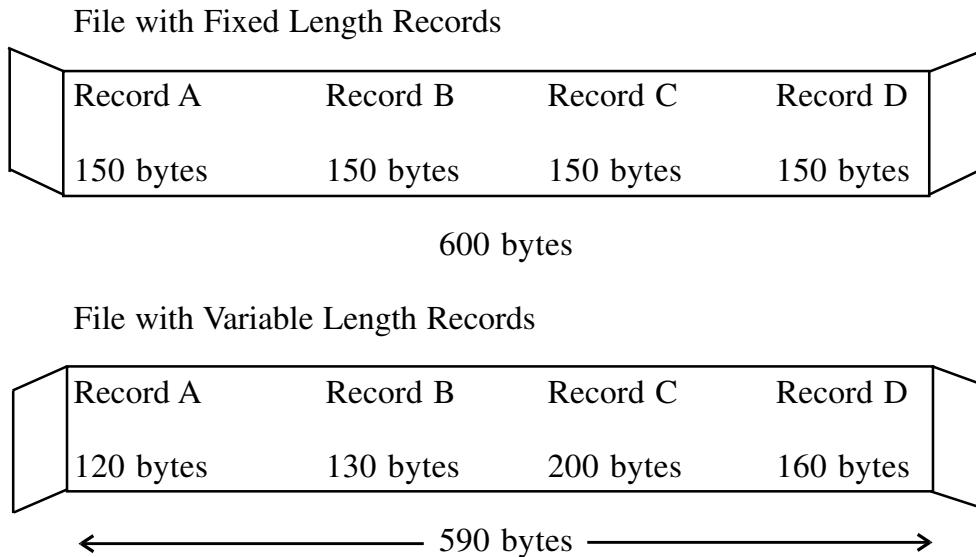


Fig. 2.5 Fixed and variable length records.

- **Fixed Length Records**

In this case, all the records in a file have the same number of bytes. Such a file is called a flat file. If all the records are expected to contain essentially the same quantity of data, then fixed length records are used.

- **Variable Length Records**

In this case, records vary in length. Use of variable length records conserves storage space when the quantity of information, of various records in a file, differs significantly.

2.12 Logical Versus Physical Record

A logical record contains all the data related to a single entity. It may be a payroll record for an employee or a record of marks secured by a student in a particular examination. A physical record refers to a record whose data fields are stored physically next to one another. It is also the amount of data that is treated as a single unit by the input-output device. Portions of the same logical record may

be located in different physical records or several logical records may be located in one physical record. For example, in case of magnetic tape, number of logical records are stored in the form of a block to increase the data transfer speed and this block is referred to as a physical record as shown in Fig. 2.6

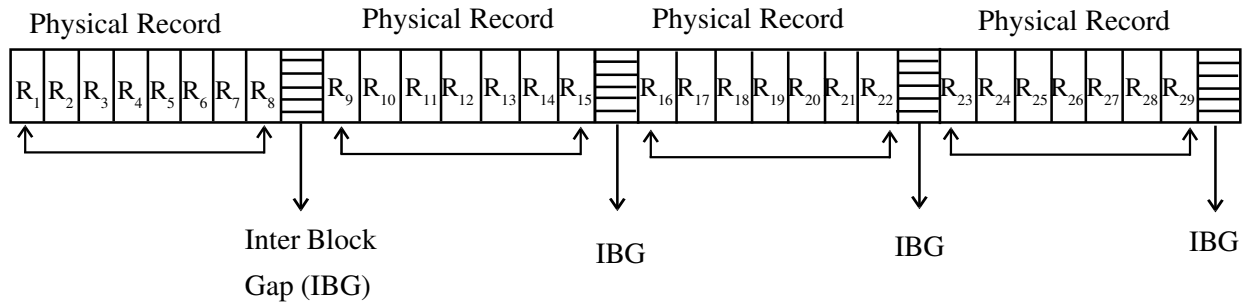


Fig. 2.6 Physical and logical Record

Intext Questions 2.2

1. Identify various data processing activities.
 2. define the various steps of data processing cycles.
-

2.13 What you have Learnt

In this lesson, we have learnt how raw data is converted into useful information. The importance of computer in carrying out the various data processing activities has been explained. Discussion about hierarchy of data is also included in this lesson.

2.14 Terminal Questions

1. Define the terms data, data processing and information.
 2. Identify various activities involved in manipulation.
 3. Write down the various computer processing operations briefly.
 4. How data is organised before processing on a computer starts? Discuss briefly.
-

2.15 Feed back to In-Text Questions

In-Text Questions 2.1

1.
 - (a) The word "data" is the plural of datum which means fact, observation, assumption or occurrence. On the other hand, information can be defined as data that has been transformed into a meaningful and useful form for specific purposes.
 - (b) Data processing is the process through which facts and figures are collected, assigned meaning, communicated to other and retained for future use. It is a series of actions or operations that converts data into useful information. In data processing system, we include the resources that used to accomplish the processing of data.
2.
 - (a) It involves the arrangement of data items in a desired sequence.
 - (b) It means to condense or reduce masses of data to a more usable and concise form.

Intext Questions 2.2

1. Data processing activities are grouped under following five basic categories.
 - (i) Collection
 - (ii) Conversion
 - (iii) Manipulation
 - (iv) Storage and retrieval
 - (v) Communication
 2. The various steps involved in data processing cycle are as follows:
 - (i) Data input
 - (ii) Data processing
 - (iii) Output
 - (iv) Storage
-

3

Computer Software

3.1 Introduction

If you want to get something done by a persons, you will tell him what to do in a language that he understands. Similarly, if you want to get some work done by the computer, you have to tell the computer in a language that the computer understands, i.e., machine language. The machine language consists of only binary digits, i.e. 0 and 1. It was felt quite difficult and tedious for human beings to thinks in binary numbers. For communicating with the computer, it was thought that it is advisable to develop a third language, a programming language, that can be understood by both human beings and the computer. Thus a programming language is a set of rules that provides a way of instructing the computer to perform certain operations.

Programming languages are said to be lower or higher, depending on whether they are closer to the language the computer itself uses (lower, which means 0s and 1s) or to the language that people use (higher, which means more English like).

3.2 Objectives

After going through this lesson you will be in a position to:

- identify various levels (or generations) of language such as machine language, assembly language, high-level language
-

- explain the concept of software
- distinguish between system software and application software
- distinguish between compiler and interpreter
- define operating system and its various functions

3.3 Computer Language

The languages in which programs are written are called programming languages. These languages can be classified into following categories.

- Machine language
- Assembly language
- High level language

Machine Languages

We think of computers as being quite complicated, but actually their basis is very simple. It is based on the concept of electricity being turned "on" and "off". From this on/off, yes/no, two-stage system, sophisticated ways of representing data have been constructed using the binary system of numbers. The binary system is based on two digits 0 and 1.

By contrast, the decimal system that we all use is based on ten digits 0 through 9. The numbers 1, 2 and 3 in the decimal system are represented in the binary system as 1, 10 and 11 respectively. Letters of the alphabet are also represented in binary numbers. In one system, the letter A is represented as 1000001. Commas, semicolons and other special characters are also represented as bunches of 0s and 1s.

In the early days of computers, with machines as the ENIAC, which uses vacuum tubes, one could actually see the tubes lit up or unlit, corresponding to the 1/0 binary state—the switch was either on or off.

In addition, in those early days there was no such things as software. There was only hardware with electrical on/off switches. Whenever a program was to be

run, all the switches had to be set - sometimes as many as 6,000 switches for a single program. Then for the next program, the switches had to be reset, a process that might take weeks.

Since those days, machine switches have been replaced by machine programming, programs with statements consisting of 0s and 1s that electrically set the switches, with 0 representing off and 1 responding on. This has made changing from one program to another considerably easier.

Still, programming directly in machine language - the lowest level of programming language, in which instruction is represented as 0s and 1s - is very tedious and time consuming. A programmer must keep track of a tremendous amount of detail. Moreover, he/she must understand the technical operations of the computer. For example, consider a hypothetical line from a program segment, which multiplies two numbers.

```
11110010 01110011 11010010 00010000 01110000 00101011
```

Clearly, working with this kind of code is not for everybody.

Programming in machine code has one advantage over programming at other language levels - its execution is very fast and efficient because the computer can accept the machine code as it is. However, in addition to the complexity involved in working at this level, there is a severe disadvantage to machine language - there is no one standard machine language. Machine language differs from machine to machine.

The languages are machine-dependent, and the programs written in machine language for one computer model will not, in all likelihood, run on a computer of different model. Although many programs or subroutines in machine language for a particular computer is supplied by the manufacturer, still few applications programs are written by users in machine language.

Assembly Languages

From the above discussion it was felt that working with 0s and 1s could turn people themselves into ciphers. In the 1950s, to reduce programming complexity and provide some standardization, assembly languages were developed. Assembly languages, also known as symbolic languages use

abbreviations or mnemonic code - codes more easily memorized to replace the 0s and 1s of machine languages.

The hypothetical machine language segment we saw above is as follows:

```
11110010 01110011 11010010 00010000 01110000 00101011
```

This could be expressed in assembly language statement as :

```
PACK 210 (8, 13), 02B (4,7)
```

Actually, assembly languages do not replace machine languages. In fact, for an assembly language program to be executed, it must be converted to machine code. The assembly language program is referred to as a 'source program' whereas, the machine language program is an 'object program'.

Assembly language code is very similar in form to machine language code. In fact, assembly languages had a one-to-one correspondence which means that 15 assembly statements, would be translated into 15 machine language statements. This one-to-one correspondence was still laborious. However, assembly language instructions (called macro instructions) were devised, which executed batches of one-to-one instructions.

Assembly languages offer several advantages:

- They are more standardized and easier to use than machine languages.
- They operate very efficiently, although not as efficient as the machine languages.
- They are easier to debug.

However, there are still some disadvantages:

- Assembly language programs are usually very long and difficult to debug.
 - Though less abstract than machine languages, assembly language programs are still complex.
 - Though more standardized than machine languages, assembly languages are still machine dependent.
-

High Level Languages

High Level Languages helped programmers by reducing further the number of computer operations details they had to specify, so that they could concentrate more on the logic needed to solve the problem.

The following example gives you a code segment in which the same instruction - " Calculate gross pay" - is expressed in three different languages: machine, assembly and COBOL.

Machine Language:

```
11110010 011100011 1101 001000010000 0111 000000101011
11110010 011100011 1101 000100001000 0111 000000101111
11111100 01010010 1101 001000010010 1101 001000011101
11110000 01000101 1101 001000010011 0000 000000111110
11110011 01000011 0111 000001010000 1101 001000010100
10010110 11110000 0111 000001010100
```

Assembly Language

```
PACK 210 (8, 13), 02B (4,7)
PACK 218 (8, 13), 02F (4,7)
MP 212 (G, 13), 21D (3, 13)
SRP 213 (5, 13), 03E (0), 5
UNPK 050 (5, 7), 214 (4, 13)
IO 054 (7), X 'FO'
```

COBOL

```
MULTIPLY HOURS -WORKED BY PAY-RATE
GIVING GROSS -PAY ROUNDED
```

3.4 Type of High-Level Languages

Languages are often referred to as generations, the idea being that machine languages were the first generation and assembly languages were the second generation. High-level languages are sometimes used to refer all languages above the assembly level. Here we will subdivide high-level languages into three generation.

- Procedural-oriented or third generation
- Problem-oriented or fourth generation
- Natural or fifth generation

Procedural-oriented Languages

High-level languages are often classified according to whether they solve general problems or specific problems. General-purpose programming languages are called procedural languages or third generation languages. They are languages such as Pascal, BASIC, COBOL, and FORTRAN, which are designed to express the logic, the procedure, of a problem. Because of their flexibility, procedural languages are able to solve a variety of problems.

Procedural languages have many advantages over machine and assembly languages:

- The program statements resemble English and hence are easier to work with.
- Because of their English-like nature, less time is required to develop a program for a given problem.
- Once coded, programs are easier to understand and modify.
- The procedural languages are machine-independent.

However, procedure-oriented languages still have some disadvantages compared to machine and assembly languages:

- Programs are executed more slowly.
 - The languages use computer resources less efficiently.
-

Problem-oriented Languages and Application Generators

Third-generation languages, such as BASIC or Pascal, require you to instruct the computer in step-by-step fashion. Fourth-generation languages, also known as problem-oriented languages, are high-level languages designed to solve specific problems or develop specific applications by enabling you to describe what you want rather than step-by-step procedures for getting there.

Fourth- generation languages may be categorized into several kinds of application development tools:

- Personal computer applications software
- Query languages and report generators
- Decision support systems and financial planning languages.
- Application generators

Personal Computer applications software

You may be knowing about various application softwares for PCs, but the ones we are particularly concerned with here are word processors, spreadsheets, database management, business graphics and integrated packages. Knowledge of spread sheet, Foxpro or Power Point can help you develop your own applications.

Query languages and report generators

Query languages allow people who are not programmers to search a database using certain selection commands. Query languages, for example, are used by airline or railway reservation personnel needing ticket information. Report generators are designed for people needing to prepare reports easily and quickly. Examples of query languages and report generators include QBE, SQL, RPG, etc.

Decision support systems and financial planning languages

Decision support systems are interactive software designed to help managers make decisions. Financial planning languages are particular kinds of decision support systems that are employed for mathematical, statistical and forecasting modelling. Both types of languages find applications in developing complicated business model hypothetical representations of management problems.

Application generators

An application generator consists of a software system with number of program modules, preprogrammed for various functions, so that the programmer or user can simply state which function is needed for a particular application, and the system will select the appropriate module and run a program to meet the user's needs.

Table 3.1 summarizes some of the major difference between third-generation languages (3GLs) and fourth-generation languages (4GLs.).

Table 3.1

Difference between 3 GLs and 4 GLs

Third-generation Languages	Fourth-generation Languages
Intended for use by professional programmers.	May be used by a non-programming personnel as well as a professional programmer.
Requires specification of how to perform task.	Requires specification of what task is to be performed (system determines how to perform the task).
All alternatives must be specified.	Default alternatives are built in; an end user need not specify these alternatives
Require large number of procedural instructions.	Require fewer instructions.
Code may be difficult to read, understand and maintain.	Code is easy to understand and maintain because of English-like commands.
Language developed for batch operations.	Language developed primarily for on-line use.
Can be difficult to learn.	Easy to learn.
Difficult to debug.	Easy to debug.
Typically file-oriented.	Typically database-oriented.

Natural Languages

Natural languages are still in the developmental stages, but they promise to have profound effect, particularly in the areas of artificial intelligence and expert systems. Natural languages have two characteristics:

- They are designed to make the connections that humans have with computers more natural - more humanlike.
- They are designed to allow the computer to become "smarter" - to actually simulate the learning process by remembering and improving upon earlier information.

Two popular natural languages are LISP and PROLOG.

3.5 Compilers and Interpreters

For a high-level language to work on the computer it must be translated into machine language. There are two kinds of translators - compilers and interpreters. High-level languages are called either compiled languages or interpreted languages.

In a compiled language, a translation program is run to convert the high-level language program, which is called the source code, into a machine language code. This translation process is called compilation.

The machine language code is called the object code and can be saved and either run (executed) immediately or later. Some of the most widely used compiled languages are COBOL, C, C ++, FORTRAN, etc.

In an interpreted language, a translation program converts each program statement into machine code just before the program statement is to be executed. Translation and execution occur immediately, one after another, one statement at a time.

Unlike the compiled language, no object code is stored and there is no compilation. The most frequently used interpreted language is BASIC.

Compiled languages are better than interpreted languages as they can be executed faster and more efficiently once the object code has been obtained. On the other

hand, interpreted languages do not need to create object code and so are usually easier to develop and test.

Intext Questions 3.1

1. What is machine language?
 2. Describe briefly the advantages of high level language.
 3. Write True or False for the following statement:
 - (a) Machine language is machine-dependent.
 - (b) Assembly language is second generation language.
 - (c) A program written in a high level language is called a object program.
 - (d) Portability is one of the features of high level language.
 - (e) Query languages allow people who are not programmers to search a database using certain selection commands.
-

3.6 What is Software

As we know what computer is a machine which cannot do anything without instructions from the user. In order to do any specific job you have to give a sequence of instructions to the computer. This set of instructions in a proper sequence is called a computer program. Software refers to the set of computer programs that cause the hardware (computer system) to function in the desired manner. Hardware means physical components of the computer system. We can say that hardware which cannot perform any calculation, comparison or manipulation without being instructed to do so. These instructions play a vital role in the performance of a computer. A complete set of instructions written to solve a problem on a computer is called software.

3.7 Types of Software

Computer software is normally classified into two broad categories:

- (i) System Software
- (ii) Application Software

3.7.1 System Software

System software includes general programs written for a computer. It consists of pre-written programs and documentation supplied by the manufacturer along with the computer. These programs are held permanently in the machine. The primary objectives of this software are to:

- (i) enhance the efficiency of hardware utilization, and
- (ii) make computers simple to use.

The first objective is realized by that component of system software which is commonly known as operating system.

System Software is a set of instruction to the machine to interpret and execute application software, for example, language translators (called compilers and interpreters), operating systems, utilities and special purpose software.

Language Translator

A language translator is a system software which translates a computer program written by a user into a machine understandable form.

The most elemental form of programming uses only the binary digits 0 and 1, which is directly understood by the electronic circuits. A program that have only binary digits is called a machine language program. It is difficult to write or understand machine language programs, as these consist of 0s and 1s.

Assembly language provides a significant improvement over machine language, Assembly language programs are written using mnemonic codes like ADD, STORE, etc., rather than their machine language representations in binary digits. Therefore, programming in assembly language is easier. For its execution, it needs to be translated into machine language code. This translation is done by an assembler. Both machine language and assembly language programs are machine dependant.

A program written in a high level language needs to be translated into machine language code before execution. This translation is done either through a compiler or through an interpreter. A compiler is a translator program which reads an entire program written in a high level language and converts it into machine language code.

An interpreter on the other hand, is a translator which converts one statement of the program into machine code, execute it and then goes on to perform the same for the next statement, and it continues doing so till the end of the program or on error occurrence.

Operating System

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

The first operating system, called batch processing operating system was developed for the second generation computers. This operating system executes jobs serially one after another from a batch of jobs submitted for execution. The central processing unit is kept busy only during the processing cycle of a job and remains idle during the input and output operations. A multi programming opening system handle multiple jobs simultaneously by overlapping the input/output and processing cycles of various jobs.

Other types of operating system which are popular today are 'multi-processing operating systems' and 'real time operating systems'. A multi-processing operating system uses multiple CPUs to process multiple jobs. A real time operating system is a type of interactive operating system which strict time limitation. Receiving and processing data is done quickly enough to produce output, to control, direct or effect the outcome of ongoing activity. The reservation system used by railway, airlines, hotel are examples of real time applications.

The function of an operating system can be compared to the functions of a principal in a school. The principal of the school provides an environment in

which the other employees can do useful work. The principal will do this by allocating resources available such as allocating time for each period, allocating classes to teachers and providing rooms for different classes etc. Similarly, an operating system governs the working of the computer and input/output devices. The operating system programs act as an interface between the user's programs and the computer's components and help in the execution of user's programs. The major functions of an operating system are:

- (i) User identification and keeping of the resources used by the users. Thus un-authorized users cannot use the computer.
- (ii) Sharing of computer resources among many users. The sharing is achieved by permitting simultaneous executions of more than one user program. This is usually called multi-programming. A mix of programs can keep the whole memory occupied, all devices active, and the control unit and ALU constantly busy, thus increasing utilization of hardware.
- (iii) Executive batches of programs, one after another, without human intervention.
- (iv) Protection of user's data and programs.
- (v) Controlling the transfer of data and programs between the main memory and secondary storage and other I/O devices.
- (vi) Providing programs to select appropriate translators.
- (vii) Providing facilities to detect and correct errors in a user's program.

An operating system understands a fixed set of commands. This set of commands is often called job control language (JCL). The JCL commands are used by the computer users to indicate their requirements to the operating system. The operating system which is used with a micro-computer is called CP/M (control program for microprocessor). Another operating system which is gaining popularity and which is available on a variety of different machines is UNIX (UNIX was developed by Bell laboratories). DOS is an operating system commonly used in PCs.

The second objective of simplifying computer usage is achieved by enabling the users to write their own programs in languages other than the machine language.

The only language understood by computers is machine language. If any other language is used, the programs must first be translated into the machine language before they can be executed. One of the most significant aspect of computers is that such translations can be made automatically using other programs. A computer manufacturer provides many programs for the translation of many languages into the machine language. These programs are called translators or compilers and form part of the system software.

We rarely talk about computer hardware alone. It is the hardware and software both which make up a computer system. There is definitely some substance in the argument that system software is as important, if not more, as its hardware.

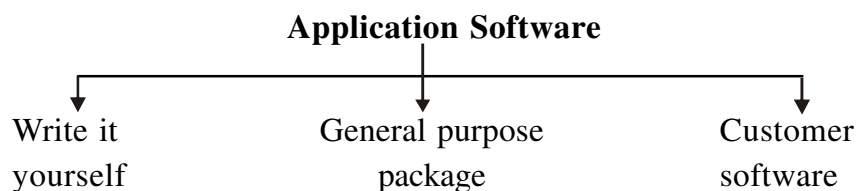
Utility Software

Utility software may be considered as a system software which is used quite often in the development of a program. Sort merge programs are used to sort records in a particular manner or sequence. Such programs are normally provided by the manufacturers. The programmer can also develop his own utility software and keep it in the secondary memory of the computer.

3.7.2 Application Software

Application software is written to perform a specific task or process, such as accounting, payroll, mailing list, result preparation and so on. The majority of application software is written in high-level languages.

Assuming that the task to be carried out has been correctly identified, carefully defined, the prospective user will come across the following alternative sources for this application software.



- (a) **Write it yourself** : The program written by the user can be the most satisfactory solution. It will be an exact match to the needs of the business. The program can grow with the business.
-

- (b) **General purpose application packages** : Application packages refer to a set of computer programs, which have been written to perform specific, commonly required tasks. Each program is written in such a way that it is applicable to a large number of users. The main advantage is that it is relatively cheap as cost of the package is spread over a number of customers. The major disadvantage of application package is that it is not likely to fulfil all the requirements of the prospective users.
- (c) **Customer software** : It refers to computer programs specially written to match the exact needs of the user. It is precisely the same as getting one's clothes stitched from a tailor to fit exactly rather than buying a ready-made dress. The most important advantage is that such software fulfils all the needs of the customer. The major disadvantage is that customer software costs much more than general purpose application software, because the package is specially made for one particular customer.

Common Application Packages

Some of the common requirements of the users of personal computers have been identified and common applications packages have been developed to meet their needs. These packages include word processor, database processing, spreadsheet calculations, mail-merge, presentations and communications (email). These packages have been prepared so that they are simple to use. They also provide graphical user interface to make them very user friendly. These packages are readily available in the market and one can purchase them, install it on his/her computer easily and start using the package.

Intext Questions 3.2

1. Define software.
 2. Differentiate between system software and application software.
 3. Write True or False for the following statements.
 - (a) A program written in a high level language needs to be translated into machine language code before execution.
-

- (b) An operating system is a must to operate a computer system.
 - (c) JCL stands for Job Common Language.
 - (d) Application software is written to perform a specific task or process.
 - (e) Utility software cannot be considered as an application software.
-

3.8 What you have learnt

In this lesson, you learnt about various levels of language and their advantages and disadvantages briefly. Importance of two types of software, namely, system software and application software has been explained in a simplified manner so that it could be easily understood by you.

3.9 Terminal Questions

1. Differentiate between hardware and software?
2. What are the advantages and disadvantages of machine language?
3. What is the difference between source program and object program?
4. Explain assembly language. What are its advantages over machine language?
5. Define operating system. Write down its various functions.
6. Describe briefly about application software.

3.10 Feedback to In-Text Questions

In-Text Questions 3.1

1. Machine language consists of sequence of instructions written in the form of binary numbers (1s and 0s) to which the computer responds directly. Computer understands this language only.
 2. Advantages of high level language are:
-

- (a) Readability
 - (b) Easy learning
 - (c) Easy debugging
 - (d) Easy software development
3. (a) True (b) True (c) False (d) True (e) True

In-Text Questions 3.2

1. Software refers to the set of computer programs that cause the hardware to function in the desired manner. Software is a general term that is used to describe any single program or a group of programs.
 2. Application software is a set of programs to carry out operations for a specific application. System software is a set of programs written for performing tasks such as controlling all operations required to move data into or out of the computer.
 3. (a) True (b) True (c) False (d) True (e) False
-

4

Operating System

4.1 Introduction

Most of the personal computers you see today comes with Microsoft Windows operating system. It is an operating system meant for IBM compatible PCs. Operating system is the foundation software that makes it possible for you to use your computer and have it perform the tasks you need. Windows 98 is an updated version of Windows 95.

4.2 Objectives

After going through this lesson, you will be in a position to

- explain various features of Windows 98
- learn various Icons and their features
- define file management features of Windows 98

4.3 Main features of Windows 98

The following are the main features of Windows 98 operating system.

- Creates the link between you, the user, and the computer by providing an
-

interface in which you can communicate with the computer

- Serves as the base software on which a variety of programs can operate. Examples include MS Word, MS Excel, MS Power Point, and MS Access
- Handles internal functions, such as managing the computer memory, input and output operations
- Provides a series of utilities - specialized programs to manage your system - and commands that you can use to manage your files, folders, and disks
- A Graphical User Interface (GUI), which uses pictures, symbols, windows, and words on your screen that you control with the mouse
- Built-in programs, or accessories like, WordPad, Notepad, Calculator, Paint program etc.
- The ability to display several documents and run a number of programs in their own windows, all at the same time.
- The World Wide Web (WWW) is easily accessible from anywhere in windows. You can communicate over the Internet. Using Internet tools included with windows, you can send e-mail, chat, and view news.
- Supports DVD and digital audio, so that you can play high quality digital movies and audio on your computer. You can also watch TV.

4.3.1 Using the Mouse

Windows98 is designed for use with a mouse. Although, you can use the keyboard for most actions in Windows 98, many of these actions are easier to do with a mouse. You will use four basic mouse actions throughout this course.

1. **Pointing** : Moving the mouse to place the pointer on an item is called pointing.
 2. **Clicking** : Pointing to an item on your screen and then quickly pressing and releasing the left mouse button is called clicking. You select items on the screen by clicking. Occasionally there are operations you perform
-

by clicking the right mouse button, but unless you are instructed, click the left mouse button.

3. **Double-clicking** : Pointing to an item and then quickly pressing and releasing the mouse button twice is called double-clicking. Whenever you are unsure of the command to use for an operation, try clicking the item you want to affect. If you get no result, try double-clicking. This often displays a dialog box in which you can make changes to the item you double-clicked.
4. **Dragging** : Pointing to an item and then holding down the mouse button as you move the mouse is called dragging. You can use this action to select data, and to move and copy text or objects.

4.4 Symbol for Menu Commands

As a shortcut, we have used a special convention to indicate menu commands throughout this course. When you want to choose a command from the menu bar, it will follow this pattern: menu name - command. For example, "Choose File → Open" is a shorter way of saying "Choose the File command from the menu bar, then choose the Open command from the File menu". Sometimes you will see a sequence of menu selections that goes three or more levels.

4.4.1 Desktop

When you turn on your computer, the screen first displays Basic Input Output Information (BIOS) about your computer. Few seconds later, Windows takes control and the Windows logo flashes on your screen with the cloudy sky. You could see "Microsoft Windows 98" . The first thing that always appears after loading is that you will be asked to type your password to enter Windows (it is the password you typed the first time Windows was launched). The Desktop is where you might place useful things always accessible to you. Depending on which options you choose during installations, you will see some or all the objects as shown in Figure 4.1

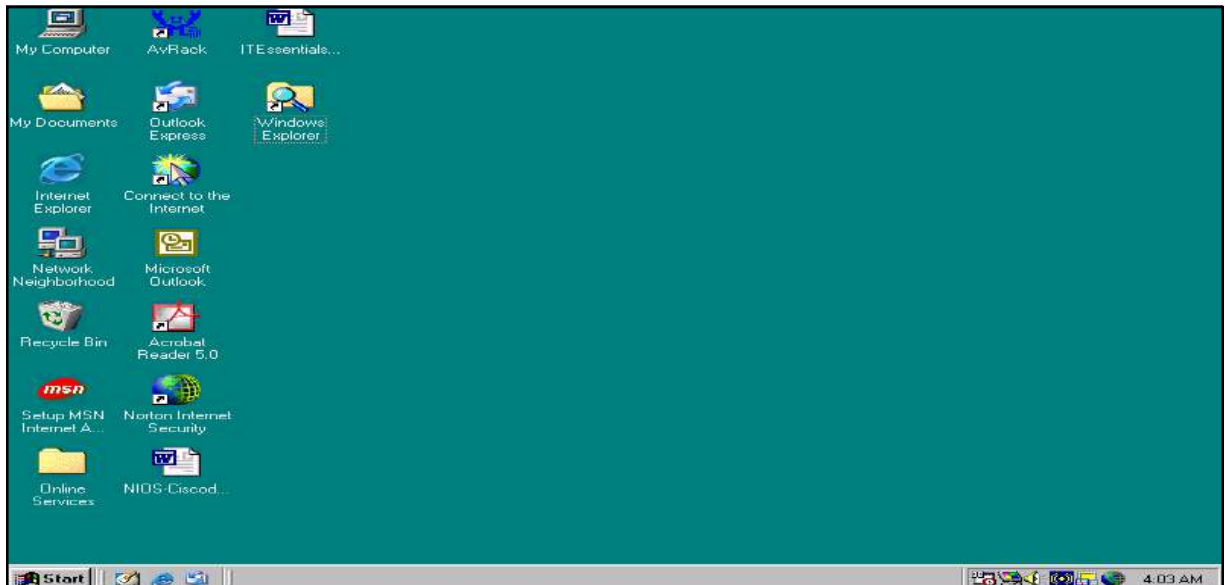





Fig. 4.1 The Windows Desktop

4.4.2 Desktop Icons

1. **My Computer** : My Computer is the most important Desktop item (see fig. 4.1). This is where everything is present: your hard drive, floppy, CDROM, computer's main control (Control Panel). All the computer resources can be accessed from My Computer - it is the gateway to your computer. To access My Computer, double-click on its icon in the Desktop.

My computer contains:

- (i) Hard Disk Drive(s) (C:D:)
- (ii) 3.5 Floppy Drive (A:)
CD-Rom Drive (E:)
- (iii) Printers
- (iv) Control Panel
- (v) Dial-up Networking (used especially for Internet Connection)
- (vi) Scheduled Tasks (give your computer jobs to perform when you are out)

My Computer window (see fig. 4.2) looks just like any window you open, you can minimize, this Window with , maximize with  and close by clicking on .

In My Computer you are forbidden to Delete, Copy, Rename anything, try to press delete key and nothing will happen. The items in My Computer window are so crucial that you cannot run your computer without them. For example, if you remove the hard drive where would you store your files? Since My Computer items are related to the computer system they should remain intact.

Suppose you want to use My Computer to view your hard disk. On the Desktop double-click My Computer. The My Computer window (see Figure 4.2) with different icons will appear.

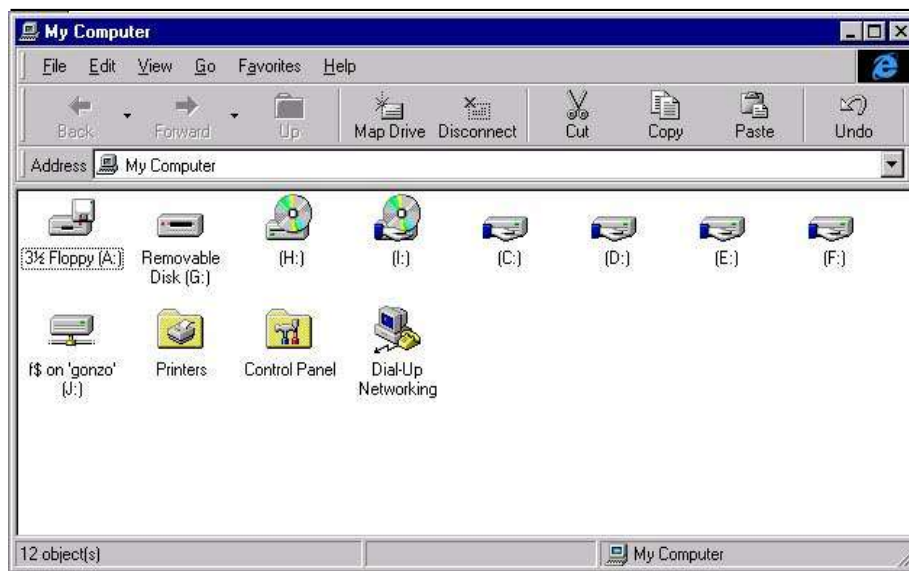


Fig. 4.2 My Computer

Double-click the icon (see fig. 4.2) that represents your hard disk. Your hard disk window appears, and you can easily see the contents of your hard disk.

Similarly you can double-click the other icons in My Computer.

To view the contents of the floppy disk, which is usually designated drive A:

To view the contents of the hard disk, which is usually designated drive C:

To view the contents of the hard disk, which is usually designated drive D:

To view the contents of compact disk in the CD-ROM drive designated as E:

To set up printer and view information about available printers and print job status.

To view tools you can use to modify your computer settings.

To schedule or view tasks for computer maintenance

To view web folder contents

To view shared information on another computer by using a modem.

The Properties Window in My Computer

To learn more about each item, just choose one icon, right-click and select Properties from menu. The Properties window is divided into four tabs: General, Tools, Sharing, and Compression. If you select the Floppy icon, the General tab shows information about your floppy, storage device. It tells you how much space you still have on your disc, and the total size in KB or MB. You should have a total of 1.45,664kb if you are using a 1.44 MB floppy. At the top of the window is the floppy table, usually it is blank. The pie chart shows clearly the Used Space and Free Space (see Figure 4.3)

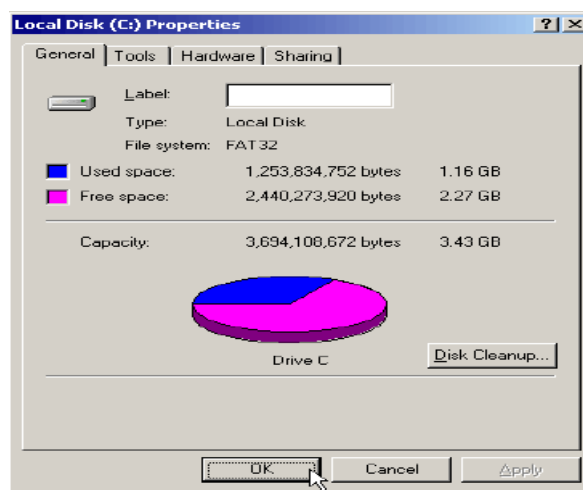


Fig. 4.3 The Properties Window

The Tools tab in fig. 4.3 features the following three important Windows utilities:

Error-checking status: It is a utility that let you check your computer for errors. If you accidentally turn your computer off without shutting down, it will run when you restart your computer and perform a scan on your hard drive.

Backup status: This utility lets you backup important stuff so you will never loose anything in case your computer was damaged. You should always keep two copies of your work you store in computer.

Defragmentation status : Use this utility to make your computer run faster. The more you use the computer the more " it gets tired and old". Defragmentation will refresh it.

The Sharing tab will tell you whether your hard disk is shared by other network users or not.

The Compression tab is almost never used. Suppose one day you run out of disk space, you can use the Compression utility to squeeze your files and generate more room.

Beside My Computer there are quite a few more icon on the Desktop. These are as follows :

2. **My Documents :** This is where you store all your personal stuff like files, anything you create with your computer. You can put your things somewhere else also. Think of My Documents as your Desktop drawer where you can put your papers.
 3. **Recycle Bin :** Provides space for deleted files from folders or Desktop, and provides a second chance to recover files deleted.
 4. **Internet Explorer :** It is a piece of software called a browser that lets you explore the Internet. You might have another icon like "Internet Connection" that will help you configure your computer to access the Internet.
 5. **Network Neighbourhood :** Provides access to shared resources on your computer and other computers on your local and wide area networks.
-

If your computer is connected to a network, you can use Network Neighbourhood to browse network resources in the same way you browse the contents of your computer.

To open network resources follow the steps given below :

- i. On the Desktop, double-click **Network Neighbourhood**. The computers in your workgroup will appear. If you want to view all of the resources available on the network, double-click **Entire Network**.
 - ii. Double-click the servers and folders you want to open.
6. **Online Services** : As online services is a subscription computer service you can use it to access a wide variety of information and exchange messages with other users. With online services, you can obtain informations on news, sports, weather, the stocks market, and many more. You can get access to different libraries and databases. You can send mail to other users of the online service. Examples of online services include America Online, CompuServe, Prodigy, and MSN.
 7. **Date, Time, Volume** : At the bottom right corner of the screen, you can tell what time is it . By clicking you mouse over the clock icon, you will get the date. You can also control the sound by clicking on the speaker icon.
 8. **Start Button** : This button lies at the bottom left of our screen. It provides an easy way to start different programs and other applications.
 9. **Taskbar** : This lies at the bottom of your screen. This displays all open applications and windows. The Start button and the Taskbar work together and help you in managing your applications opener close windows easier.

You will learn more about these Desktop icons in the next few sections of this lesson.

Intext Questions 4.1

1. Write any five main features of Windows 98.
 2. Explain the following terms used in application of mouse : (a) Pointing, (b) Clicking, (c) Double-clicking, (d) Dragging
 3. What is Desktop? What are the main items you see on a Desktop?
 4. What are the main components of My Computer?
-

4.5 Start Button and Taskbar

At the bottom left corner is the Start button. Place the mouse over Start and you will see "click here to begin". Start gives you access to all applications on your computer, every time you install a software on your system, you can easily have access through the Start button. Click on Start and the Start menu appears. Figure 4.4 shows the main menu items available when you click on the Start button.

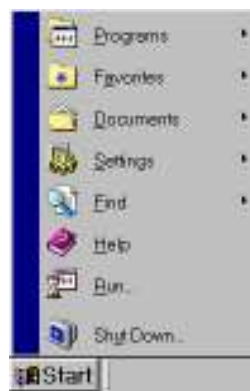


Fig. 4.4

4.5.1 Programs Submenu

Every program on your computer can be accessed from Programs submenu, you can also access the Computer Accessories like System Tools, Communication and Entertainment features. Move your mouse pointer over the programs items on the Start menu, and another menu will pop up. In fig 4.5 we clicked on programs and then on accessories. In the next column we clicked on games and then on Hearts.

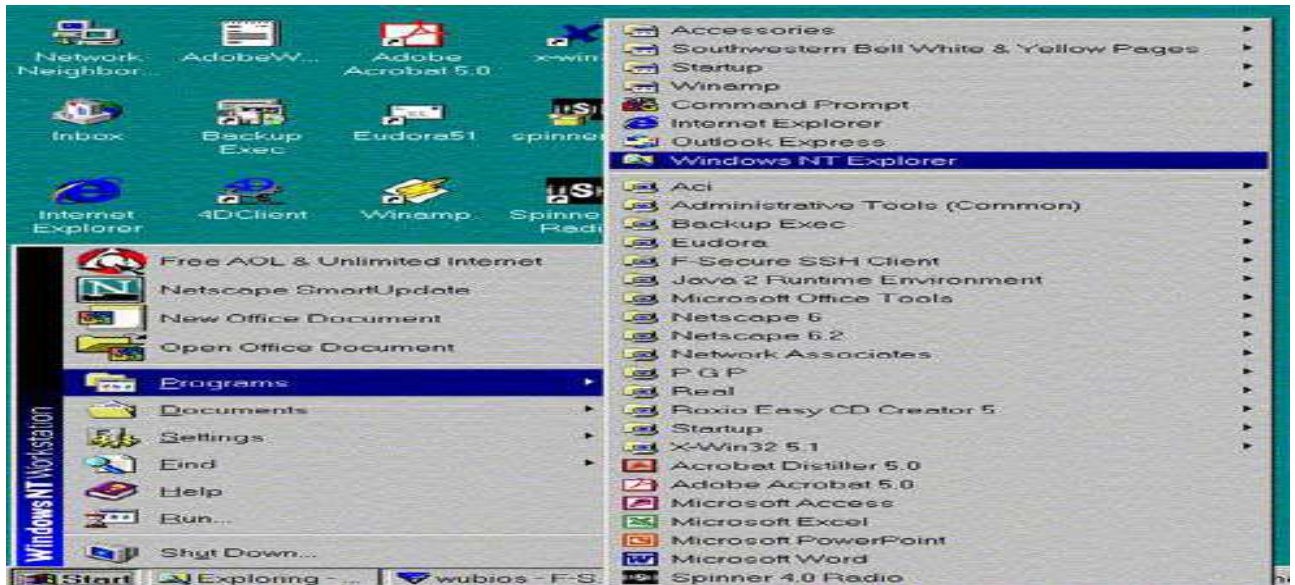


Fig. 4.5 Programs Submenus

When you open programs and files on top of other programs and files on the Desktop, Windows 98 keeps track of all your open windows and lists them on the Taskbar. The advantage of the Taskbar is that when you have multiple applications open at the same time, you can see all of them listed in the foreground, whether or not they are hidden by another windows. Even if an application window is minimized on the Desktop, it can still be accessed from the Taskbar. By clicking on the applications window on the Taskbar, you can bring that application to the front of other windows, and restore if it was minimized. The more windows you open, the more items will be listed on the Taskbar. You can increase the height of the Taskbar for multiple lines of open items. To increase the height of the Taskbar, place the mouse pointer over its top border, the pointer will turn into double-headed sizing arrow (↕). then, click and hold your left mouse button and drag up.

4.5.2 Favorites Submenu

While surfing the internet you can store your favorite addresses in this folder so that you don't have to memorize them. This feature is very useful for Internet users. You can also quickly get to documents or folder or to other computers on your network.

4.5.3 Documents Submenu

Windows keeps track of recent documents used. The Documents option of the Start menu makes available the last 15 documents you had opened in any

application. Click on Documents in fig. 4.4. and you will see a list of documents you have been working on recently. The documents are listed alphabetically, not by use, and will be listed whether you open them from within an application or from the folder where they are stored. If you find these documents irrelevant and don't want to access, you can clear the list: Right click on the Taskbar. In a small Window you will find 'properties'. On clicking there, another Window will appear. Click on the 'Start Menu Programs'. There, at the documents menu, click on the 'Clear' button.

4.5.4 Settings

From here you have access to the Control Panel (see Figure 4.5), Printers, Taskbar and Start menu, Folder Options, Active Desktop, and Windows update. You can add/delete programs, change monitor settings, connect to networks, create a Startup Disk, Customize Desktop display properties, and set up Dialup Networking. The main features of Control Panel are :

1. You can do customizing of most of your Windows environment. You can display pictures, patterns, or even scanned photographs as your wallpaper, the background of your Desktop. Using the different tabs in the Display Properties of dialog box, you can also change items such as the icons on your Desktop, the colors of individual windows, and the size of the objects on your screen. You can even add items to your Active Desktop or set up a screen saver.
 2. You can quickly install Programs, such as MS Word or a game, using the Add/Remove Programs feature in Control Panel.
 3. You can also connect your computer to a network using Network features in Control Panel. To connect your computer to a network follow the steps given below :
 - i. Click **Start**, select **Settings**→ **Control Panel**, and then double-click Network dialog box appears. Click Add. Select **Network**. Component Type dialog box appears. Click **Client**, and then click **Add**. A list of client software appears.
 - ii. In the **Manufactures** list, click the name of the manufacture of your network software.
-

- iii. In the **Network Clients** list, select the client software you are using, and then click **OK**. The client software is added to your computer.
 - iv. On the **Configuration** tab, select your client, and then click **Properties**. Enter configuration options for your network, and then click **OK**. (If you don't know the options for your network, contact your network administrator.)
 - v. Click **OK** and then click **OK** again. The client software is installed and your computer restarts.
4. If you have problems with Setup or have trouble in starting Windows98, you can use a Startup Disk to start your computer and run Setup or gain access to your system files. If you have problems with your hard disk, for example, you can use a Startup Disk to start your computer and troubleshoot your hard disk.

To create a Startup Disk from within Windows98 follow the steps given below:

- i. Click **Start**, point to **Setting**, click **Control Panel**, and then double-click **Add/Remove Programs**. The Add/Remove Programs Properties dialog box appears.
- ii. Click the **Startup Disk** tab, and then click **Create Disk**.
- iii. Label a floppy disk "Windows 98 Startup Disk". insert it into your floppy disk drive, and then click **OK**.

To start your computer using Startup Disk follow the steps given below:

- i. Insert the Startup Disk in the floppy disk drive.
 - ii. Restart your computer. The Windows 98 Startup menu appears. Type the number of the appropriate CD-ROM option, and press **ENTER**. Follow instructions on your screen. After a series of scans, the MS-DOS prompt appears. From this prompt, you can gain access to the system files on the Startup Disk.
-

5. You can Add Printer wizard to quickly set up printers, select default printers, and change printer settings. The wizard guides you step-by-step through printer set up.

4.5.5 Find

When you are looking for a particular folder or file, you can use the Find command instead of opening numerous folders. The Find command lets you quickly search a specific drive or your entire computer or a computer on your network.

4.5.6 Help

When you are stuck, Windows can give you a hand, just go to **START**→**HELP** or press **F1** on your Keyboard.

4.5.7 Run

This is the fastest way to start a program or install a software on your computer or start programs or open folders or connect to network resources. With Run you will have instant access to everything. However, the Run option is for more experienced user. Selecting Run brings up the RUN window where you can type the name of a file or folder that you wish to open. If you don't know the entire path of folders within which a specific file or folder can be found, you can click on the Browse button within this window to find what your are looking for. Although, this is generally not the quickest way to open a document or run a program, sometimes it simply seems more direct than opening a sequence of folders to get the one you need.

4.5.8 Shut Down

You should Shut Down Windows at the end of a session, never turn the power off, close all applications and click **START**, select Shut Down and press **OK**. Many new computers automatically turn the power off once you have Shut Down. In case your computer do not have this feature turn the power off.

4.5.9 LOG Off

This feature is used only when you share your computer with others. If you are the only one using the computer you need not choose this feature.

4.5 Windows Explorer

Windows Explorer is a feature that you can use to view the contents of your computer and network drives in a hierarchical structure. Instead of opening drives and folders in separate windows, you can browse through them in a single window. The left side of the Windows Explorer window contains a list of your drives and folders, and the right side displays the contents of a selected folder. To use Windows Explorer, click on Start button and then click Programs→Windows Explorer (see fig.4.6)

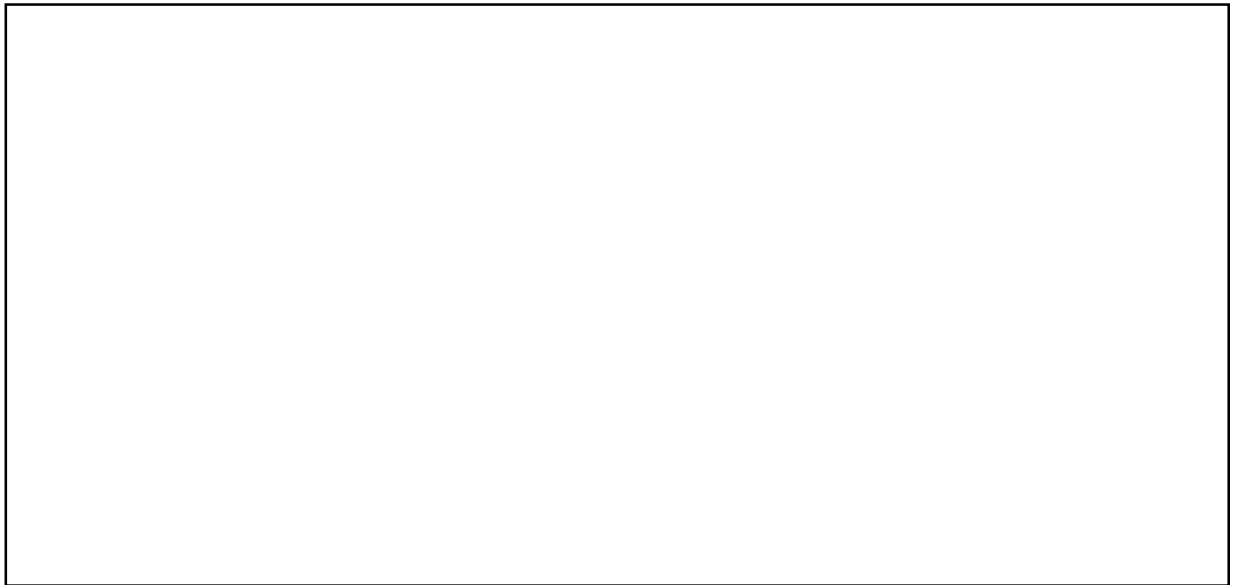


Fig 4.6 Explorer Window

Windows Explorer has the following features:

- You can copy a file or folder containing a number of files from one folder to another folder or one drive to another drive by simply placing the cursor on the source file or folder and dragging it to the destination folder or drive. You can also do this using **Copy** and **Paste**.
 - You can delete a file or folder by selecting the file or folder and then clicking on the **Delete** on the Toolbar. Windows will send it to the **Recycle Bin**.
 - You can move a file or folder from one folder or drive to another folder or drive using **Cut** and **Paste**.
 - By mistake if you have deleted, or moved or copied a file or folder and
-

want the file or folder in its original place, you can do it pointing the cursor on Undo and then clicking.

- You can open a file and its program by double clicking a file.

4.7 Managing Files, Folders and Windows

To use Windows, you need to know two things: files and folders. The following first three icons represent different folders and the next three icons represent different types of files (fig.4.7).



is one of the high.

Fig. 4.7 Files and Folders



Book1.lnk

Files always lies in folders. In Windows98, you can organize your documents and programs to suite your preferences. You can store these files in folders, and you can move, copy, rename, and even search for files and folders.

File names in Windows 98 can be up to 255 characters, including spaces. However, file name cannot contain any of the following characters: \ / : ? < > |.

4.7.1 Creating Folders

When you use a program and save your work, or when you install a program, you are creating files. You can store files in many location - on the hard disk, a network drive, a floppy disk, and so on. To better organize files, you can also store them in folders.

To create a folder follow the steps given below:

1. On the Desktop, double-click **My Computer**. My Computer Window opens.

2. Double-click disk drive or folder in which you want to create a folder. The drive or folder window opens.
3. On the **File** menu, point to New, and then click **Folder**. Type a folder name, and press **ENTER**. The new folder appears in the location you selected.

To find a file or folder follow the steps given below:

1. Click the **Start** button, point to **Find**, and then click **Files** or **Folders**. The Find dialog box appears.
2. Now type the file or folder name you want to find. Click the Look in down arrow, or click Browse to specify where to search. Click **Find now**. After a moment, the results of the search appear.

To open a file or folder follow the steps given below:

1. On the Desktop, double-click **My Computer**. The My Computer window opens. Otherwise, you can click on **Start** and Choose **Programs**→**Windows Explorer**.
2. Double click the drive that contains the file or folder you want to open.
3. Double click the file or folder you want to open.

To rename a file or folder follow the steps given below:

1. On the Desktop, double-click **My Computer**. The My Computer window opens. Otherwise, you can click on **Start** and choose **Programs**→**Windows Explorer**.
2. Double click the drive that contains the file or folder you want to rename. Select the file or folder you want to rename.
3. On the **File** menu, click **Rename**. Type a name, and then press **ENTER**.

To copy or move a file or folder from one location to another location follow the steps given below:

1. On the Desktop, double-click **My Computer**. The My Computer window opens. Or click on **Start** and **Choose** Programs→Windows Explorer.
-

2. Double click the drive that contains the file or folder you want to move. Select the file or folder you want to copy or move.
3. On the **Edit** menu, click **Copy** to copy the file, or click Cut to move the file. Double-click the folder in which you want to place the file or folder. On the **Edit** menu, click **Paste**. The file appears in its new location.

To delete a file or folder follow the steps given below:

1. On the Desktop, double-click **My Computer**. The My Computer window opens. Or click on Start and Choose **Programs→Windows Explorer**.
2. Double click the drive that contains the file or folder you want to delete. Select the file or folder you want to delete.
3. On the File menu, click **Delete**. The 'Confirm File Delete' dialog box appears. Click **Yes**. The file is deleted moved to the Recycle Bin.

To permanently delete a file or folder follow the steps given below:

1. On the Desktop, double-click **Recycle Bin**. The Recycle Bin window opens. Select the file or folder you want to permanently delete.
2. On the Toolbar click on **Delete**. The Confirm File Delete dialog box appears. Click **Yes**. The file is permanently deleted. (If you want all files and folders in Recycle Bin are to be deleted, then on the **File menu**, click **Empty Recycle Bin**. The Confirm File Delete dialog box appears. Click **Yes**. All the files are deleted from the Recycle Bin.)

4.7.2 Shortcuts

For an easy access to a file that you use frequently, you can create a shortcut to it. A shortcut does not change the location of a file - the shortcut is just a pointer that lets you open the file quickly. If you delete a shortcut, the original file is not deleted. There are a number of ways to create a shortcut to a file. The most direct route is as follows:

1. On the Desktop, double-click **My Computer**. The My Computer window opens. Or Click on **Start** and Choose **Programs→Windows Explorer**.
-

2. Double click the drive that contains the file or folder you want to create a shortcut.
3. Select the file or folder you want to create shortcut. Right-click on the file or folder. On the menu that appears, click **Create Shortcut**. The shortcut appears on the selected drive or folder. (If you want to place the shortcut on the Desktop right-click on the file or folder. On the menu that appears, click **Send To→Desktop** (created shortcut). The shortcut appears on the Desktop.)

To remove a shortcut from your Desktop, drag the shortcut to the Recycle Bin icon or use the right mouse button pointing the shortcut to be deleted and click **Delete**. To rename a shortcut, close all other open windows, click the right mouse button pointing a file or folder icon to be renamed, click **Rename** on the menubar that appears. Type the new name to replace old name and then press **ENTER**.

4.7.3 Windows Most Common Tasks

Every Folder in Windows has a layout as shown in Figure 4.6 This layout is called a window. The Menu Bar, Title Bar, Taskbar, Scroll Box, and three squares (–□×) at top right are common to all windows, whether you have a Folder or File - Open or Close. To start an application or access a file/folder you have to Open, when you finish you should Close. You have already learned about Taskbar. We will discuss the other parts of window now.

1. Minimize, Maximize/Restore, and Close

Every window has 3 squares at the top right of the screen – □ x. The minus sign is Minimize. A minimized window is still open, and a window button appears on the Taskbar. The double-square □ is Maximize/Restore. You can maximize the window of your file to fit the entire screen. If you click again the application takes the usual size (you can see the Desktop behind) if you have several files opened, “maximizing” put them on top of each other so you can see only one file at the top, by "restoring" you will have all the files visible on the desktop, just choose one and start working: you can write and draw at the same time very easily. To close anything just press on x.

2. Title Bar

The Title bar resides at the top of the screen. It displays the name of the program in which you are working. It also displays the name of the file or folder or drive you are working with. You can also use Title bar for enlarging and reducing the window by double-clicking on the Title bar. Suppose you opened two or more windows (file, folder), to see in which one you are working, look at the Title bar and the Title bar shows the active window. The advantage of Windows is that you can move the window and place it wherever you want; it is like moving a piece of paper on your actual desktop. Just click the mouse on the Title bar (the window should not be maximized; if it fits the screen you will not be able to move it) hold it and drag. To drag an item you need to click on the **Title bar**, hold it, then move the mouse. The whole process is called **Drag and Drop**. You can also move a window (if window is not Maximum) by right clicking on Title bar and selecting Move. By using the mouse you can even change the size of the window : (not maximized) take the mouse pointer to the border of the window, a double-edged arrow will appear:

↕ let you change the height, ↔ is for the Width. ↖ at the edge of the window let you resample - the ratio between Height and Width is a same.

3. The Scroll Bar

When you exceed the size of the Window, a vertical rectangle - Scroll bar - appears at your right. The same applies if you went over the Width limit, an horizontal Scroll box appears. To go through the text, just Drag the scroll bar until you reach the location, you can also click on the arrows at the edges of scroll bar to browse. A Window cannot extend beyond the size of your screen, therefore, the scroll bar is the perfect solution to extend beyond the boundaries. The size of the scroll bar (rectangle) indicates the length of the document.

4. Menu Bar

Windows 98 provides a variety of menus you can use to choose and perform different tasks. The most frequently used menu in Window 98 is Start menu, which you have already learned. In addition to Start menu, there are many windows and programs that have their own sets of menus and commands.

Intext Questions 4.2

1. What are the features of Windows Explorer?
 2. What are the steps for permanently deleting a file?
 3. What is a shortcut?
-

4.8 Sharing Folders and Printers

You can share your documents (files) and printers attached to your computer with other people on the network. To use file and print sharing, you must first choose which of the following two types of access you want to give to other users:

- a. **Share-level access control** is the default access setting. It lets you require a password for each shared resource.
- b. **User-level access control** lets you specify who has access to each shared resource, but it does not require a password.

By default you will be having a Share-level access control. To change it to User-level access control, follow the steps given below :

- i. Click **Start**, select **Settings** → **Control Panel**, and then double-click **Network**. Click the Access Control tab. The **Network** dialog box appears. Click on **User-level access control**.
- ii. **In Obtain list of users and groups from**, type the name of the domain or server you want to use, and then click **OK**. You may be prompted to supply additional information about the domain or server you specified.
- iii. Restart your computer.

To set up file-sharing and print sharing services, follow the steps given below :

- i. Click **Start**, select, **Settings** → **Control Panel**, and then double-click **Network**. Click the Access Control tab. The Network dialog box appears. Click on **File and Print Sharing**.
-

- ii. Select the check box(es) the sharing options you want, click **OK**, and then click **OK** again. A message prompts you to insert your Windows 98 CD or Setup disk so that File and Print Sharing can be installed. You must restart your computer before the new settings will take effect. After you have set up file-sharing services, you can share a folder or printer.

To share a folder with share-level access control, follow the steps given below:

- i. In my Computer, right-click the folder you want to share, and then click Sharing in the submenu. **Properties** dialog box will appear. Click on the Sharing tab, and then click **Shared As**.
- ii. In **Share Name**, type a name for the folder. In Comment, you can type a brief comment or description of the folder.
- iii. In Access Type, click Read Only, or full depends on Password. Regardless of which type of access you select, you have the option of adding a password.
- iv. Type a password if you want to use one, and then click OK. Retype the password and then click OK. The folder or printer icon changes to a folder or printer with a hand, indicating that the item is now shared.

To share a folder with User-level access control, follow the step given below :

- i. Follow the steps i. and ii. as stated above.
 - ii. Click **Add**. In the **Add Users** dialog box, click the name(s) of the person(s) to whom you want to grant permission(s). You can scroll the list of users, or you can type the name and the list will scroll automatically. If you want to grant the same permission to everyone connected to your network, leave **The World** selected.
 - iii. Click the type of access permissions you want to give to the selected user(s). **Read Only** means the user has the ability to read, but not change remove files. Full **Access** means the user has the ability to read, delete, and change file. **Custom** means the user has a combination of privileges that you specify.
 - iv. When you are finished adding users permission, click **OK**.
-

To share a printer with share-level access control, follow the steps given below:

- i. In **My Computer**, double-click the **Printers** folder. Right-click the printer you want to share, and then click **Sharing**. On Sharing tab, click **Shared As**.
- ii. In Share Name, type a name for the printer. In Comment, you can type a brief comment or a description of the printer.
- iii. In **Password**, you can type a password, retype the password to verify it. Click **OK**.
- iv. Click **OK**. If you typed a password, retype the password to verify it. Click **OK**.

To share a printer with User-level access control, follow the steps given below:

- i. Follow the steps i. And ii. as stated above.
- ii. Click **Add**. In the **Add Users** dialog box, click the name(s) of the person(s) to whom you want to grant permissions. You can scroll the list of users, or you can simply type the name and the list will scroll automatically. If you want to grant the same permissions to everyone connected to your network, leave **The World** selected.
- iii. Click **Full Access**. The name(s) of the person(s) you have chosen will be moved to the box on the right side of the screen. When you are finished adding users and permissions, click **OK**.

If you have correct permissions, you can use files and printers that are shared on other computers. To use a shared folder or printer, follow the steps given below:

- i. Double-click **Network Neighborhood**
 - ii. Double-click the icon for the computer that controls the folder or printer you want to use. If the computer you are looking for is not in your workgroup or domain, first click **Entire Network**, and then double-click the appropriate workgroup or domain.
 - iii. To use a shared folder or printer, double-click its icon.
-

4.9 MS-DOS-Based Program

You can run MS-DOS-based programs by using the MS-DOS Prompt command in the Programs menu. Click on **Start** and then choose **Programs → MS-DOS Prompt**. This opens an MS-DOS command window in which you can run your MS-DOS-based programs.

Using **MS-DOS Commands** : We have given some examples below to know how to use MS-DOS based commands:

1. Be sure that the MS-DOS window is active.
2. At the MS-DOS prompt (C:\), type DIR and press Enter. The files and folders stored in your open folder on the hard disk appear in a list. The DIR command means “directory” and lists all files and folders in your current directory (folder).
3. At the next MS-DOS prompt, type “CD\Windows” and then press ENTER. The folder changes to the Windows folder in the same drive (if it exists). The CD command means “change directory”. The backslash (\) and Windows indicates the directory path for your Windows folder.

The following Table gives a list of MS-DOS commands.

Table 4.1
List of DOS Commands

command /?	List switches for the DOS command "command"
path=c:\windows;c:\dos	specify in which directories DOS searches for commands or programs
prompt \$p\$g	make the DOS prompt display the current directory
dir	list files in the current directory in one column
dir /w	list files in five columns
dir /p	list files one page at a time

<code>dir *.exe</code>	list all files with an "EXE" extension
<code>dir z???.exe</code>	list "EXE" files that have four letters and start with z
<code>dir winsock.dll/s</code>	searches for the file "winsock.dll" in the current directory
<code>type file.ext</code>	view the contents of the text file "file.ext"
<code>edit file.ext</code>	use the DOS 5 editor to edit the file "file.ext"
<code>a:</code>	change to the A: drive
<code>md c:\myfiles</code>	make a new subdirectory named "myfiles"
<code>cd c:\myfiles</code>	change to subdirectory "myfiles"
<code>rd c:\myfiles</code>	remove the existing subdirectory names "myfiles"
<code>del file.ext</code>	delete a file named "file.ext"
<code>ren f1 f2</code>	rename file "f1" to "f2"
<code>copy f1 f2</code>	copy file 'f1' to "f2"
<code>verify on</code>	turn on verification of copy commands
<code>verify off</code>	turn off verification of copy commands
<code>xcopy d1 d2 /s</code>	copy all files and subdirectories in directory "d1"
<code>xcopy d1 d2 /p</code>	ask for confirmation of each file before copying
<code>diskcopy a: b:</code>	duplicate a disk using two floppy drives
<code>diskcopy a: a:</code>	duplicate a disk using the same floppy drive
<code>format a:</code>	format floppy disk in drive a:
<code>format a: /s:</code>	format a bootable floppy disk (include system)
<code>backup c:\d1*.txt a:</code>	back up all files with the extension ".TXT" in "c:\d1\" to the "a:" floppy drive

backup c:\ a: /s	back up the entire C: drive to floppy drive a:
restore a: c:\d1*.txt	restore certain files to C: from A:
restore a : C:\s	restore back-up files and subdirectories
ver	check the version of DOS
time	check or correct the system time
date	check or correct the system date
cls	clear the screen
scandisk	scan and check the disk C: for errors
chkdsk	check disk and memory usage of the current disk
chkdsk /f	fix errors reported by chkdsk
chkdsk filename	check a particular file
chkdsk a:	check a particular disk (a floppy in the a: drive)
mem	check memory usage

Note : Replace the drive letter as needed for your own computer. Also, you can type these commands in either upper or lower case letter, because DOS does not distinguish case.

4.10 What you have Learnt

In this lesson you learnt about the need and functions of operating system. In particular, you learnt the basic features of Windows98. Windows is a GUI which use pictures, symbols and words on the computer monitor and can be controlled by a mouse. It handles internal function of the computer and manages computer memory input and output operations. Now you will be in a position to operate the computer.

4.11 Terminal Questions

1. Write the utilities of the following Tools tab options of the Properties Window in My Computer :
-

- (a) Error-checking status,
 - (b) Backup status, and Defragmentation status.
2. What are the utilities of the following in Windows 98?
 - (a) My Document
 - (b) Recycle Bin
 - (c) Internet Explorer, and
 - (d) Network Neighborhood
3. Explain briefly the functions of Start button and Taskbar.
4. Explain the steps in creating a folder.
5. Explain the steps in opening a file or folder.
6. Explain the following
 - (a) Minimize, Maximize/Restore, and Close
 - (b) Title Bar
 - (c) The scroll Box, and
 - (d) Menu Bar.
7. Explain the steps in using files and printers controlled by others?
8. What is a shortcut. How do you create shortcut.

4.12 Feedback to In-Text Questions

In-Text Questions 4.1

1. Creates the link between the user and computer. It handles internal function such as managing the computer memory. It is a graphical user interface. The world wide web is easily accessible from anywhere through Windows. Supports DVD and digital audio.
-

2.
 - (a) Moving the mouse to place the pointer on an item is called pointing.
 - (b) Pointing to an item on the screen and the quickly pressing and releasing the left mouse button is called clicking.
 - (c) Pointing to an item on the screen and then quickly pressing and releasing the mouse button twice is called double clicking.
 - (d) Pointing to an item on the screen and then holding down the mouse button is called dragging.
3. The Desktop is place where useful things are accessible by the user, depending on which options you choose during installations.
4. My Computer contains

hard drives 'C', floppy drive, CD ROM drive, printers, control panel, Dial up Networking and Scheduled task.

In-Text Questions 4.2

1. Instead of opening drives and folders in separate windows, the user can browse the content in a single Window in a hierarchical structure.
 2. On the desktop double click Recycle Bin. In Recycle Bin select the file or folder you want to permanently delete. On the toolbar click delete. The confirm file delete dialog box appears. Then click yes, the file is permanently deleted from the computer.
 3. Shortcut is just a pointer that lets you open the file quickly. If you delete a shortcut, the original file is not deleted.
-

5

Data Communication and Networking

5.1 Introduction

Today computer is available in many offices and homes and therefore there is a need to share data and programs among various computers. With the advancement of data communication facilities the communication between computers has increased and thus it has extended the power of computer beyond the computer room. Now a user sitting at one place can communicate with computers of any remote site through communication channel. The aim of this lesson is to introduce you the various aspects of computer network.

5.2 Objectives

After going through this lesson, you will be in a position to

- learn the basic elements of data communication system
 - describe communication protocols and data transmission modes
 - explain the use of computer network
 - describe different components of computer network
-

- identify different types of network
- understand what is internet and e-mail and its uses in modern communication
- appreciate the use of satellite communication

5.3 Data Communication

We all are acquainted with some sorts of communication in our day to day life. For communication of information and messages we use telephone and postal communication systems. Similarly data and information from one computer system can be transmitted to other systems across geographical areas. Thus data transmission is the movement of information using some standard methods. These methods include electrical signals carried along a conductor, optical signals along an optical fibers and electromagnetic areas.

Suppose a manager has to write several letters to various clients. First he has to use his PC and Word Processing package to prepare the letter, if the PC is connected to all the client's PC through networking, he can send the letters to all the clients within minutes. Thus irrespective of geographical areas, if PCs are connected through communication channel, the data and information, computer files and any other programs can be transmitted to other computer systems within seconds. The modern form of communication like e-mail and Internet is possible only because of computer networking.

Basic Elements of a Communication System

The following are the basic requirements for working of a communication system.

1. The sender (source) who creates the message to be transmitted
2. A medium that carries the message
3. The receiver (sink) who receives the message

In data communication four basic terms are frequently used. They are:

Data : A collection of facts in raw forms that become information after processing.

Signals : Electric or electromagnetic encoding of data.

Signaling : Propagation of signals across a communication medium.

Transmission : Communication of data achieved by the processing of signals.

5.4 Communication Protocols

You may be wondering how computers send and receive data across communication links. The answer is data communication software. It is this software that enables us to communicate with other systems. The data communication software instructs computer systems and devices as to how exactly data is to be transferred from one place to another. The procedure of data transformation in the form of software is commonly known as protocol.

The data transmission software or protocols perform the following functions for the efficient and error free transmission of data.

1. **Data sequencing** : A long message to be transmitted is broken into smaller packets of fixed size for error free data transmission.
2. **Data Routing** : It is the process of finding the most efficient route between source and destination before sending the data.
3. **Flow control** : All machines are not equally efficient in terms of speed. Hence the flow control regulates the process of sending data between fast sender and slow receiver.
4. **Error Control** : Error detecting and recovering is the one of the main functions of communication software. It ensures that data are transmitted without any error.

5.5 Data Transmission Modes

There are three ways for transmitting data from one point to another as shown in Fig. 5.1

1. **Simplex** : In simplex mode the communication can take place in one direction. The receiver receives the signal from the transmitting device. In this mode the flow of information is Uni-directional. Hence it is rarely used for data communication.
-

2. **Half-duplex** : In half-duplex mode the communication channel is used in both directions, but only in one direction at a time. Thus a half-duplex line can alternately send and receive data.
3. **Full-duplex** : In full duplex the communication channel is used in both directions at the same time. Use of full-duplex line improves the efficiency as the line turnaround time required in half-duplex arrangement is eliminated. Example of this mode of transmission is the telephone line.

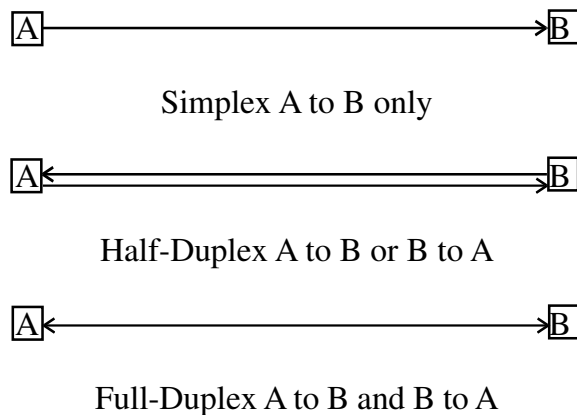


Fig 5.1 Simplex, Half-Duplex and Full-Duplex.

5.5.1 Digital and Analog Transmission

Data is transmitted from one point to another point by means of electrical signals that may be in digital and analog form. So one should know the fundamental difference between analog and digital signals. In analog signal the transmission power varies over a continuous range with respect to sound, light and radio waves. On the other hand, a digital signal may assume only discrete set of values within a given range. (see fig. 5.2 and 5.3) Examples are computer and computer related equipment. Analog signal is measured in Volts and its frequency is in Hertz (Hz). A digital signal is a sequence of voltage represented in binary form. When digital data are to be sent over an analog form the digital signal must be converted to analog form. So the technique by which a digital signal is converted to analog form is known as modulation. And the reverse process, that is the conversion

of analog signal to its digital form, is known as demodulation. The device, which converts digital signal into analog, and the reverse, is known as modem.

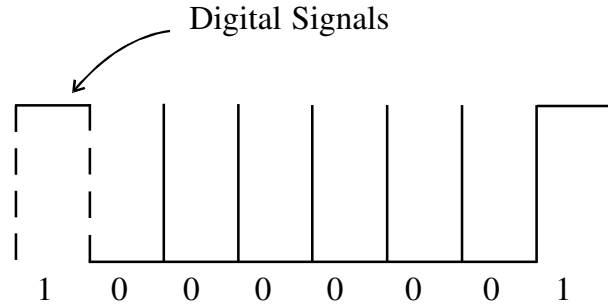


Fig. 5.2 Analog Signal

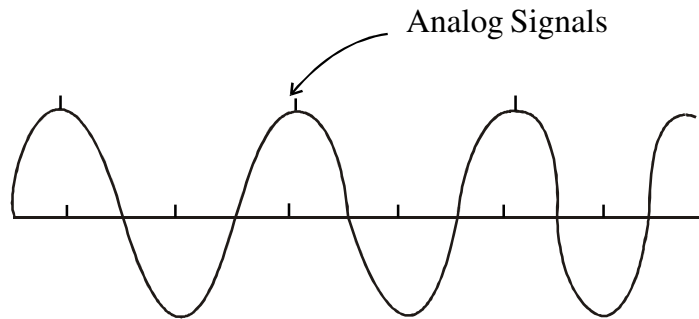


Fig. 5.3 Digital Signal

5.5.2 Asynchronous and Synchronous Transmission

Data transmission through a medium can be either asynchronous or synchronous. In asynchronous transmission data is transmitted character by character as you go on typing on a keyboard. Hence there is irregular gaps between characters. However, it is cheaper to implement, as you do not have to save the data before sending. On the other hand, in the synchronous mode, the saved data is transmitted block by block. Each block can contain many characters. Synchronous transmission is well suited for remote communication between a computer and related devices like card reader and printers.

5.6 Types of Communication Services

A term used to describe the data-handling capacity of a communication service is bandwidth. Bandwidth is the range of frequencies that is available for the transmission of data. A narrow range of frequencies in a communication system is analogous to a garden hose with a small diameter. The flow of information in such a system its data rate is restricted, just as is the flow of water in the narrow hose. Wider bandwidths permit more rapid information flow. The communication data transfer rate is measured in a unit called baud. Baud is identical to bits per second. Therefore, a rate of 300 baud is 300 bits per second.

Communication companies such as American Telephone and Telegraph (AT&T) and Western Union are called common carriers, and they provide three general classes of service for both voice and data communication:

1. Narrowband handles low data volumes. Data transmission rates are from 45 to 300 baud. The low-speed devices might use narrow band communications.
2. Voiceband handles moderate data transmission volumes between 300 and 9600 baud. They are used for applications ranging from operating a CRT to running a line printer. Their major application is for telephone voice communication hence, the term voiceband.
3. Broadband handles very large volumes of data. These systems provide data transmission rates of 1 million baud or more. High-speed data analysis and satellite communications are examples of broadband communication systems.

5.7 Communication Media

Following are the major communication devices which are frequently used :

1. **Wire Pairs** : Wire pairs are commonly used in local telephone communication and for short distance digital data communication. They are usually made up of copper and the pair of wires is twisted together. Data transmission speed is normally 9600 bits per second in a distance of 100 meter.
-

2. **Coaxial Cables** : Coaxial cable is groups of specially wrapped and insulated wires that are able to transfer data at higher rate. They consist of a central copper wire surrounded by an insulation over which copper mesh is placed. They are used for long distance telephone lines and local area network for their noise immunity and faster data transfer.
3. **Microwave** : Microwave system uses very high frequency radio signals to transmit data through space. The transmitter and receiver of a microwave system should be in line-of-sight because the radio signal cannot bend. With microwave very long distance transmission is not possible. In order to overcome the problems of line of sight and power amplification of weak signal, repeaters are used at intervals of 25 to 30 kilometers between the transmitting and receiving end.
4. **Communication Satellite** : The problem of line-sight and repeaters are overcome by using satellites which are the most widely used data transmission media in modern days. A communication satellite is a microwave relay station placed in outer space. INSAT-1 B is such a satellite that can be accessible from anywhere in India. In satellite communication, microwave signal is transmitted from a transmitter on earth to the satellite at space. The satellite amplifies the weak signal and transmits it back to the receiver. The main advantage of satellite communication is that it is a single microwave relay station visible from any point of a very large area. In microwave the data transmission rate is 16 giga bits per second. They are mostly used to link big metropolitan cities.

Intext Questions 5.1

1. What is communication protocol?
 2. What is the difference between asynchronous and synchronous transmission?
 3. Write True or False for the following statement
-

- (a) The basic requirements for working of a communication system are sender, medium and receiver.
 - (b) Electric or Electromagnetic encoding of data is called Transmission.
 - (c) In full duplex the communication channel issued in both directions at the same time.
 - (d) Analog signal is measured in Volts and its frequency is in Hertz
 - (e) The technique by which a digital signal is converted to analog form is known as modulation.
-

5.8 Computer Network

A computer network is interconnection of various computer systems located at different places. In computer network two or more computers are linked together with a medium and data communication devices for the purpose of communication data and sharing resources. The computer that provides resources to other computers on a network is known as server. In the network the individual computers, which access shared network resources, are known as nodes.

5.9 Types of Networks

There are many different types of networks. However, from an end user's point of view there are two basic types:

- **Local-Area Networks (LANs)**

The computers are geographically close together (that is, in the same building).

- **Wide-Area Networks (WANs)**

The computers are farther apart and are connected by telephone lines or radio waves.

In addition to these types, the following characteristics are also used to categorize different types of networks.

- **Topology**

The geometric arrangement of a computer system. Common topologies include bus, star, and ring.

- **Protocol**

The protocol defines a common set of rules and signals that computers on the network use to communicate. One of the most popular protocols for LANs is called Ethernet. Another popular LAN protocol for PCs is the IBM token-ring network.

- **Architecture**

Networks can be broadly classified as using either peer-to-peer or client/server architecture.

Computers on a network are sometimes called nodes. Computers and devices that allocate resources for a network are called servers.

- **Local Area Network (LAN)**

LAN is a computer network that spans a relatively small area. Most LANs are confined to a single building or group of buildings. However, one LAN can be connected to other LANS over any distance via telephone lines and radio waves. A system of LANs connected in this way is called a wide-area network (WAN).

Most LANs as shown in Fig. 5.4 connect workstations and personal computers. Each node (individual computer) in a LAN has its own CPU with which it executes programs, but it is also able to access data and devices anywhere on the LAN. This means that many users can share expensive devices, such as laser printers, as well as data. Users can also use the LAN to communicate with each other, by sending e-mail or engaging in chart sessions.

There are many different types of LANs-token-ring networks, Ethernets, and ARCnets being the most common for PCs.

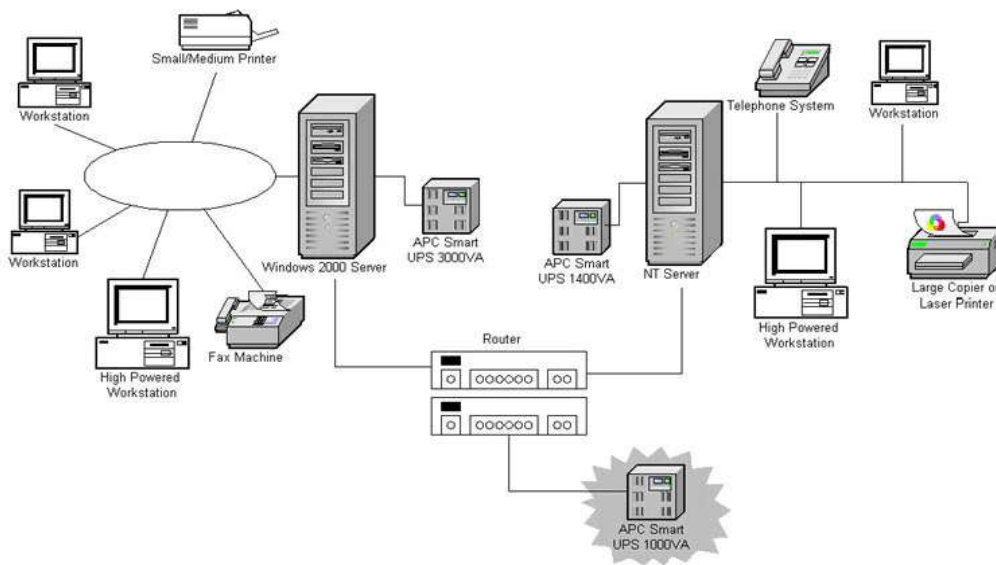


Fig. 5.4 Local Area Network

LANs are capable of transmitting data at very fast rates, much faster than data can be transmitted over a telephone line; but the distance are limited, and there is also a limit on the number of computers that can be attached to a single LAN.

Wide Area Network (WAN)

A WAN is a computer network that spans a relatively large geographical area. Typically, A WAN consists of two or more local-area networks (LANs).

Computers connected to a wide-area network are often connected through public networks, such as the telephone system. They can also be connected through leased lines or satellites. The largest WAN in existence is the Internet.

A typical WAN set up is shown in Fig. 5.5

Fig. 5.5 Wide Area Network

Network Topologies

As we have seen earlier, topology is the geometric arrangement of the computers in a network. Common topologies include star, ring and bus.

Star Network

The star network as shown in Fig 5.6 is frequently used to connect one or more small computers or peripheral devices to a large host computer or CPU. Many organizations use the star network or a variation of it in a time-sharing system, in which several users are able to share a central processor.

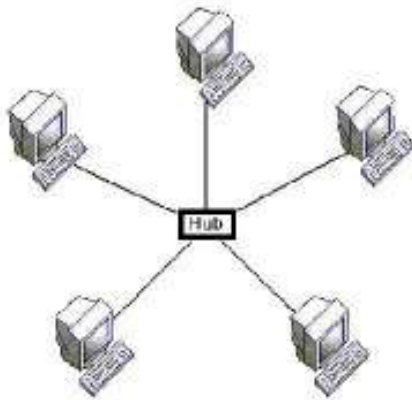


Fig. 5.6 Star Topology

In a time-sharing setup, each terminal receives a fixed amount of the CPU's time, called a time slice. If you are sitting at a terminal and cannot complete your task during the time slice, the computer will come back to you to allow you to do so. Actually, because the CPU operates so much faster than terminals, you will probably not even notice that the CPU is away.

By establishing time-sharing, many people in a large organization can use a centralized computing facility. Time-sharing can also be purchased from an outside service, which is an economical way to operate for a small company that cannot afford its own large computer.

Star network is frequently used in a LAN to connect several microcomputers to a central unit that works as a communications controller. If the user of one microcomputer wants to send a document or message to a user at another computer, the message is routed through the central communications controller. Another common use of the star network is the feasibility of connecting several microcomputers to a mainframe computer that allows access to an organization's database.

Access and control of star network typically is maintained by a polling system. Polling means that the central computer, or communications controller "polls" or asks each device in the network if it has a message to send and then allows each in turn to transmit data.

Ring Network

The ring network (see Fig. 5.7) is a Local Area Network (LAN) whose topology

is a ring - can be as simple as a circle or point-to-point connections of computers at dispersed locations, with no central host computer or communications controller. That is, all of the nodes are connected in a closed loop. Messages travel around the ring, with each node reading those messages addressed to it. One of the advantages of ring networks is that they can span larger distance than other types of networks, such as bus networks, because each node regenerates messages as they pass through it.

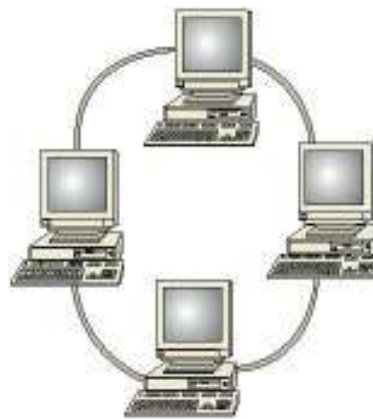


Fig. 5.7 Ring Topology

Access and control of ring networks are typically maintained by a "token-passing" system. IBM's Token-Ring network is thought by some observers to be a watershed event comparable to the development of the IBM PCV itself, because the Token-Ring network is designed to link all types of computers together, including not only personal computers but also possible mini computers and mainframes.

A Token-Ring network as shown in Fig. 5.7 resembles a merry-go-round. To deliver a message, you would hand over your addressed note to a rider (the token) on the merry-go-round, who would drop it off at the appropriate place.

Bus Network

Bus networks (see Fig. 5.8) are similar to ring network that the ends are not connected. All communications are carried on a common cable or bus and are available to each device on the network.

Access and control of bus networks are typically maintained by a method called contention, whereby if a line is unused, a terminal or device can transmit its message at will, but if two or more terminals initiate messages simultaneously, they must stop and transmit again at different intervals.

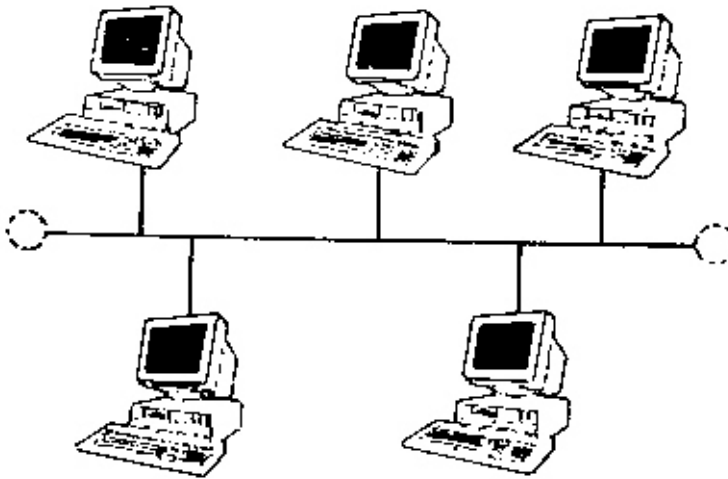


Fig. 5.8 Bus Network

5.10 Network Protocols

A protocol is an agreed-upon format for transmitting data between two devices. The protocol determines the following :

- The type of error checking to be used.
- Data compression method, if any
- How the sending device will indicate that it has finished sending a message
- How the receiving device will indicate that it has received a message

There are a variety of standard protocols from which programmers can choose. Each has its own particular advantages and disadvantages; for example, some are simpler than the others, some are more reliable, and some are faster.

From a user's point of view, the only interesting aspect about protocols is that your computer or device must support the right ones if you want to communicate with other computers. The protocol can be implemented either in hardware or in software. Some of the popular protocols are TCP/IP, HTTP, FTP, SMTP, POP, Token-Ring, Ethernet, Xmodem, Kermit, MNP, etc.

5.11 Network Architecture

The term architecture can refer to either hardware or software, or a combination of hardware and software. The architecture of a system always defines its broad outlines, and may define precise mechanisms as well.

An open architecture allows the system to be connected easily to devices and programs made by other manufacturers. Open architectures use off-the-shelf components and conform to approved standards. A system with a closed architecture, on the other hand, is one whose design is proprietary, making it difficult to connect the system to other systems. As we have seen before, network architectures can be broadly classified as using either peer-to-peer or client/server architecture.

Peer-to-peer Architecture

This is a type of network in which each workstation has equivalent capabilities and responsibilities. This differs from client/server architecture, in which some workstations are dedicated to serving the others. Peer-to-peer networks are generally simpler and less expensive, but they usually do not offer the same performance under heavy loads.

Client/Server Architecture

This is a network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processors dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are less powerful PCs workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power.

Intext Questions 5.2

1. Distinguish between LAN and WAN.
 2. What are the different types of network topology?
 3. Write True (T) or False (F) for the following statements :
 - (a) Networks used to interconnect computers in a single room, within a building or buildings on one site are called Wide Area network (WAN).
 - (b) The term Wide Area Network (WAN) is used to describe a computer network spanning a regional, national or global area.
 - (c) The speed of data transmission is much higher in WAN than in LAN.
-

5.12 Important terms used in Networking

(a) Internet

The newest type of network to be used within an organisation is an internet or internet web. Such networks enable computers (or network) of any type to communicate easily. The hardware and software needs are the same as for the internet, specifically TCP/IP, server and browser software used for the World Wide Web. Because most organisations have a need for more dynamic ways to link people and information, the internet market is expanding day by day. Moreover, there is no need to adjust the network when a new user joins in. With the help of Internet, all computers of an organisation can work as stand-alone systems, connected to a mainframe, or part of a LAN or WAN.

(b) E-Mail

E-mail stands for electronic mail. This is one of the most widely used features of Internet. Mails are regularly used today where without the help of postage stamp we can transfer mails anywhere in the world. With electronic mail the service is similar. But here data is transmitted through Internet and therefore within minutes the message reaches the destination may it be anywhere in the

world. Therefore the mailing system through e-mail is excessively fast and is being used widely for mail transfer.

(c) Voice Messaging

It is a new communication approach which is similar to electronic mail except that it is audio message rather than text messages that are processed. A sender speaks into a telephone rather than typing, giving the name of the recipient and the message. That sender's voice signal is then digitised and stored. The system can then either deliver the message at a specified time in future or it can be retrieved from a database by the recipient. The message is reconverted back into its analog format when it is delivered or retrieved so that the recipient hears it as the original sender's voice on a telephone. Voice messaging requires a computer with an ability to store the audio messages in digital form and then convert them back in an audio form upon verification. Each user has a voice mailbox in secondary storage and special equipment converts the audio message to and from the digital form. The main advantage of voice mail over electronic mail is that the sender does not have to type. Voice mail also makes it easy to include people in the firm's environment in an communication network.

(d) E-Commerce

Electronic commerce or e-commerce as it is popularly known refers to the paperless exchange of business information using Electronic Data Interchange, Electronic mail, Electronic Bulletin Boards, Electronic Fund Transfer and other network based technologies. Electronic Commerce (EC) not only automates manual process and paper transactions, but it also helps organisations to move into a fully electronic environment and change the way they usually operate. Few organisations have recently started conducting EC over Internet, the network of networks. Internet has also helped EC to boost up because it is a low cost alternative to the proprietary networks. EC standards are however under development. Electronic Data Interchange (EDI) is still the dominant part of EC.

Information Technology has transformed the way people work. Electronic Commerce (EC) has unearthed yet another revolution which is changing the way business houses buy and sell products and services. EC is associated with buying

and selling of products and services over computer communication networks. EC transfers information electronically from computer to computer in autonomous way. EC has, in fact, transformed the way organisations operate.

(e) Electronic Data Interchange (EDI)

EDI is the computer-to-computer exchange of business documents in a standard format. These formats look much like standard forms and are highly structured.

(f) Teleconferencing

It refers to electronic meetings that involve people who are at physically different sites. Telecommunication technology allows participants to interact with one another without travelling to the same location.

5.13 What you have Learnt

In this lesson you learnt about the importance and modes of communication through computers. Computers can communicate with one another through computer networking. There are two types of computer network: LAN and WAN. We discussed about the physical arrangements of computer and peripherals in network topology. There are three types of network topology: star topology, bus topology and ring topology. At the end we discussed about Internet and e-mail.

5.14 Terminal Questions

1. What is computer network? What are its main objectives?
 2. Distinguish between analog and digital transmission of data.
 3. Explain in brief different communication media.
 4. What is the difference between simplex and full-duplex transmission?
 5. Explain network architecture briefly.
 6. Define teleconferencing.
-

5.15 Feedback to In-Text Questions

In-text Questions 5.1

1. The data communication software instructs computer systems and devices as to how exactly data is to be transferred from one place to another. The procedure of data transformation in the form of software is commonly called protocol.
2. In asynchronous transmission data is transmitted character by character as you go on typing on a keyboard. On the other hand, in the synchronous mode, the saved data is transmitted block by block.
3. (a) True (b) False (c) True (d) True (e) True

In-Text Questions 5.2

1. LAN is restricted to limited geographical area whereas WAN covers long distance. In LAN the computer terminals and peripheral devices are connected with wires and cables whereas in WAN communication is done through telephone lines and satellite links. The speed of data transmission is much higher in LAN than in WAN.
 2. There are three major types of network topology. They are star topology, bus topology and ring topology.
 3. (a) False (b) True (c) False
-

6

Fundamentals of Internet and Java Programming

6.1 Introduction

Internet is the world's largest computer network, the network of networks, scattered all over the world. It was created nearly 25 years ago as a project for the U.S. Department of Defence. Its goal was to create a method for widely separated computers to transfer data efficiently even in the event of a nuclear attack. From a handful of computers and users in the 1960s, today the internet has grown to thousands of regional networks that can connect millions of users. Using internet, organisations all over the world can exchange data, people can communicate with each other in a faster and effective way. With the help of video conferencing over the internet, it has become possible that people can even see each other while they are communicating. Even one can do all his shopping sitting back at home. He does not bother to visit crowded market place for wasting his valuable time. Gradually, shopkeepers are also interested to opt for electronic commerce which provides them greater reach, faster and better ways to do business over the internet. Don't get surprised, if tomorrow you come to know that the paanwalla in your locality has started selling his paan over the internet.

Thus, the internet has thrown up a number of technological challenges like using the net for secure, private and fast communication as well as supporting video

and 3D graphics. Given this scenario of the internet, there was an implicit need to come up with a programming language to enable authors to incorporate program application fragments (applets) into the web document.

Finally this gave birth to Java programming language. We will discuss about this language in detail in later part of this lesson.

6.2 Objectives

After going through this lesson you will be in a position to :

- explain various terminology used in internet
- use various services provided by internet
- search for desired information over internet
- enjoy internet surfing
- define e-mail and its various features
- explain the features of World Wide Web (WWW)
- explain the utility of Java programming language and its characteristics
- identify the relationship between java and the internet

6.3 Internet - The History

In 1969, the U. S. Defence Department funded a project to develop a network, which can withstand the bombing. Basically the idea was to develop a very secured network which can work even after a nuclear attack. This project was known as ARPANET. The proposed network was not supposed to have a central control-which would be an obvious target. Ten years of research brought Local Area Networks (LANs) and workstations were developed to get connected to LAN. These workstations and LANs were then connected to the ARPANET. For next decade the ARPANET grew and its decentralized features helped its rapid expansion. Computers connected to ARPANET used a standard or rule to communicate with each other. This standard used by ARPANET is known as NCP (National Control Protocol). Protocol is a network term used to indicate

the standard used by a network for communication. But the passing time and rapid change in information technology suppressed NCP and brought TCP/IP (Transmission Control Protocol/Internet Protocol) into the world of networking. TCP converts messages into streams of packets at the source, and they are reassembled back into messages at the destination. Internet Protocol handles the dispatch of these packets. IP handles the addressing, and makes sure that a packet reaches its destination through multiple nodes and even across multiple networks with multiple standards. This flexibility of TCP/IP to handle multiple protocols encourages other networks to get connected to ARPANET. Slowly the ARPANET became a massive network of networks and now it is known as 'Internet'.

Internet is termed by some people as the world of largest democracy with no government. It has no state head to control it.

Why do people want to get connected to Internet? May be because of the freedom it provides. The internet is a rare example of a large democracy with no head of state, no official censors, no bosses, no board of directors. Nobody controls the Internet and in principle, any computer can speak to any other computer, as long as it obeys the technical rules of the TCP/IP protocol. This freedom of Internet helped it to move out of its original base in military and research institutions, into elementary and high schools, colleges, public libraries, commercial sectors even into the shop of a vegetable vendor.

6.4 Services of Internet - E-mail, FTP, Telnet, WWW

But what does one do with the Internet? May be four things basically mail, discussion groups, long-distance computing, and file transfers. Internet mail (e-mail or electronic mail) is much faster as compared to normal postal mail. One can also send software and certain forms of compressed digital image as an attachment. News groups or discussion groups facilitate Internet user to join for various kinds of debate, discussion and new sharing. Long-distance computing was an original inspiration for development of ARPANET and does still provide a very useful service on Internet. Programmers can maintain accounts on distant, powerful computers which execute programs. File transfers service allows Internet users to access remote machines and retrieve programs, data or text.

(a) E-Mail (Electronic Mail)

E-mail or electronic mail is paperless method of sending messages, notes or letters from one person to another or even many people at the same time via the Internet. E-mail is very fast compared to the normal post. E-mail messages usually take only few seconds to arrive at their destination. One can send messages anytime of the day or night and it will get delivered immediately. You need not have to wait for the post office to open and you don't have to get worried about holidays. It works 24 hours a day, seven days a week. What's more, the copy of the message you have sent will be available whenever you want to look at it even in the middle of the night. You have the privilege of sending something extra even such as a file, graphics, images etc. along with your e-mail. The biggest advantage to using e--mail is that it is cheap, especially when sending messages to other states or countries and at the same time it can be delivered to a number of people around the world.

Although e-mail is faster and cheaper, it has many of the components of regular mail. It allows you to compose note, get the address of the recipient and send it. Once the mail is received and read, it can be forwarded, replied. One can even store it for later use, or delete. In e-mail even the sender can request for delivery receipt and read receipt from the recipient.

(i) Features of E-mail

- One-to-one or one-to-many communication
- Instant communication
- Physical presence of recipient is not required
- Most inexpensive mail service, 24-hours a day and seven days a week
- Encourages informal communication

(ii) Components of an E-mail Address

As in the case of normal mail system, e-mail is also based upon the concept of a recipient address. The e-mail address provides all the information required to get a message to the recipient from anywhere in the world. Consider the following e-mail ID

john@ hotmail.com

In the example above, "john" is the local part, which is the name of a mailbox on the destination computer, where finally the mail will be delivered. Hotmail is the mailserver where the mailbox "john" exists,.com is the type of organisation on net, which is hosting the mail server.

There are six main categories;

.com Commercial institutions or organization

.edu Educational institutions

.gov Government site

.mil Military site

.net Gateways and administrative hosts

.org Private organizations

(b) FTP (File Transfer Protocol)

File Transfer Protocol, is an Internet utility software used to upload and download files. It gives access to directories or folders on remote computers and allows software, data and text files to be transferred between different kinds of computers. FTP works on the basis of the same principle as that of Client/Server. FTP client is a program running on your computer that enables you to talk to and get stuff from remote computers. The FTP client takes FTP commands and sends them as requests for information from the remote computer, known as FTP servers. To access remote FTP server it is required but not necessary to have an account in the FTP server. When the FTP client gets connected, FTP server asks for the identification in terms of User Login name and password of the FTP client. If one does not have an account in the remote FTP server, still he can connect to the server using anonymous login. This has been properly shown in Fig. 6.1

Using anonymous login anyone can login to a FTP server and can access public file archives, anywhere in the world, without having an account. One can easily Login to the FTP site with the username anonymous and e-mail address as password.

The basic objectives of FTP are

- to give flexibility and promote sharing of computer programs, files and data.
- to transfer data reliably and more efficiently over network.
- to encourage implicit or indirect use of remote computers using Internet.
- to shield a user from variations in file storage systems among hosts.

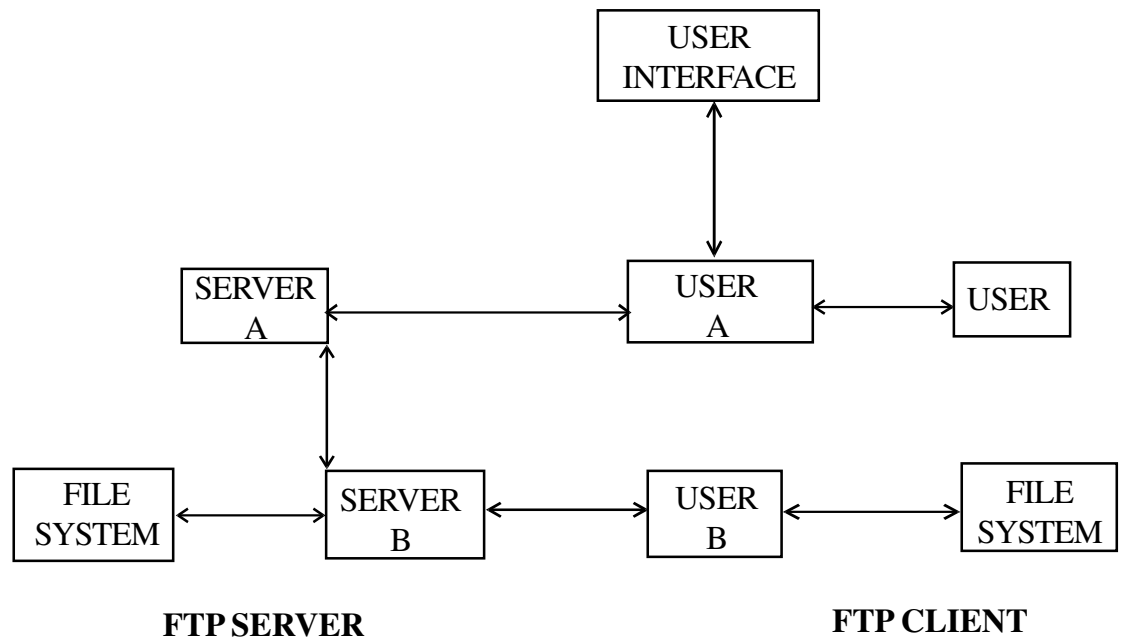


Fig. 6.1

The basic steps in an FTP session are:

- Start up your FTP client; by typing ftp on your system's command line c;>' prompt (or, if you are in a Windows, double-click on the FTP icon).
 - Give the FTP client an address to connect to. This is the FTP server address to which the FTP client will get connected.
 - Identify yourself to the FTP remote site by giving the Login Name
 - Give the remote site a password
-

- Remote site will verify the Login Name/Password to allow the FTP client to access its files.
- Look directory for files in FTP server
- Change Directories if required
- Set the transfer mode (optional);
- Get the file(s) you want; and
- Quit

(c) Telnet (Remote Computing)

Telnet or remote computing is telecommunication utility software, which uses available telecommunication facility and allows you to become a user on a remote computer. Once you gain access to the remote computer, you can use it for the intended purpose. The TELNET works in a very step by step procedure. The commands types on the client computer are sent to the local Internet Service Provider (ISP), and then from the ISP to the remote computer that you have gained access. Most of the ISP provides facility to TELNET into your own account from another city and check your e-mail while you are travelling or away on business.

The following steps are required for a TELNET session

- Start up the TELNET program;
 - Give the TELNET program an address to connect to
 - Make a note of what the "escape character" is;
 - Log in to the remote computer;
 - Set the terminal emulation;
 - Play around on the remote computer; and
 - Quit.
-

Intext Questions 6.1

1. Define Internet briefly.
 2. Write down the features of e-mail.
 3. What are the basic objectives of FTP?
 4. Fill in the blank.
 - (a) is very fast compared to the normal postal mail.
 - (b) is an Internet utility software used to upload and download files.
 - (c) The commands typed on the computer are sent to the local
 - (d) Internet is considered as the world's largest computer network, the of networks, scattered all over the world.
-

6.5 World Wide WEB (WWW)

WWW is the acronym for the World Wide Web. It is also commonly known as 'The Web'. The WWW is a hypertext based information retrieval tool. One can easily surf the Web by jumping from one document to another using the links in those documents. These documents can be in many formats, such as text, graphics, animation, sound and latest is video. They may also be a combination of all these. All the information on Internet are presented to the user as a document or more popularly known as Web Page. All these Web Pages are linked to each other or even to section within a Web Page and these links are known as Hyper Links.

The tool used to view these Web Pages on Internet is known as Internet browser or simply browser. It is a software program specifically developed to extract information on user request from the Internet and present them as a Web Page to the viewer. There are several browsers available in the market. However the most popular are Internet Explorer from Microsoft and Netscape from Netscape Inc. The process of using browser to view information on Internet is known as Browsing or Surfing. This has been show in Fig. 6.2

Fig.6.2

(a) Internet Address

Just like every house, every office, every location has an address every page on the Internet has a unique address. The address is used to get the web page for user from Internet.

Just as the address of a house or office is known as its postal address, the address on the Internet is known as URL (Uniform Resource Locator). A typical Internet

address or URL would look like:

`http://www.xyz.org/computers/internet/url.htm`

The URL locates a particular web page, among all the computers connected to the Internet. The URL contains the components that specify the protocol, server, and pathname of an item. Let us examine the URL given above (`http://www.nois.org/computers/inbtenet/url.htm`).

The protocol is followed by a colon (**http:**) the server is preceded by two slashes (`//www.xyz.org`), and each segment of the pathname is preceded by a single slash (`/computers/internet/url.htm`). A protocol is a set of rules that tells the computer how to interpret the information at that address.

The first component, the protocol, defines the manner for interpreting computer information. Many Internet pages use **HTTP** (Hyper Text Transfer Protocol). Other common Internet protocols that one might come across are **FTP** (File Transfer Protocol), **NEWS** (Usenet news groups protocol), and **GOPHER** (an alternative transfer protocol). Gopher protocol is mostly obsolete now.

The second component, the server (**www.xyz.edu**), identifies the computer system that stores the information you seek and is always preceded by two slashes. A server is a computer that has information stored on it and sends it to the client, when a request is made. Each server on the Internet has a unique address name whose text refers to the organization maintaining the server.

The last component (`/computers/internet/`), defines the path within the server where the requests item (**url.htm**) will be found. Most of the web pages will have **.htm** or **.html** as their secondary or extension name.

6.5.1 How to get connected to Internet

There are various type of connectivity to get connected to Internet. They all can be broadly classified as:

- Gateway Access
 - Dial-up Connection
 - Shell Connection
-

- TCP/IP Connection
- Leased Connection

Gateway Access

Gateway Access is also known as Level-One connection. It is the access to the Internet from a network, which is not on the Internet. The gateway allows the two different types of networks to "talk" to each other. But the users of the Gateway Network have limited access to the Internet. They might not be able to use all the tools available on Internet. The local Internet Service Provider (ISP) normally defines this limitation.

Good example of network with Level One Connectivity within India is that of VSNL (Videsh Sanchar Nigam Limited). All access to Internet from India are through VSNL gateway as shown in Fig. 6.4

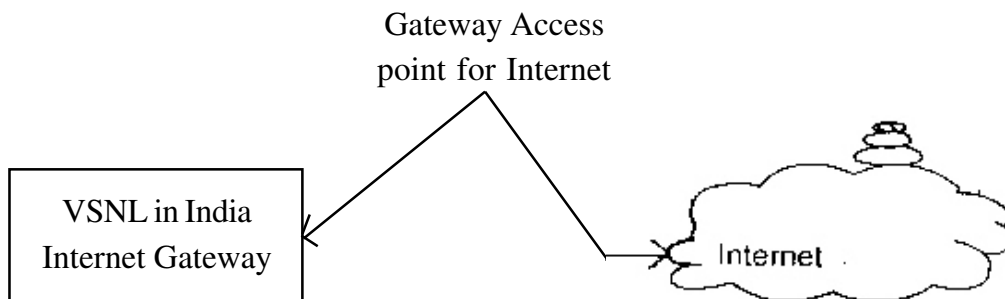


Fig. 6.4

Dial-up Connection

'Dial-up' connection is also known as Level Two connection. This provides connection to Internet through a dial-up terminal connection. The computer, which provides Internet access is known as 'Host' and the computer that receives the access, is 'Client' or 'Terminal'. The client computer uses a modem to access a "host" and acts as if it is a terminal directly connected to that host. So this type of connection is also known as 'Remote Modem Access' connection. And the host to which the client gets connected is actually connected to the Internet by a full time connection as shown in Fig 6.5

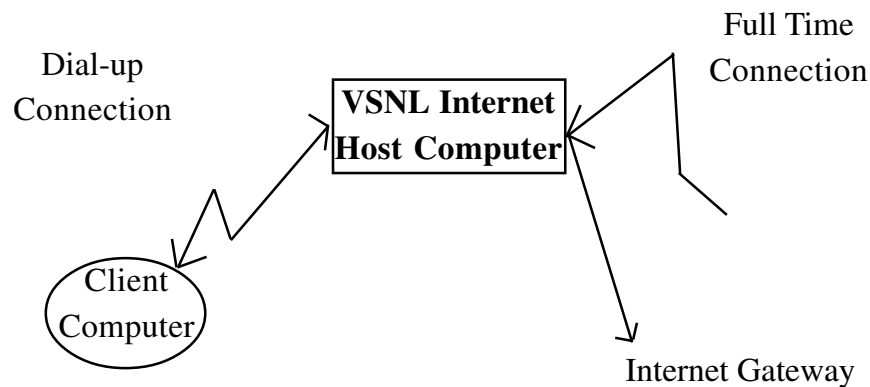


Fig. 6.5

In dial-up connection to Internet, Host carries all the command that are types on a client machine and forwards them to Internet. It also receives the data or information from the Internet on behalf of the 'Client' and passes it to them. The client computer acted as a 'dumb' terminal connected to remote host.

This type of connection can further be divided into two categories.

Shell Connection

In this type of Internet connection, the user will get only textual matter of a Web Page. This connection does not support Graphics display. However the user will be able to surf the Internet, do FTP, receive mail. Shell Accounts were the only type of Internet access available for many years before the Internet entered in to the world of graphics and became more users friendly.

TCP/IP Connection

Today's graphical World Wide Web browsers provides easier access with multimedia sound and pictures. The major difference between Shell and TCP/IP account is that shell can only display text and does not support graphics display, whereas TCP/IP can display both. Hence TCP/IP is more popular Internet connection. Shell accounts are slowly phasing out from the Internet scenario.

To access any of these dial-up accounts you need the followings;

- Computer
 - Modem
 - Telephone Connection
 - Shell or TCP/IP account from the ISP
 - Internet client software such as Internet browser
-

Leased Connection

Leased connection is also known as direct Internet access or Level Three connection. It is the secured, dedicated and most expensive, level of Internet connection. With leased connection, your computer is dedicated and directly connected to the Internet using high-speed transmission lines. It is on-line twenty-four hours a day, seven days a week. Leased Internet connections are limited to large corporations and universities who could afford the cost. These have been clearly shown in Fig. 6.6.

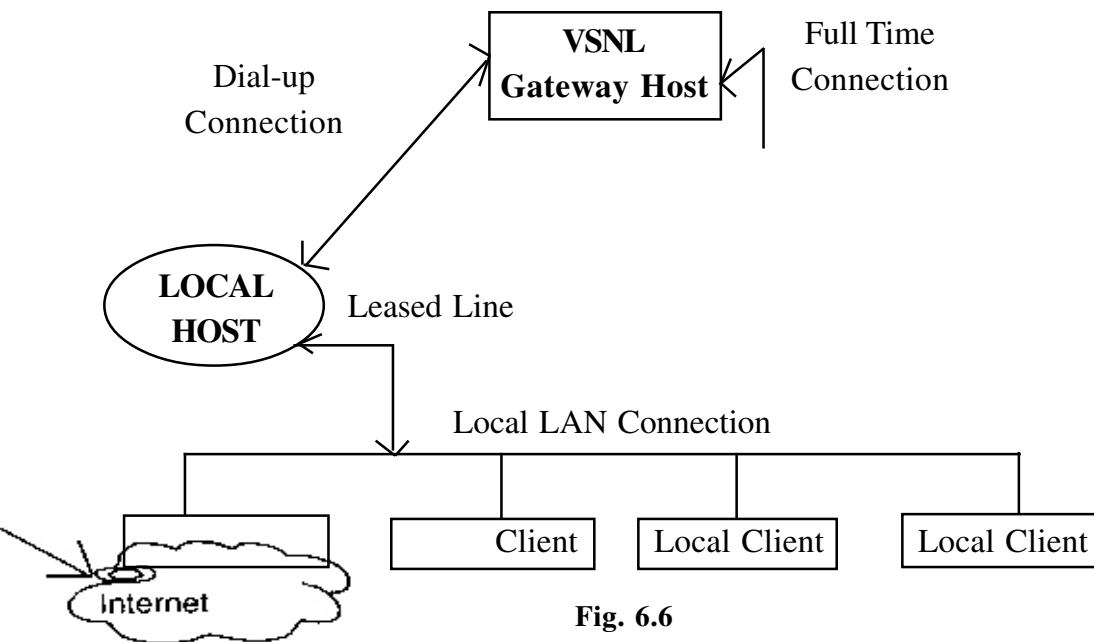


Fig. 6.6

Next Question 0.4

1. Define World Wide Web briefly
2. What are the various ways of connectivity to get hook on the Internet? Explain them briefly.
3. State whether the following statements are True or False:
 - (a) www is the acronym for Word Wide Web.
 - (b) In TCP/IP connection, the user will get only textual matter of a web page.
 - (c) Leased connection is also known as direct Internet access.
 - (d) Dial-up connection is also known as Level Two connection.

6.6 Java and C++

Java is a derivative of C and C++: simple, familiar syntax and with fewer complex features. It does not have many of the poorly understood, confusing and rarely used features of C++. There are no pointers in Java only tightly bounded arrays. This eliminates the possibility of over writing memory and corrupting data unwittingly. Pointers have been the source of raw power in C and C++ programs as well as the primary feature that helped introduce bugs into almost all programs. Java is sophisticated enough to help programmers express complex ideas using arrays and has an efficient automatic garbage collector based memory management scheme. In addition, unlike C or C++, Java does not support structures, enums or functions. Every programmable object is an instance of a 'class' in object oriented Java. This class definition permits both static and dynamic inheritance and hence full reuse of code.

Java does not support multiple inheritance whose semantic and usage has been quite controversial in other languages. Following the principles of structured programs there are no goto statements, no automatic typecasting or operator overloading.

6.7 Characteristic of Java

(i) Simple and Small

Java is easily programmable, with a minimum of training, since it is based on C++. Most of the development tools available today are huge in size and complexity, and require powerful development platforms. But Java's components are comparatively very small in size, enabling construction of compact software that can run 'stand-alone' even on small machines.

(ii) Object Oriented

An object-oriented design enables the development of small, modular, yet self-contained units, each of which could later be used as building block for any specific application. It also facilitates a clean definition of interfaces and allows 'plug-and-play' features, wherein reusable software components can be distributed easily across diverse hardware platforms. Programmers can even attach their applications and data together, and bundle the combination as a single

entity.

Java retains many of the object oriented features and the 'look and feel' of C++. Hence, programmers can migrate easily to Java and be productive quickly.

To be truly considered as 'object oriented' a programming language should support a minimum of four characteristics.

- Encapsulation - implements information hiding and modularity (abstraction).
- Polymorphism - the same message sent to different objects results in behaviour that is dependent on the nature of the object receiving the message.
- Inheritance - you define new classes and behaviour based on existing classes to obtain code re-use and code organization.
- Dynamic binding - objects could come from anywhere possibly across the network. You need to be able to send messages to objects without having to know their specific type at the time you write your code. Dynamic binding provides maximum flexibility while a program is executing.

Now let us see that are objects. They are software programming models. In your everyday life you are surrounded by objects: cars, coffee machines, trees and so on. Software applications contain objects : buttons on user interfaces, spreadsheets, spreadsheet cells and so on. These objects have state and behaviour. You can represent all these things with software constructs called objects, which can also be defined by their state and behaviour. A car can be modelled by an object. It has state (how fast it is going, in which direction, its fuel consumption and so on) and behaviour (starts, stops, turns etc.)

Classes are fundamental entities in Java, as they are in any object oriented language. These classes are ways of representing certain sets of data that have something in common and of providing an interface for the use of this data. Encapsulating the data makes it easy to use, safe from outside interference and also make changing the interface and data content easy. It is really simple to reuse a great portion of code already written in Java, without bothering about

how it is implemented. Java has a huge repository of reusable code, called the Application Programming Interface (API), which greatly simplifies basic programming.

(iii) Portable and Interpreted

A software application written in Java, be it a word-processor or spreadsheet, a multimedia tool or virtual reality designer, will run perfectly on any computer, on any platform. A Java application is a combination of a compiled and interpreted language. The source code written by a programmer is first compiled into Byte Code which is uniform for all machines. This Byte Code is compatible with the Java interpreter which is termed as Java Virtual Machine. The interpreter converts the Byte Code into specific machine code for direct execution, on any machine to which it has been ported.

So, you may write your code anywhere and run it anywhere else. This is very useful because most programmers worry whether their programs will run on other systems.

(iv) Architecture-Neutral

The Java Compiler generates an architecture-neutral object file, 'generic' Byte Code instructions, which have nothing to do with a particular computer architecture. This compiled code is executable on any processor, given the presence of the Java run-time system, which will translate it into native machine code on the fly. The same version of the software runs on all platforms, independent of any CPU or hardware architecture, across networks.

(v) Dynamic

By making interconnections between modules at run-time, Java completely avoids dependency on other software components like libraries. Therefore, Java applications are always ready-to-run, without any side-effects, irrespective of changes or revisions that may take place in those external components.

(vi) Distributed

Java has an extensive library of routines for easy distribution of applications as it abides by industry standard networking protocols like TCP/IP, HTTP, FTP. Java

applications can open and access objects across the net via URLs with the same ease that programmers are used to when accessing a local file system over a network.

(vii) Automatic Memory Management

The Java garbage collector keeps tracks of all objects generated, automatically freeing the memory used by objects that have no further use and are not referred to by other existing objects. This contributes enormously to making the code robust.

(viii) Multi-Threaded

Multithreading is a way of building applications with multiple processes. Support for multithreading enables the efficient execution of programs that potentially have multiple threads of control. Thus, the garbage collector is efficiently run as a background process overcoming the biggest drawback of garbage collector based languages. Multithreading is supported via inheritable thread class libraries. The result is better interactive response and near real-time features. This enables Java to interface and support many features of modern operating systems and new protocols.

Modern network based applications, such as the Hot Java WWW browser, typically need to do several things at the same time. A user working with Hot Java can run several animation concurrently while downloading an image and scrolling the page, Java's multithreading capability provides the means to build applications with many concurrent threads of activity. Multithreading thus results in a high degree of interactivity for the end user.

(ix) Robust

Java programs are robust because explicit memory manipulations by the programmer are prevented.

(x) Secured

Java programs are secured : distributed applications have to exhibit the highest levels of security concerns. A typecode verifier in the Java interpreter ensures that the compiled code is strictly language compliant, thus trapping all malafide modifications, more so the computer viruses parading as legal code.

6.8 How Java ignores the identity of your chip?

The chip inside your computer understands a machine code only. Since this is too complex, programmers use a high level language. This is the source code. This is converted into a machine code which the CPU understands. However, the machine code varies from one chip to another. A compiler is used to execute the conversion. However, the translation takes place in the software developer's office and it is the translated version you buy and install in your computer.

Naturally, an application converted into the machine code for an Intel chip will vary from the version for a Motorola chip, which will be different from the version for a Sparc Workstation, which will have nothing in common with that required for a Silicon Graphics machine and so on.

But, Java follows a different route. In this context, we should know more about compiling and interpreting. Compiled programs make the entire machine code available to the computer at one go. Everytime you use the application, every line of machine code is installed in the memory of your computer. In contrast, interpreted software is converted according to your commands : everytime you want to perform a particular operation, the interpreter seeks out the appropriate lines of source code and converts them, thus taking more time.

We have seen earlier that Java is perfectly portable across different computer platforms. A platform is a combination of the chip and the operating system, such as Intel-plus-Windows 95. Java is actually a combination of a compiled and interpreted language. First, the source code is compiled using the conventional system into a machine code. This is called byte code. The compiled version of Java is identical for all computers. This byte code is actually compatible with a computer that exists not as a hardware but as software : the Java Virtual Machine (JVM).

Essentially, a JVM is nothing but an interpreter. It takes the bytecode and converts it into the specific machine code for the actual chip of your computer. This offers perfect portability. All that a computer, which wants to run Java applications, will need is the appropriate interpreter which naturally will vary for different chips. To the software developer, your computer is no longer an Intel PC, a PowerMac, an Alpha Workstation or a Sun Spare Station; it is simply a JVM. So, he has to write only one version of a program which has to be compiled in only

one way, liberating you from having to worry about compatibility between software and hardware. The hardware configuration before and after Java is shown in Fig. 6.7 and Fig. 6.8 respectively.

SOURCE CODE

Fig.6.8 Hardware Compatibility Post-Java

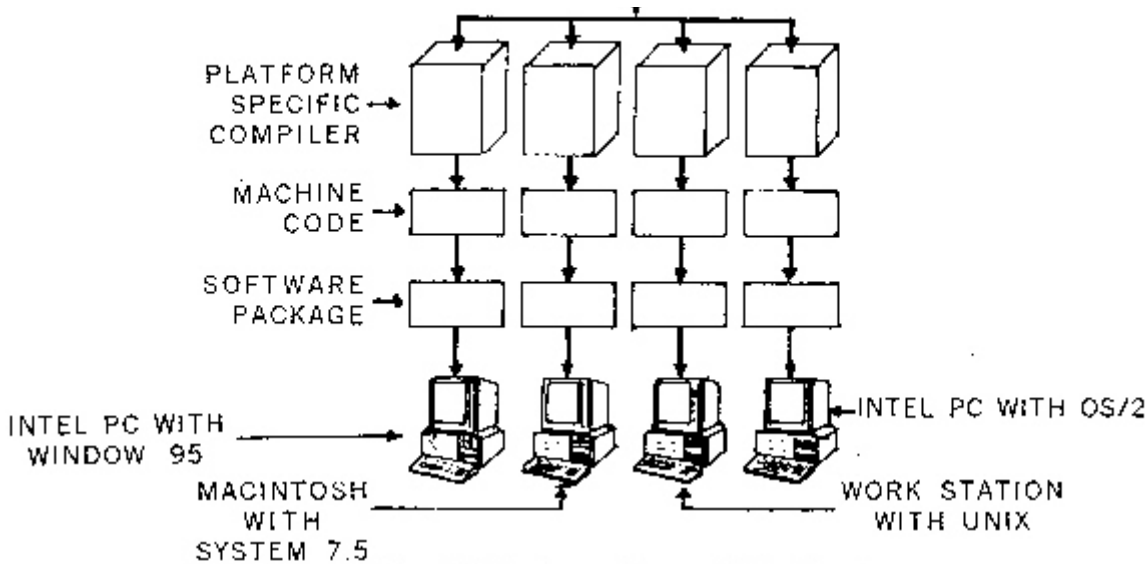
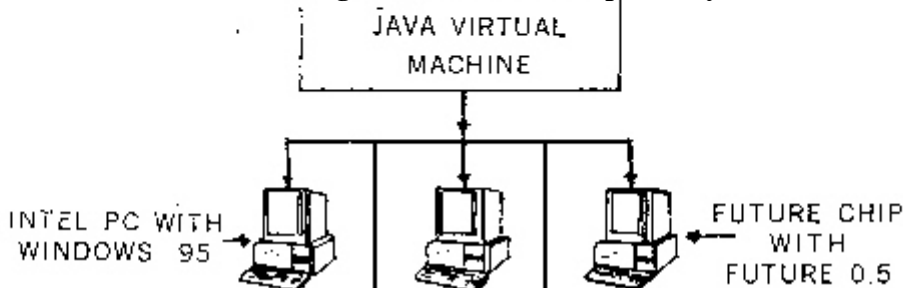


Fig. 6.7 Hardware Compatibility Pre-Java

SOURCE CODE

Fig.6.8 Hardware Compatibility Post-Java



6.9 Software Business after Java

Huge software packages with features that you may virtually never use will become redundant. Instead, computing will be dominated by tiny disposable applications - called applets - written for specific task only.

Applets are nothing but Java programs, usually small and having very limited functionality, so that they can be downloaded by a browser and run. This adds life to static Web contents with minimum effort on the part of the web page designer.

The applets are referenced by a special tag in an HTML file - the <applet> tag. Applets can be located anywhere, either on your local machine or somewhere out on the Internet. It has to be specified as a URL. The location of the applet is completely invisible to the user. The browser fetches and runs it.

Consider a stock market analyst who downloads the day's stock quotations from around the world and runs calculations on them. Today he must have the software on the hard disk and use the Net only as a data source. Post-Java, he will download from the Net, not just the prices, but along with them, an applet that lodges itself in the memory of his computer for say, half an hour. Its task will be to perform comparisons. If the analyst needs to turn the data into bar-charts, he has to use a different applet which will also be available on the Net. After his use, the applet will disappear from the computer.

You need no longer worry about the high prices for the packages and their upgrades. In the applet universe, users will need nothing more than a simple word processing package. If you want any other occasionally used feature, you may download for temporary use from the Net on a just-in-time-basis. You need not buy packages for permanent ownership. Instead, you will be billed on pay-per-use basis.

Further, Java will demolish the entry barrier of expensive distribution and retail networks that mega software vendors have created to keep small companies out of the market. Small, low-cost programs will become viable once they are distributed on the Net. Innovation and programming skills will dominate at the expense of hardsell. With respect to software the pre and post Java scene is illustrated in Fig. 6.9 and Fig. 6.10 respectively.

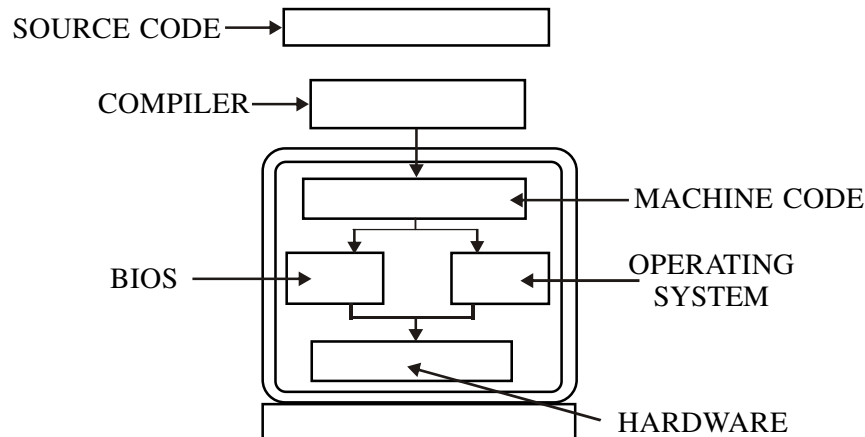


Fig. 6.9 A-Software Architecture Pre-Java

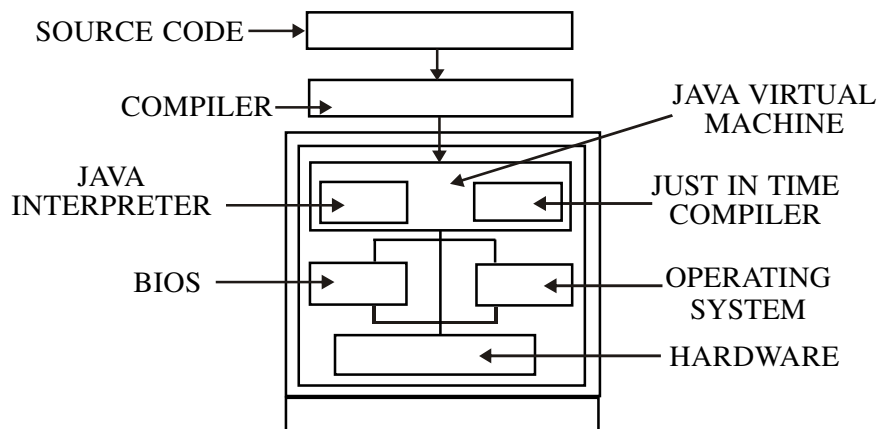


Fig. 6.10 Software Architecture Post-Java

6.10 Java and the Internet

The Java programming language enables the Web documents authors to deliver small application programs to anyone browsing the pages of the html documents. The page becomes alive because it can create game score boards, execute animated cartoons, audio files and video clippings. In addition, it changed the way Internet and WWW worked by allowing architecturally neutral compiled code to be dynamically loaded from anywhere in the network of heterogeneous systems and executed transparently. The Internet usage before and after Java is shown in Fig. 6.11 and Fig. 6.12.

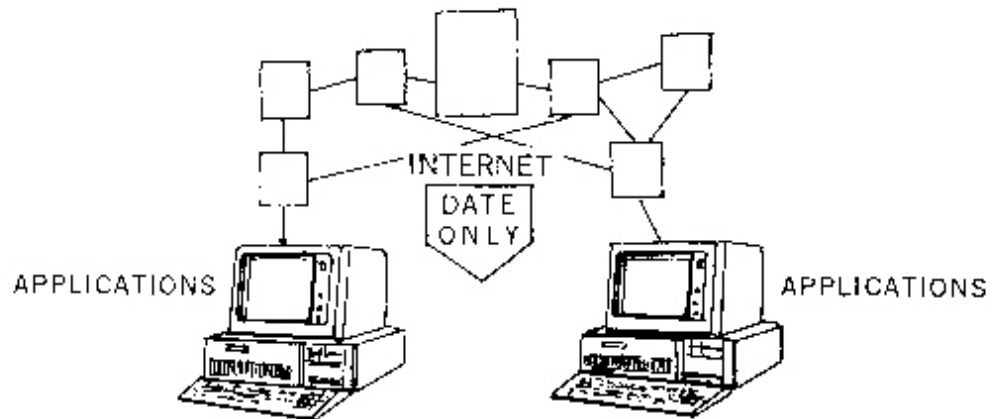


Fig.6.11 Internet Usage Pre-Java

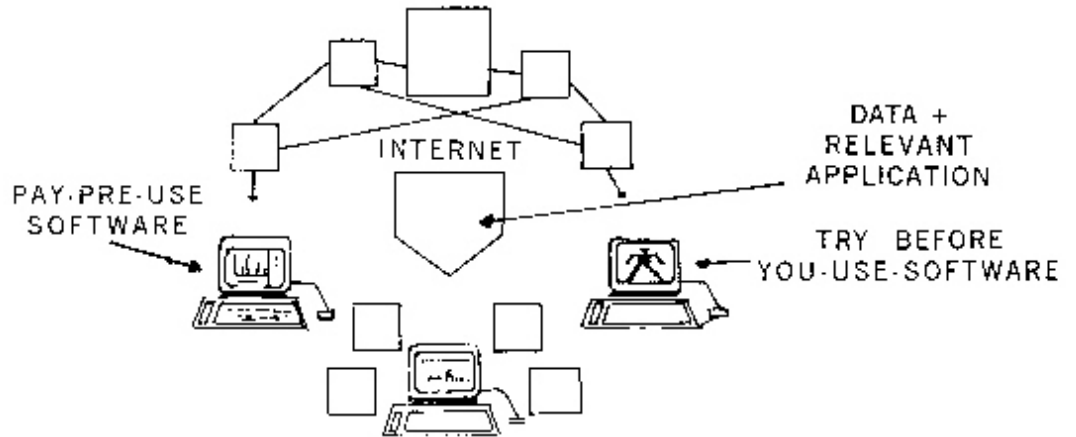


Fig. 6.12 Internet Usage Post-Java

With Java, data and programs - the twin staples of computing - do not have to be stored on your computer anymore. They can reside anywhere on the Internet, called up by whoever needs them and whenever they need them. The WWW introduced millions of users to the world of the Internet. But for their charm, most Web pages themselves are nearly as static as their book or magazine counterparts. Although the Web is technically equipped to supply video and sound, the user needs to have the right software to make use of them.

Java allows you to package software along with data. In other words, when you download a video clip, you also automatically get the software needed to play it.

One thing is certain. The Web has become a powerful computer which anybody can use, so costly desktop systems will soon be replaced by a simple computer connected to the Internet and capable of executing any program dynamically loaded from anywhere in the world. Companies will sell computing capabilities over the Internet, just like our electricity boards supply us with power. In the near future, a sizable part of our preferred activity will be over the Internet and our success depends on our ability to adapt to and use the Internet technologies.

Future of Java

Java runs slower than conventional programming languages and the software libraries that streamline a programmer's task are still being written for it. Java has been a programming language for clients with a rather weak server support: Most of the programs developed in Java are applets designed to perform small functions like animating logos, sketching cartoon characters, etc. Application software for client/server environments are to be developed. For these kinds of applications, Java needs a clean, higher-level interface to databases and servers, like ODBC (Object Data Base Connectivity) and OLE (Object Linking and Embedding), to access data.

While these issues are being addressed, India has also jumped on the Java bandwagon. Java is seen as a very good opportunity for the Indian software industry. With an online connectivity to the rest of the world and a steady growth rate in the industry, Java would probably catalyse the reorientation required for the Indian software industry to make its mark globally.

Intext Questions 6.3

1. Write down the various characteristics of Java Programming language.
 2. State whether the following statements are True or False:
 - (a) There are no pointers in Java-only tightly bounded arrays.
 - (b) Java does not retain many of the object oriented features and the 'look and feel' of C++.
 - (c) A Java application is a combination of a compiled and interpreted
-

language.

- (d) Java programs are robust but lack in security.
-

6.11 What you have learnt

In this lesson, we have learnt the basic concept of Internet, its history and various types of services provided by the internet. In this process, we could get an idea about the use of e-mail and file transfer protocol. Now we have fair knowledge to access Internet and can put it to various uses. Java programming language along with its characteristics has been explained briefly in the last part of this lesson.

6.12 Terminal Questions

1. Write down a brief history of Internet.
2. Write down the various features of e-mail.
3. What are the basic objectives of FTP? Explain briefly.
4. State briefly about various types of connectivity to get hooked on to Internet
5. What do you know about Java programming language?

6.13 Feedback to In-Text Questions

In-Text Questions 6.1

1. Internet is considered as the network of networks, scattered all over the world. It is undoubtedly the world's largest network carrying information on almost any subject under the sun.
 2. Features of e-mail are:
 - (a) One-to-one or one-to-many communication.
 - (b) Instant communication
-

- (c) Physical presence of recipient not required
 - (d) Most inexpensive mail service, 24 hours a day and seven days a week.
 - (e) Encourages informal communication
3. Basic objectives of FTP are:
- (a) to give flexibility and promote sharing of computer programs, files and data.
 - (b) to transfer data reliably and more efficiently over network.
 - (c) to encourage implicit or indirect use of remote computers using Internet.
 - (d) to shield a user from variations in file storage systems among hosts.
4. (a) E-mail (b) FTP (c) ISP
- (d) network

In-Text Questions 6.2

1. The world wide web or simply the web is built on top of the Internet. It is hypertext based information retrieval tool.
2. Various ways of connectivity are:
- (a) Gateway Access
 - (b) Dial-up Connection
 - (c) Shell Connection
 - (d) TCP/IP Connection
 - (e) Leased Connection
3. (a) False (b) False (c) True (d) True
-

In-Text Questions 6.3

1. Various characteristics are:
 - (a) Simple and small
 - (b) Object oriented
 - (c) Portable and interpreted
 - (d) Dynamic and distributed
 - (e) Robust and Secure
2. (a) True (b) False (c) True (d) False

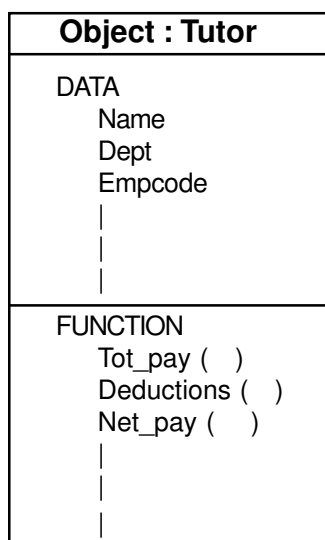
7

Introduction to C++

7.1 Introduction

C++ is an extension to C Programming language. It was developed at AT&T Bell Laboratories in the early 1980s by Bjarne Stroustrup. It is a deviation from traditional procedural languages in the sense that it follows object oriented programming (OOP) approach which is quite suitable for managing large and complex programs.

An object oriented language combines the data to its function or code in such a way that access to data is allowed only through its function or code. Such combination of data and code is called an object. For example, an object called Tutor may contain data and function or code as shown below:



The data part contains the Name, Dept and Employee code. The function part consists of three functions: To_pay (), Deductions () and Net_pay ()

Thus, the various objects in the object-oriented language interact with each other through their respective codes or functions as shown in Fig. 7.1

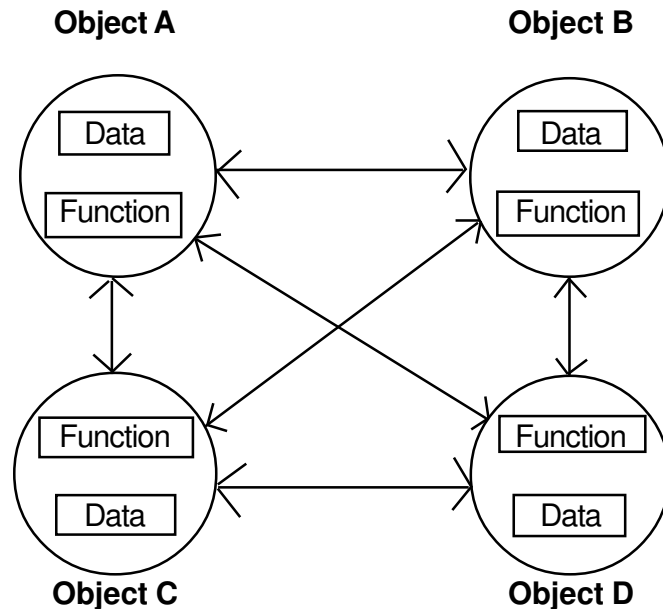


Fig. 7.1: The Object-oriented approach

It may be noted here that the data of an object can be accessed only by the functions associated with that object. However, functions of one object can access the functions of other objects.

7.2 Objectives

After going through this lesson, you would be able to:

- learn about C++ character set, tokens and basic data types
 - explain the utility of different types of operators used in C++
 - identify the difference between implicit and explicit conversions
 - explain about Input/Output streams supported by C++
 - explain the structure of a C++ program
 - write a simple program in C++
-

7.3 C++ Character Set

Character set is a set of valid characters that a language can recognise. A character represents any letter, digit or any other special character. The C++ has the following character set.

Letters A-Z, a-z

Digits 0-9

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

White spaces Horizontal tab (→), Blank space,
 Carriage return (↵) Newline, form feed.

7.4 Basic Data Types

Every program specifies a set of operations to be done on some data in a particular sequence. However, the data can be of many types such as a number, a character, boolean value etc. C++ supports a large number of data types. The built in or basic data types supported by C++ are integer, floating point and character type. A brief discussion on these types is given below:

7.4.1 Integer type (int)

An integer is an integral whole number without a decimal point. These numbers are used for counting. For example 26, 373, -1729 are valid integers.

Normally an integer can hold numbers from -32768 to 32767. However, if the need be, a long integer (long int) can also be used to hold integers from -2, 147, 483, 648 to 2, 147, 483, 648.

7.4.2 Floating point type (float)

A floating point number has a decimal point. Even if it has an integral value, it must include a decimal point at the end. These numbers are used for measuring quantities.

Examples of valid floating point numbers are: 27.4, -927., 40.03

A float type data can be used to hold numbers from 3.4×10^{-38} to $3.4 \times 10^{+38}$ with six or seven digits of precision. However, for more precision a double precision type (double) can be used to hold numbers from 1.7×10^{-308} to $1.7 \times 10^{+308}$ with about 15 digits of precision.

7.4.3 Character Type ((Char)

It is a non numeric data type consisting of single alphanumeric character. Examples of valid character types are : 'A', '9', 'P', '8', '&'.

It may be noted here that 9 and '9' are of different data types. The former is of type int and later of type char.

7.5 Tokens

A token is a group of characters that logically belong together. The programmer can write a program by using tokens. C++ uses the following types of tokens.

- Keywords
- Identifiers
- Literals
- Punctuators
- Operators

7.5.1 Keywords

There are some reserved words in C++ which have predefined meaning to compiler called keywords. Some commonly used keywords are given below:

List of Keywords

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

7.5.2 Identifiers

Symbolic names can be used in C++ for various data items used by a programmer in his program. For example, if a programmer wants to store a value 50 in a memory location, he/she choose any symbolic name (say MARKS) and use it as given below:

MARKS = 50

The symbol '=' is an assignment operator. The significance of the above statement is that 'MARKS' is a symbolic name for a memory location where the value 50 is being stored.

A symbolic name is generally known as an identifier. The identifier is a sequence of characters taken from C++ character set. The rules for the formation of an identifier are:

- (i) An identifier can consist of alphabets, digits and/or underscores.
- (ii) It must not start with a digit.
- (iii) C++ is case sensitive, i.e., upper case and lower case letters are considered different from each other. It may be noted that TOTAL and total are two different identifier names.
- (iv) It should not be a reserved word.

7.5.3 Literals

Literals (often referred to as constants) are data items that never change their value during the execution of the program. The following types of literals are available in C++.

- (i) integer-constants
- (ii) character-constants
- (iii) floating-constants
- (iv) string-literals

(i) Integer constants

Integer constants are whole numbers without any fractional part. It may contain either + or – sign, but decimal point or commas does not appear in any integer constant. C++ allows three types of integer constants.

- Decimal (Base 10)
 - Octal (Base 8)
 - Hexadecimal (Base 16)
-

- **Decimal integer constants**

It consists of sequence of digits and should not begin with 0 (zero). For example 124, - 179, + 108.

- **Octal integer constants**

It consists of sequence of digits starting with 0 (zero). For example, 014, 012.

- **Hexadecimal integer constant**

It consists of sequence of digits preceded by ox or OX. For example OXD, OXC. The suffix l or L and u or U attached to any constant forces it to be represented as a long and unsigned respectively.

(ii) **Character constants**

A character constant in C++ must contain one or more characters and must be enclosed in single quotation marks. For example 'A', '9', etc. C++ allows nongraphic characters which cannot be typed directly from keyboard, e.g., backspace, tab, carriage return etc. These characters can be represented by using a escape sequence. An escape sequence represents a single character. The following table gives a listing of common escape sequences.

Escape Sequence	Nongraphic Character
\a	Bell (beep)
\b	Backspace
\f	Formfeed
\n	Newline or line feed
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\?	Question mark
\\	Backslash
\'	Single quote
\"	Double quote
\xhh	Hexadecimal number (hh represents the number in hexadecimal)
\000	Octal number (00 represents the number in octal)
\0	Null

(iii) Floating constants

They are also called real constants. They are numbers having fractional parts. They may be written in fractional form or exponent form.

A real constant in fractional form consists of signed or unsigned digits including a decimal point between digits. For example 3.0, -17.0, -0.627 etc.

A real constant in exponent form has two parts: a mantissa and an exponent. The mantissa is either an integer or a real constant followed by letter E or e and the exponent which must be an integer. For example 2E03, 1.23E07.

(iv) String Literals

A sequence of character enclosed within double quotes is called a string literal. String literal is by default (automatically) added with a special character '\0' which denotes the end of the string. Therefore the size of the string is increased by one character. For example "COMPUTER" will be represented as "COMPUTER\0" in the memory and its size is 9 characters.

7.5.4 Punctuators

The following characters are used as punctuators in C++.

Brackets []	opening and closing brackets indicate single and multidimensional array subscript.
Parentheses ()	opening and closing brackets indicate functions calls, function parameters for grouping expressions etc.
Braces { }	opening and closing braces indicate the start and end of a compound statement.
Comma ,	it is used as a separator in a function argument list.
Semicolon ;	it is used as a statement terminator.
Colon :	it indicates a labelled statement or conditional operator symbol.
Asterisk *	it is used in pointer declaration or as multiplication operator.
Equal sign =	it is used as an assignment operator.
Pound sign #	it is used as pre-processor directive.

7.5.5 Operators

Operators are special symbols used for specific purposes.

C++ provides six types of operators.

- Arithmetical operators
- Relational operators
- Logical operators
- Unary operators
- Assignment operators
- Conditional operators
- Comma operator

Arithmetical operators

An operator that performs an arithmetic (numeric) operation: +, -, *, /, or %. For these operations always two or more than two operands are required. Therefore these operators are called binary operators. The following table shows the arithmetic operators.

Operators	Meaning	Example	Result
+	Addition	8 + 5	13
-	Subtraction	8 - 5	3
*	Multiplication	8 * 5	40
/	Division	8 / 5	1
%	Modulus/Remainder	8 % 5	3

Note : There is no operator which gives exponent in C++

Relational operators

The relational operators are used to test the relation between two values. All relational operators are binary operators and therefore require two operands. A relational expression returns zero when the relation is false and a non-zero when it is true. The following table shows the relational operators.

Relational operators	Meaning
<	Less than
< =	Less than or equal to
= =	Equal to
>	Greater than
> =	Greater than or equal to
! =	Not equal

Example 1

```
int x = 2; int l = 1;
int y = 3;
int z = 5;
```

The following statements are true

- (i) l == 1
- (ii) x < y
- (iii) z > y
- (iv) y >= 1
- (v) x != 1
- (vi) l <= 1

Logical operators

The logical operators are used to combine one or more relational expression. The following table shows the logical operators.

Operators	Meaning
	OR
&&	AND
!	NOT

The NOT operator is called the unary operator because it requires only one operand.

Example 2

```
int x = 5; int z = 9; int y = 7;
(x > y) && (z > y)
```

The first expression (x > y) evaluates to false and second expression (z > y) evaluates to true. Therefore, the final expression is false.

In AND operation, if any one of the expression is false, the entire expression is false.

In OR operation, if any one of the expression is true, the entire expression is true.

In NOT operation, only one expression is required.

If the expression is true, the NOT operation of true is false and vice versa.

Unary operators

C++ provides two unary operators for which only one variable is required.

For Example 3

```
a = - 50;      a = - b;  
a = + 50;      a = + b;
```

Here plus sign (+) and minus sign (-) are unary because they are not used between two variables.

Assignment operator

The assignment operator '=' stores the value of the expression on the right hand side of the equal sign to the operand on the left hand side.

Example 4

```
int m = 5, n = 7;  
int x, y, z;  
x = y = z = 0;
```

In addition to standard assignment operator shown above, C++ also support compound assignment operators.

C++ provides two special operators viz '+' and '-' for incrementing and decrementing the value of a variable by 1. The increment/decrement operator can be used with any type of variable but it cannot be used with any constant.

With the prefix version of these operators, C++ performs the increment or decrement operation before using the value of the operand. For instance the following code.

```
int sum, ctr;  
sum = 12;  
ctr = 4;  
  
sum = sum + (++ctr);
```

will produce the value of sum as 17 because ctr will be first incremented and then added to sum producing value 17.

Similarly, the following code

```
sum = 12;
ctr = 4;
```

```
sum = sum + ( - - ctr);
```

will produce the value of sum as 15 because ctr will be first decremented and then added to sum producing value 15.

With the postfix version of these operators, C++ first uses the value of the operand in evaluating the expression before incrementing or decrementing the operand's value. For example, the following code

```
sum = 12;
ctr = 4;
```

```
sum = sum + (ctr + +);
```

will produce the value of sum as 16 because ctr will be first used in the expression producing the value of sum as 16 and then increment the value of ctr by 1 (ctr becomes now 5)

Similarly, the following code

```
sum = 12;
ctr = 4;
```

`sum = sum + (ctr - -)` will produce the value of sum as 16 because ctr will be first used with its value 4 producing value of sum as 16 and then decrement the value of ctr by 1 (ctr becomes 3).

Let us study the use of compound assignment operators in the following table :

Operator	Example	Equivalent to
<code>+ =</code>	<code>A + = 2</code>	<code>A = A + 2</code>
<code>- =</code>	<code>A - = 2</code>	<code>A = A - 2</code>
<code>% =</code>	<code>A % = 2</code>	<code>A = A % 2</code>
<code>/ =</code>	<code>A / = 2</code>	<code>A = A / 2</code>
<code>^ =</code>	<code>A ^ = 2</code>	<code>A = A ^ 2</code>

Example 5

```
int x = 2; // first
```

```
x + = 5; // second
```

In the second statement, the value of x is 7

Conditional operator

The conditional operator ?: is called ternary operator as it requires three operands.

The format of the conditional operator is:

Conditional_expression ? expression1 : expression2;

If the value of conditional_expression is true then the expression1 is evaluated, otherwise expression2 is evaluated.

Example 6

```
int a = 5;
```

```
int b = 6;
```

```
big = (a > b) ? a : b;
```

The condition evaluates to false, therefore big gets the value from b and it becomes 6.

The comma operator

The comma operator gives left to right evaluation of expressions.

It enables to put more than one expression separated by comma on a single line.

Example 7

```
int i = 20, j = 25;
```

```
int sq = i * i, cube = j * j * j;
```

In the above statements, comma is used as a separator between two statements / expressions.

7.6 The sizeof operator

As we know that different types of variables, constant, etc. require different amounts of memory to store them. The sizeof operator can be used to find how many bytes are required for an object to store in memory.

For Example 8

```
sizeof (char) returns 1
```

```
sizeof (int) returns 2
```

```
sizeof (float) returns 4
```

if k is integer variable, the sizeof (k) returns 2.

The sizeof operator determines the amount of memory required for an object at compile time rather than at run time.

7.7 The order of Precedence

The order in which the Arithmetic operators (+, -, ¹, /, %) are used in a given expression is called the order of precedence. The following table shows the order of precedence.

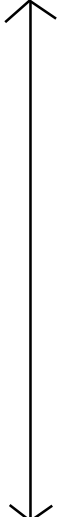
Order	Operators
First	()
Second	*, /, %
Third	+, -

The expression in the following example is calculated from left or right.

Example 9

$$\begin{aligned}
 & 5 + 4 / 3 * (5 - 3) / 2 \\
 = & 5 + 4 / 3 * 2 / 2 \\
 = & 5 + 1 * 2 / 2 \\
 = & 5 + 2 / 2 \\
 = & 5 + 1 \\
 = & 6
 \end{aligned}$$

The following table shows the precedence of operators.

++, -- (post increment / decrement)	Highest  Lowest
++ (Pre increment) -- (Pre decrement), sizeof	
(), ! (not), - (unary), + (unary)	
¹ , /, %	
+, -	
<, <=, >, >=	
=, !=	
&&	
? :	
=	
Comma operator	Lowest

In-Text Questions 7.1

1. Fill in the blanks
 - (a) A keyword is a word of C++.
 - (b) Pascal is a language whereas C++ is an language
 - (c) An identifier must not start with a
 - (d) constants are whole numbers without any fractional part.
 - (e) A sequence of characters enclosed within double quotes is called a literal.
 2. Mark True (T) or False (F) for the following:
 - (a) An identifier name can start with the underscore '- '.
 - (b) A token is a group of characters that logically belong together.
 - (c) Literals are data items that generally change their value during program execution.
 - (d) An escape sequence represents a single character.
 - (e) The symbol # is used as pre-processor directive.
 3. List out the various types of token used in C++.
 4. List out the data types supported by C++.
-

7.8 Type Conversion

The process in which one pre-defined type of expression is converted into another type is called conversion. There are two types of conversion in C++.

- Implicit conversion
- Explicit conversion

7.8.1 Implicit conversion

Data type can be mixed in the expression.

Example 10

```
double a;  
int b = 5;  
float c = 8.5;  
a = b * c;
```

When two operands of different type are encountered in the same expression, the lower type variable is converted to the higher type variable. The following table shows the order of data types.

Order of data types	
Data type	order
long double	(highest)
double	↑
float	↑
long	↑
int	↑
char	(lowest)

The int value of b is converted to type float and stored in a temporary variable before being multiplied by the float variable c. The result is then converted to double so that it can be assigned to the double variable a.

7.8.2 Explicit conversion

It is also called type casting. It temporarily changes a variable data type from its declared data type to a new one. It may be noted here that type casting can only be done on the right hand side of the assignment statement.

```
T_Pay = double (salary) + bonus;
```

Initially variable salary is defined as float but for the above calculation it is first converted to double data type and then added to the variable bonus.

7.9 Constants

A number which does not change its value during execution of a program is known as a constant. Any attempt to change the value of a constant will result in an error message. A constant in C++ can be of any of the basic data types. const qualifier can be used to declare constant as shown below:

```
const float Pi = 3.1215;
```

The above declaration means that Pi is a constant of float types having a value 3.1415.

Examples of valid constant declarations are:

```
const int rate = 50;
```

```
const float Pi = 3.1415;
```

```
const char ch = 'A';
```

7.10 Variables

A variable is the most fundamental aspect of any computer language. It is a location in the computer memory which can store data and is given a symbolic name for easy reference. The variables can be used to hold different values at different values at different times during the execution of a program.

To understand more clearly we should study the following statements:

Total = 20.00; ... (i)

Net = Total - 12.00; ... (ii)

In statement (i), a value 20.00 has been stored in a memory location Total. The variable Total is used in statement (ii) for the calculation of another variable Net. The point worth noting is that the variable Total is used in statement (ii) by its name not by its value.

Before a variable is used in a program, it has to be defined. This activity enables the compiler to make available the appropriate type of location in the memory. The definition of a variable consists of the type name followed by the name of the variable. For example, a variable Total of type float can be declared as shown below:

float Total;

Similarly the variable Net can also be defined as shown below:

float Net;

Examples of valid variable declarations are:

(i) int count;

(ii) int i, j, k;

(iii) char ch, first;

(iv) float total, Net;

(v) long int sal;

7.11 Input/Output (I/O)

C++ supports input/output statements which can be used to feed new data into the computer or obtain output on an output device such as: VDU, printer etc. It provides both formatted and unformatted stream I/O statements. The following C++ streams can be used for the input/output purpose.

Stream	Description
cin	console input
cout	console output

In addition to the above I/O streams, two operators << and >> are also used. The operator << is known as put to or bit wise shift operator. The operator >> is known as extraction or get from operator.

Let us consider a situation where the user desires to display a message “My first computer” on the screen. This can be achieved through the following statement:

```
cout << “My first computer”;
```

Once the above statement is carried out by the computer, the message “My first computer” will appear on the screen.

Similarly the following program segment defines a variable sum of integer type, initializes it to value 100 and displays the contents of the variable on the screen.

```
.  
. .  
. .  
    int sum;  
    sum = 100;  
cout << “The value of variable sum=”;  
    cout << sum;  
. . .
```

Once the above program segment is obeyed by the computer, the following output is displayed on the screen:

The value of variable sum = 100

From the above discussion we see that cout is the standard output stream of C++ by virtue of which output can be displayed on the screen. However the put to operator << is also required to hand over the contents of the variable to cout as shown in Fig. 7.2

cin is the standard input stream (keyboard) and it can be used to input a value entered by the user from the keyboard. However, the get from operator >> is also required to get the typed value from cin as shown in Fig. 7.3 and store it in the memory location.

Let us consider the following program segment:

```
.  
. .  
. .  
    int marks;
```

```
cin>> marks;
```

.
.

.

In the above segment, the user has defined a variable marks of integer type in the first statement and in the second statement he is trying to read a value from the keyboard. Once this set of statements is obeyed

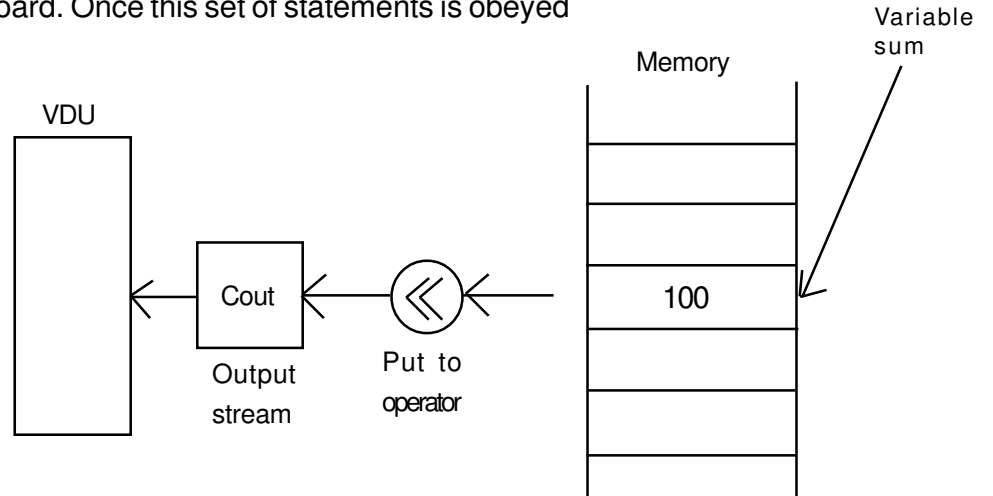


Fig. 7.2 Usage of cout and <<

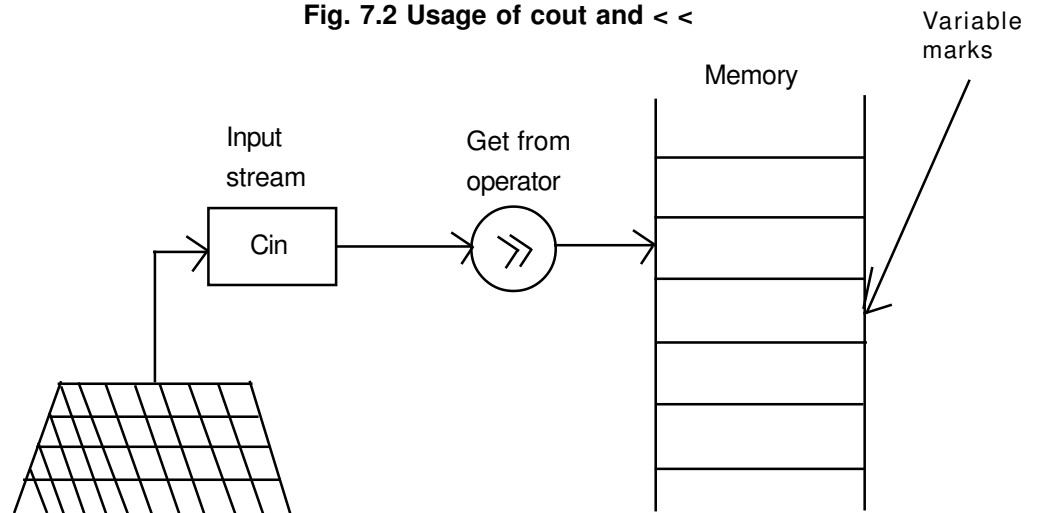


Fig 7.3 Usage of cin and >>

by the computer, whatever is typed on the keyboard (say 87) is received by the cin standard input stream. cin then hands over the typed value to get from operator >> which ultimately stores it in memory location called marks. The contents of this variable can be displayed on the screen by the statements given below:

```
cout << "marks=";
cout << marks;
```

We can use more than one output or put to operators within the same output statement as shown below. The activity is known as cascading of operators:

```
cout << "marks =" << marks;
```

The output of this statement will be

```
marks = 87
```

7.12 Structure of C++ Program

The structure of a C++ program is given below:

```
# Include <header file>
  main ( )
{
  .....
  .....
  .....
}
```

A C++ program starts with function called main (). The body of the function is enclosed between curly braces. These braces are equivalent to Pascal's BEGIN and END keywords. The program statements are written within the braces. Each statement must end by a semicolon (statement terminator).

A C++ program may contain as many functions as required. However, when the program is loaded in the memory, the control is handed over to function main () and it is the first function to be executed.

Let us now write our first program:

```
// This is may first program in C++
/* this program will illustrate different components of a simple program in
C++*/
# include <iostream.h>
void main ( )
{
cout <<"This is may first program in C++";
cout << "\n.....";
}
```

When the above program is compiled, linked and executed the following output is displayed on the VDU screen.

This is my first program in C++
.....

Various components of this program are discussed below:

(i) Comments

First three lines of the above program are comments and are ignored by the compiler. Comments are included in a program to make it more readable. If a comment is short and can be accommodated in a single line, then it is started with double slash sequence in the first line of the program. However, if there are multiple lines in a comment, it is enclosed between the two symbols `/*` and `*/`

Everything between `/*` and `*/` is ignored by the compiler.

(ii) Include `<iostream.h>`

The lines in the above program that start with symbol `#` are called directives and are instructions to the compiler. The word include with `#` tells the compiler to include the file `iostream.h` into the file of the above program. File `iostream.h` is a header file needed for input/output requirements of the program. Therefore, this file has been included at the top of the program.

(iii) void main ()

The word `main` is a function name. The brackets `()` with `main` tells that `main ()` is a function. The word `void` before `main ()` indicates that no value is being returned by the function `main ()`. Every C++ program consists of one or more functions. However, when program is loaded in the memory, the control is handed over to function `main ()` and it is the first function to be executed.

(iv) The curly brackets and body of the function main ()

A C++ program starts with function called `main ()`. The body of the function is enclosed between curly braces. These braces are equivalent to Pascal's `BEGIN` and `END` keywords. The program statements are written within the brackets. Each statement must end by a semicolon, without which an error message is generated.

Example 11

Write a C++ program that reads two values `x` and `y`, exchanges their contents and prints the output.

Solution: Let us assume `x` and `y` are of type `int`. We will use a variable `temp` of type `int` as a temporary location. The program is given below:

```
#include <iostream.h>
main ( )
```

```
{
int x, y temp;
// ... Read the values x and y

cout << "Enter the values:";
cin >> x >> y;
// ... Exchange the contents
temp = x;
x = y;
y = temp;
// display the output now
cout << "the exchanged values are:" << x << y;
}
```

In-Text Questions 7.2

1. Fill in the blanks:
 - (a) operators are used to test the relation between two values.
 - (b) C++ provides two operators for which one variable is required.
 - (c) The process in which one pre-defined type of expression is converted into another type is called.....
 - (d) A..... does not understand the contents of a comment and ignores it.
 - (e) The operator ++ is known as operator.
 2. Mark True (T) or False (F) in front of the following:
 - (a) The computer value of an arithmetic expression is always a numerical value.
 - (b) '\n' is the remainder operator.
 - (c) Arithmetic operations on characters is allowed in C++.
 - (d) the output of logical AND operation is true if one of its operand is true.
 - (e) += is a compound assignment operator.
 3. Evaluate the following expressions.
 - (a) $5/3 * 6$
 - (b) $6.0 * 5/3$
 - (c) $6 * 5/3$
-
-

7.13 What you have learnt

In this lesson you learnt the features of object oriented programming, particularly that of C++ language. We have explained the C++ character set, tokens which include variables, constants and operators. In C++, the interchanging of data takes place automatically or by the programmer (i.e., implicit & explicit call), which are given in Type Conversion. The concept of input/output statement has been explained. The concept of comment statement which makes the program more readable is also given. At the end, you should be able to write a C++ program which will take input from the user, manipulate and print it on the screen.

7.14 Terminal Questions

1. What is the difference between a keyword and an identifier?
 2. Explain the following terms:
 - (i) Literals
 - (ii) Implicit conversion
 3. How many ways are there in C++ to represent an integer constant?
 4. State the rules for comment statement?
 5. What is an Escape sequence. Describe any two backslash character constant?
 6. Explain briefly about standard Input/Output stream of C++.
 7. Write the equivalent C++ expression for the following Algebraic expression:
 - (i) $\frac{2AB + 2BC + 2CA}{2A}$
 - (ii) $\frac{4}{3} x^2$
 - (iii) $\frac{b^2 - 4ac}{2a}$
 8. Evaluate the following C++ expression:
Int a, b=2, k = 4;
 $a = b * 3/4 + k/4 + 8 - b + 5/8;$
-

9. What is the difference between:
- num = 10
- num = 10
10. Write an appropriate C++ statement for each of the following:
- (i) read the values of a, b and c
 - (ii) write the values of a and b in one line followed by the value of c on another line.
 - (iii) write the values of a and b in one line separated by blanks and value of c after two blank lines.
11. Write a program that will find out the square of a given number.

7.15 Feedback to In-Text Questions

In-text Questions 7.1

1. (a) reserved (b) structured, object-oriented
(c) digit (d) Integer (e) string
2. (a) T (b) T (c) F
(d) T (e) T
3. Various type of tokens used in C++ are:
- (i) Keywords
 - (ii) Identifiers
 - (iii) Literals
 - (iv) Punctuators
 - (v) Operators
4. Data types are:
- (i) Integer type (int)
 - (ii) Floating point type (float)
 - (iii) Character type (char)
-

In-text Questions 7.2

1. (a) Relational (b) Unary (c) Conversion
(d) Compiler (e) Increment
 2. (a) T (b) F (c) T
(d) F (e) T
 3. (a) 6 (b) 10.0 (c) 10
-

8

General Concept of OOP

8.1 Introduction

The object-oriented programming (OOP) is a different approach to programming. Object oriented technology supported by C++ is considered the latest technology in software development. It is regarded as the ultimate paradigm for the modelling of information, be that data or logic.

8.2 Objectives

After going through this lesson, you would be able to:

- learn the basic concepts used in OOP
- describe the various benefits provided by OOP
- explain the programming applications of OOP.

8.3 Object-Oriented Programming

The object-oriented programming is a different approach to programming. It has been created with a view to increase programmer's productivity by overcoming the weaknesses found in procedural programming approach. Over the years many object-oriented programming languages such as C++ and smalltalk have come up and are becoming quite popular in the market. The major need for developing such languages was to manage the ever-increasing size and complexity of programs.

8.4 Basic Concepts

The following are the basic concepts used in object-oriented programming.

- Objects
- Classes
- Data abstraction
- Modularity
- Inheritance
- Polymorphism

8.4.1 Objects

It can represent a person, a bank account or any item that a program can handle. When a program is executed, the objects interact by sending messages to one another. For example, if 'customer' and 'account' are two objects in a program, then the customer object may send message to account object requesting for a bank balance. Each object contains data and code to manipulate data. Objects can interact without having to know details of each other's data or code. It is sufficient to know the type of message accepted and the type of response returned by the objects.

8.4.2 Classes

We have just mentioned that objects contain data and function or code to manipulate that data. The entire set of data and code of an object can be made a user-defined data type with the help of a class. In fact objects are variables of type class. Once a class has been defined, we can create any number of objects associated with that class. For example, mango, apple and orange are members of class fruit. If fruit has been defined as a class, then the statement `fruit mango`, will create an object mango belonging to the class fruit.

8.4.3 Data Abstraction

Abstraction refers to the act of representing essential features without including the background details. To understand this concept more clearly, take an example of 'switch board'. You only press particular switches as per your requirement. You need not know the internal working of these switches. What is happening inside is hidden from you. This is abstraction, where you only know the essential things to operate on switch board without knowing the background details of switch board.

8.4.4 Data Encapsulation

Encapsulation is the most basic concept of OOP. It is the way of combining both data and the functions that operate on that data under a single unit. The only way to access the data is provided by the functions (that are combined along with the data). These functions are considered as member functions in C++. It is not possible to access the data directly. If you want to reach the data item in an object, you call a member function in the object. It will read the data item and return the value to you. The data is hidden, so it is considered as safe and far away from accidental alternation. Data and its functions are said to be encapsulated into a single entity.

8.4.5 Modularity

The act of partitioning a program into individual components is called modularity. It gives the following benefits.

- It reduces its complexity to some extent.
- It creates a number of well-defined, documented boundaries within the program.

Module is a separate unit in itself. It can be compiled independently though it has links with other modules. Modules work quite closely in order to achieve the program's goal.

8.4.6 Inheritance

It is the capability to define a new class in terms of an existing class. An existing class is known as a base class and the new class is known as derived class. Number of examples can be given on this aspect. For example, a motor cycle is a class in itself. It is also a member of two wheelers class. Two wheelers class in turn is a member of automotive class as shown in Fig. 8.1. The automotive is an example of base class and two wheelers is its derived class. In simple words, we can say a motor cycle is a two wheeler automotive.

C++ supports such hierarchical classification of classes. The main benefit from inheritance is that we can build a generic base class, i.e., obtain a new class by adding some new features to an existing class and so on. Every new class defined in that way consists of features of both the classes. Inheritance allows existing classes to be adapted to new application without the need for modification.

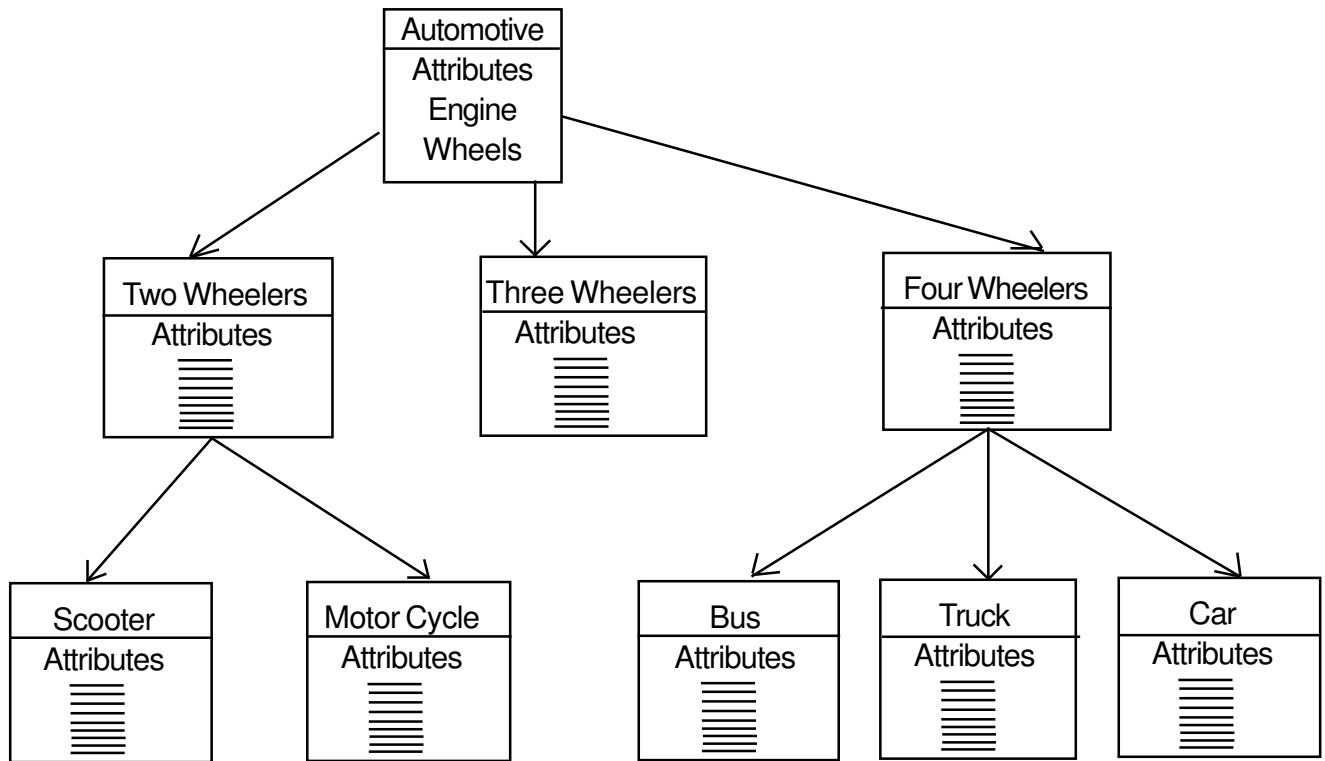


Fig. 8.1

8.4.7 Polymorphism

Polymorphism is a key to the power of OOP. It is the concept that supports the capability of data to be processed in more than one form. For example, an operation may exhibit different behaviour in different instances. The behaviour depends upon the types of data used in the operation. Let us consider the operation of addition. For two numbers, the operation will generate a sum. If the operands are strings then the operation would produce a third string by concatenation.

In-Text Questions 8.1

1. State True (T) or False (F) in case of the following.
 - (a) In procedure-oriented programming all data is shared by all functions.
 - (b) One of the striking features of OOP is division of program into objects that represent real world entities.
 - (c) Wrapping up of data of different types into a single unit is known as encapsulation.
-

- (d) Object oriented programs are executed much faster than conventional programs.
 - (e) Since C is a subset of C++, C programs will run under C++ compilers.
2. Why is data considered safe if encapsulated ?
 3. Differentiate between data hiding and encapsulation.
-

8.5 Benefits of OOP

OOP provides lot of benefits to both the program designer and the user. Object-oriented approach helps in solving many problems related to software development and quality of software product. The new technology assures greater programmer productivity, better quality of software and lesser maintenance cost. The major benefits are :

- Software complexity can be easily managed
- Object-oriented systems can be easily upgraded
- It is quite easy to partition the work in a project based on objects.

8.6 Programming Applications of OOP

OOP has become one of the programming buzzwords today. There appears to be a great deal of excitement and interest among software programmers in using OOP. Applications of OOP are gaining importance in many areas. OOP has been extensively used in the development of windows and word based systems such as MS-Windows, x-Windows etc. The promising application areas of OOP are:

- (i) Multiple data structure: This is an application where the same data structure is used many times. For example a window data structure is used multiple-times in a windowing system.
- (ii) Data in multiple programs: This is an application where the same operations are performed on a data structure in different programs. For example, record validation in an accounting system.

The other application areas of OOP are parallel programming, simulation and modelling, AI and Expert systems, Neural Networks and CAD systems.

In-Text Questions 8.2

1. Fill in the blanks.
 - (a) Small talk is anlanguage.
 - (b) An is a self contained unit of and function.
 - (c) A can be used to create objects of its own type.
 - (d) is a way of combining data with functions into an object.
 - (e) A derived class can be derived from a class.
 2. Mark True or False in front of the following :
 - (a) C++ is an extension to C programming language.
 - (b) Functions of one object cannot access the functions of other objects in C++
 - (c) Inheritance is capability to define a new class in terms of an existing class.
 - (d) Abstraction refers to the act of representing essential features without including the background details.
 - (e) An object is a new data type.
 3. Differentiate the following :
 - (a) Procedural and object-oriented programming
 - (b) An object and a class
-

8.7 What you have learnt

In this lesson you learnt about object-oriented technology, the latest technology in the software development. Various characteristics of OOP have been explained for the benefit of computer professionals. The promising application areas of OOP have also been described in brief.

8.8 Terminal Questions

1. What are the features of OOP ?
2. Explain the following terms briefly :
 - (a) Data abstraction
 - (b) Data encapsulation
 - (c) Polymorphism
 - (d) Inheritance
3. Describe briefly the various benefits of OOP?

8.9 Feedback to In-Text Questions

In-text Questions 8.1

1. (a) F (b) T
(c) F (d) F (e) T
2. Encapsulation ensures that only authorised functions access that relevant data. The measure against unauthorised access ensures data safety.
3. Data hiding is a property when the internal data structure of an object is hidden from the rest of the program. On the other hand, data encapsulation is the mechanism by which data and its associated functions are bound together within an object definition in such a way that only relevant information is exposed and rest remains hidden. Thus data encapsulation implements data hiding.

In-text Questions 8.2

1. (a) OOP (b) Object, data (c) class
(d) encapsulation (e) base
 2. (a) T (b) F (c) T
(d) T (e) F
 3. (a) A program in a procedural language is a list of instructions which tell the
-

computer to do something. The focus is on the processing, the algorithm needed to perform the desired computation. In procedural paradigm, the emphasis is on doing things. The object-oriented approach views a problem in terms of objects involved rather than procedure for doing it.

- (b) An object is an identifiable entity with some characteristics and behaviour. It represents an entity that can store data and its associated functions.

A class is a group of objects that share common properties and relationships.

9

Control Statements

9.1 Introduction

The normal flow of execution in a high level language is sequential, i.e., each statement is executed in the order of its appearance in the program. However, depending on the requirements of a problem it might be required to alter the normal sequence of execution in a program. The statements which specify the order of execution of statements are called control flow statements. There are many control flow statements provided in C++ that will be discussed in this lesson.

9.2 Objectives

After going through this lesson, you would be able to:

- differentiate between a statement and compound statement
- learn about the conditional statements
- familiarize with the loop statement
- differentiate between while and do-while loop
- identify the jump statements
- explain how to exit from the program

9.3 Statements

Statements are the instructions given to the computer to perform any kind of action. Action may be in the form of data movement, decision making etc. Statements form

the smallest executable unit within a C++ program. Statements are always terminated by semicolon.

9.4 Compound Statement

A compound statement is a grouping of statements in which each individual statement ends with a semi-colon. The group of statements is called block. Compound statements are enclosed between the pair of braces ({ }). The opening brace ({) signifies the beginning and closing brace (}) signifies the end of the block.

9.5 Null Statement

Writing only a semicolon indicates a null statement. Thus ';' is a null or empty statement. This is quite useful when the syntax of the language needs to specify a statement but the logic of the program does not need any statement. This statement is generally used in for and while looping statements.

9.6 Conditional Statements

Sometimes the program needs to be executed depending upon a particular condition. C++ provides the following statements for implementing the selection control structure.

- (i) 'if' statement
- (ii) 'if else' statement
- (iii) 'nested if' statement
- (iv) 'switch' statement

'if' statement

syntax of the 'if' statement

```
if (condition)
{
statement(s);
}
```

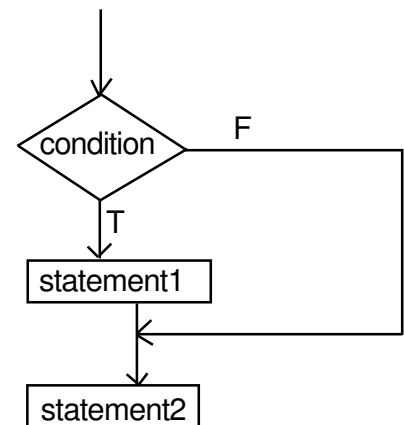


Fig. 9.1

From the flowchart it is clear that if the 'if condition' is true, statement 1 is executed; otherwise it is skipped. The statement 1 may either be a single or compound statement.

Example 1

'if else' statement

syntax of the if- else statement

if (condition)

statement1;

else

statement2;

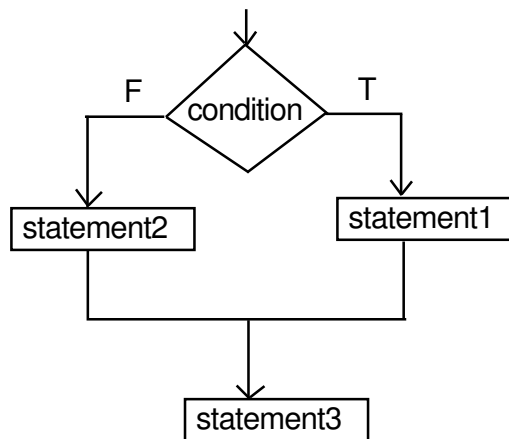


Fig. 9.2

From the above flowchart it is clear that the given condition is evaluated first. If the condition is true, statement1 is executed, followed by statement3. If the condition is false, statement2 is executed, followed by statement 3. It should be kept in mind that statement1 and statement2 can be single or compound statement.

Example 2

Write a program that will print the greatest of two numbers.

```

#include <iostream.h>
void main ()
{
int x, y;
cout << "Enter two numbers" << "\n";
cin >> x >> y ;
if ( x > y)
cout << "The bigger number is " << X;
else
cout << "The bigger number is" <<y;
}
  
```

Example 3

Write a program that will calculate the sum of odd numbers and sum of even numbers for numbers 1 to 10.

```
# include <iostream.h>
void main ()
{
int sum 1, sum2;
sum1 = sum2 = 0;
for (int i=1; i <= 10; i++)
if (i % 2 == 0)
sum 1 += i ;
else
sum2+= i ;
cout << "sum of odd number = " << sum 2 << "\n";
cout << "sum of even number = " << sum 1 << "\n";
}
```

Example 4

Write a program that will print the greatest of three numbers.

```
# include <iostream.h>
void main ()
{
int x, y, z, l;
cout << "Enter three numbers" << "\n";
cin >> x >> y >> z;
if (x > y )
l = x ;
else
l = y ;
if (l > z)
cout << "The greatest number is " << l;
else
cout << "The greatest number is " << z;
}
```

'Nested if' statement

Syntax of the nested if statement

- (i) The if block may be nested in another if or else block. This is called nesting of if or else block.

```

If (condition1)
{
    if (condition2)
    {
        statement(s);
    }
}

```

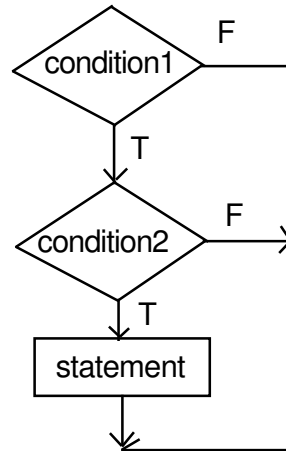


Fig. 9.3

- (ii) if (condition 1)
statement1;
else if (condition2)
statement2;
else statement 3;

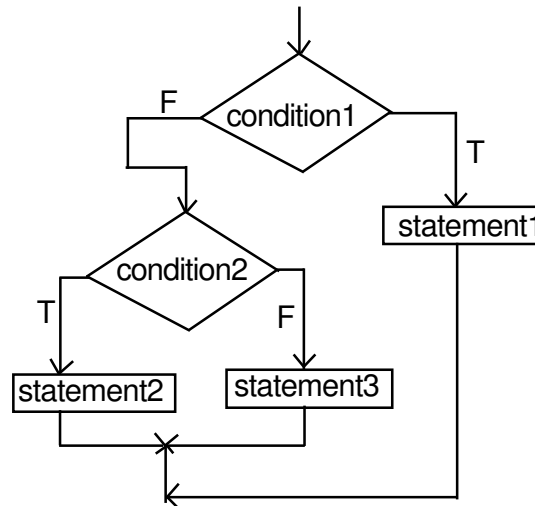


Fig. 9.4

From the flowchart it is clear that condition 1, condition2, etc. are nested in sequence until a condition is reached that evaluates to TRUE. Only one statement will be executed. If condition1 is TRUE, then statement1 is executed. If none of the conditions is TRUE, then statement 3 is executed. Regardless of which statement is executed, control is passed to the statement following statement3. Each statement can be single or compound statement.

Example 5

Write a program that will find the greatest of three numbers.

```

#include < isostream.h>
void main ( )

```

```
{
int x, y, z;
cout << "Enter three numbers" << "\n";
cin >> x >> y >> z ;
if ( x > y )
    {
    if (x >z)
    cout << "The greatest number is " << x;
    else
    cout << "The greatest number is " << z;
    }
else
    {
    if (y > z)
    cout << "The greatest number is " << y;
    else
    cout "The greatest number is " << z;
    }
}
```

Example 6

Write a program using nested if condition that will enter total marks and assign grades according to the following:

Mark	Grade
> = 90	A
> = 80 and < 90	B
> = 70 and < 80	C
> = 60 and < 70	D
< 60	Fail

```
#include <iostream.h>
void main ( )
{
int marks;
```

```
cout << "Enter total marks" << "\n";
cin >> marks;
if (marks >= 90)
cout << "A Grade";
else
if (marks >= 80)
cout << "B Grade";
else
if (marks >= 70)
cout << "C Grade";
else
if (marks >= 60)
cout << "D Grade";
else
Cout << "Failed";
}
```

switch statement

The if and if-else statements permit two way branching whereas switch statement permits multiple branching. The syntax of switch statement is:

```
switch (var / expression)
{
case constant1 : statement1;
                |
                |
                break;

case constant2 : statement2;
                |
                |
                break;

default :      statement3;
}
}
```

The execution of switch statement begins with the evaluation of expression. If the values of expression matches with the constant then the statements following this statement execute sequentially till it executes break. The break statement transfers control to the end of the switch statement. If the value of expression does not match with any constant, the statement with default is executed.

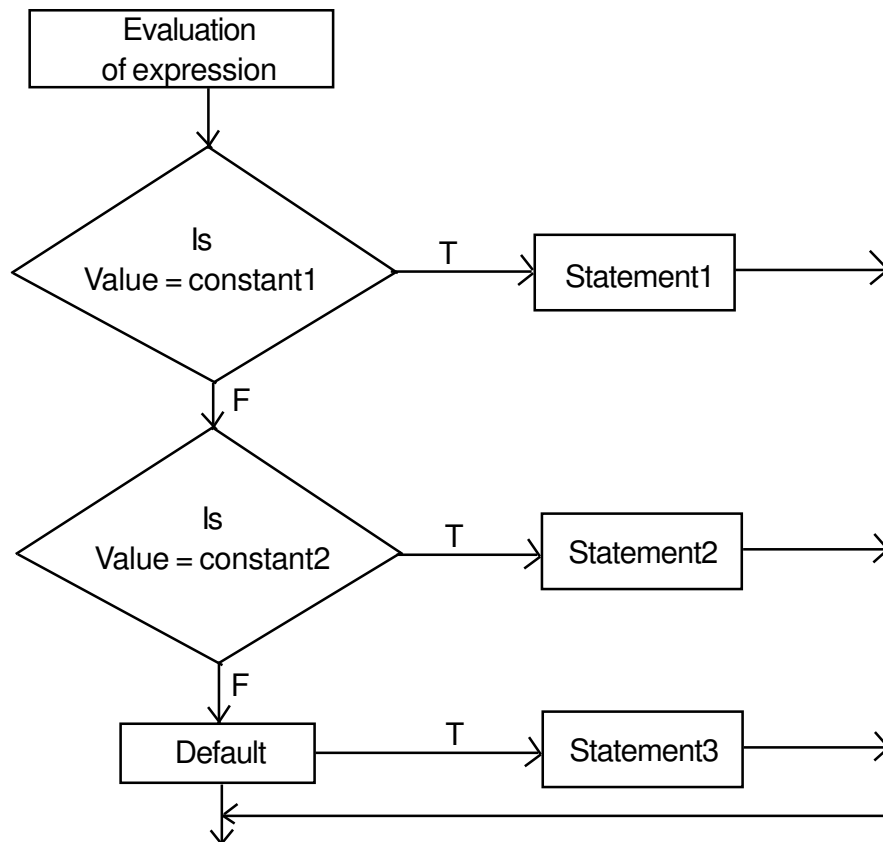


Fig.9.5

Example 7

```
#include <iostream.h>
void main ( )
{
char ch;
cout << "Enter a character";
cin >> ch;
switch (ch)
{
```

```
case 'a'   :   cout << "vowel a";
            break ;
case 'e'   :   cout << "vowel e";
            break ;
case 'o'   :   cout << "vowel o";
            break ;
case 'i'   :   cout << "vowel i"
            break ;
case 'u'   :   cout << "vowel u";
            break;
default    :   cout << "not a vowel";
}
}
```

The program needs the break statement to confine execution to particular portion of switch. If the break statement is removed from the above program then it starts executing the statements associated with case labels specified till it comes across either the break statement or end of switch statement. The above switch statement can be written as:

```
switch (ch)
{
    case 'a'   :
    case 'A'   :   cout << "vowel a";
    case 'e'   :
    case 'E'   :   cout << "vowel e";
    case 'i'   :
    case 'I'   :   cout << "vowel i";
                break;
    case 'o'   :
    case 'O'   :   cout << "vowel o";
                break;
    default    :   cout << "not a vowel";
}
}
```

The input character either in small letter or capital will be accepted.

Example 8

Write a program that will accept a one character grade code and depending on what grade code is input, display the Basic salary according to the table given below:

Grade	Basic salary
A	15000
B	12000
C	10000
D	9000

```
#include <iostream.h>
void main ( )
{
char grade;
cout << "Enter Grade" << "\n";
cin >> grade;
switch (grade)
{
case 'A' : cout << "Basic salary is 15000";
           break;
case 'B' : cout << "Basic salary is 12000;";
           break;
case 'C' : cout << "Basic salary is 10000";
           break;
case 'D' : cout << "Basic salary is 9000";
           break;
default  : cout << "Invalid grade code";
           }
}
```

In-Text Questions 9.1

1. What is a compound statement?
-

2. What is a selection statement? Write down the name of selection statements provided by C++.
3. What is the significance of default clause in switch statement?
4. Identify the error(s), if any, in the following programs and write them in correct format.

a)

```
#include <iostream.h>
void main ( )
int a, b;
cin << b; <<a;
if (a > b) Max = a
}
```

b)

```
#include <iostream.h>
void main( )
{
int x, y;
cin >> x ;
for (y = 0 ; y < 10, y ++ )
if x == y
cout << y + x ;
else
cout >> y;
}
```

5. What will be the output of the following program segments.

a)

```
cin >> x;
if ( x == 10 )
cout << "x is 10" ;
else
cout << " x is not 10" ;
```

if the input given is

- (i) 11
- (ii) 10

```
b) int year ;
    cin >> year ;
    if (year % 100 == 0)
    if (year % 400 == 0)
    cout << "Leap" ;
    else
    cout << "Not century year" ;
    if the input is
    (i) 2000
    (ii) 1971
```

```
c) int year ;
    cin >> year ;
    if (year % 100 == 0)
    {
    if (year % 400 == 0)
    cout << "Leap" ;
    }
    else
    cout << "Not century year" ;
    if the input given is
    (i) 2000
    (ii) 1971
```

6. Write equivalent switch statement for the following:

```
if (code == 'a')
cout << "season is summer" ;
else
if (code == 'r')
cout << "season is rainy" ;
else
if (code == 'w')
cout << "season is winter";
else
cout << "wrong code" ;
```

7. What would be the value of c?

```
{  
int c;  
float a, b;  
a = 245.05;  
b = 40.02 ;  
c = a + b;  
}
```

a) 285.09 b) 2850 c) 285 d) 285.0

8. What would be the value of i and k?

```
{  
int i, j, k ;  
j = 5 ;  
i = 2 * j / 2 ;  
k = 2 * (j/2) ;  
}
```

a) i=4, k=4 b) i=4, k=5 c) i=5, k=4 d) i=5, k=5

9. What is the output of the following program?

```
void main ( )  
{  
int x = 5, y = 6, z = 7;  
if (x < y + z)  
cout << "Hello \n" ;  
else  
{  
cout << "How \n" ;  
cout << "Are you \n";  
}  
}
```

a) Hello b) How c) Are you d) None of the above

10. Consider the program fragment.

```
switch (choice)
{
case 'W' : cout << "WHITE" ;
case 'R' : cout << "RED" ;
case 'B' : cout << "BLUE" ;
default  : cout << "error" ;
          break ;
}
```

What would be the output if choice = 'R'

- a) RED
- b) RED BLUE error
- c) RED error
- d) WHITE RED BLUE

9.7 Loop Construct

It is also called a Repetitive control structure. Sometimes we require a set of statements to be executed a number of times by changing the value of one or more variables each time to obtain a different result. This type of program execution is called looping. C++ provides the following constructs.

- (i) while loop
- (ii) do - while loop
- (iii) for loop

While loop

Syntax of while loop

```
while (condition)
{
statement(s);
}
```

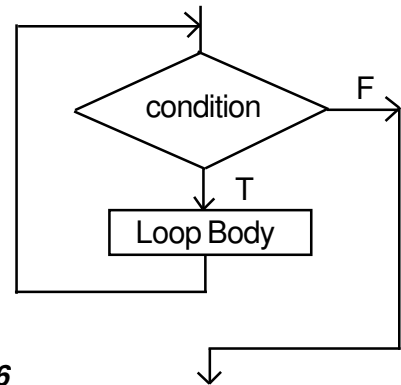


Fig. 9.6

The flow diagram indicates that a condition is first evaluated. If the condition is true, the loop body is executed and the condition is re-evaluated. Hence, the loop body is executed repeatedly as long as the condition remains true. As soon as the condition becomes false, it comes out of the loop and goes to the statement next to the 'while' loop. To make it more clear, we take the following example.

Example 9

To find the sum of first ten natural numbers i.e. 1, 2, 3,10.

```
# include <iostream.h>
void main ( )
{ int n, total = 0 ;
  n = 1 ;
  while (n <= 10)
  {
    total + = n ;
    n + + ;
  }
  cout << "sum of first ten natural number" << total :
}
```

The variable n is called a loop control variable since its value is used to control loop repetition. Normally, the three operations listed below must be performed on the loop control variable.

- (i) Initialize the loop control variable
- (ii) Test the loop control variable
- (iii) Update the loop control variable

Operation (i) must be performed before the loop is entered. Operation (ii) must be performed before each execution of the loop body; depending on the result of this test, the loop will either be repeated or make an exit. Operation (iii) must be included as part of the loop body. Unless the loop control variable is updated in the loop body, its value cannot change and loop exit will never occur.

do-while loop

Syntax of do-while loop

```
do  
{  
} while (condition);
```

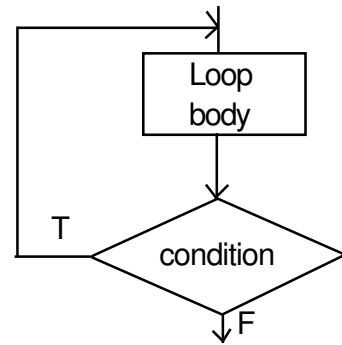


Fig. 9.7

The flow diagram indicates that after each execution of the loop body, the condition is true, the loop body is executed again. If the condition evaluates to false, loop exit occurs and the next program statement is executed.

Note : that the loop body is always executed at least once.

One important difference between the while loop and the do-while loop is the relative ordering of the conditional test and loop body execution. In the while loop, the loop repetition test is performed before each execution of the loop body; the loop body is not executed at all if the initial test fails. In the do-while loop, the loop termination test is Performed after each execution of the loop body; hence, the loop body is always executed at least once.

Let us take an example to explain it further.

Example 10

To find the sum of the first N natural number.

```
# include < iostream.h>  
void main ( )  
{  
int N, number, sum;  
cin >> N;  
sum = 0;  
number = 1;  
do  
{  
sum + = number;
```

```
    number ++ ;  
  }  
  while (number <= N) ;  
  cout << sum;  
}
```

For loop

It is a count controlled loop in the sense that the program knows in advance how many times the loop is to be executed.

syntax of for loop

```
for (initialization; decision; increment/decrement)
```

```
{  
  statement(s);  
}
```

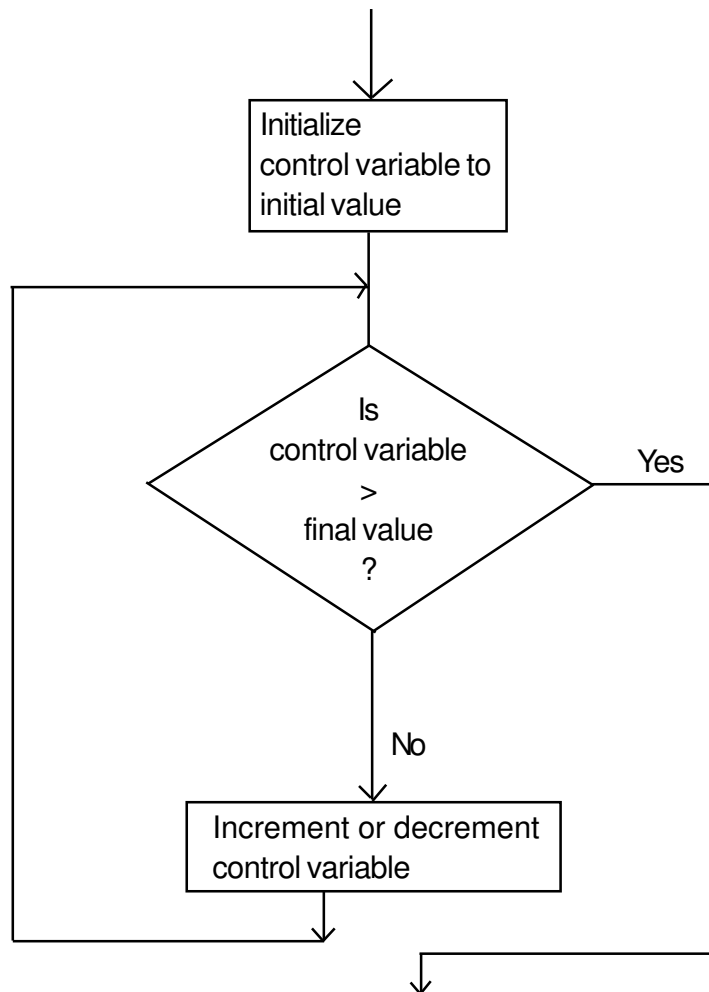


Fig. 9.8

The flow diagram indicates that in for loop three operations take place:

- (i) Initialization of loop control variable
- (ii) Testing of loop control variable
- (iii) Update the loop control variable either by incrementing or decrementing.

Operation (i) is used to initialize the value. On the other hand, operation (ii) is used to test whether the condition is true or false. If the condition is true, the program executes the body of the loop and then the value of loop control variable is updated. Again it checks the condition and so on. If the condition is true, it gets out of the loop.

Example 11

```
for (int i = 0 ; i < 5 ; i + + )  
cout << i ;
```

The output of the above program is

0 1 2 3 4

```
for (char i = 'A'; i < 'E' ; i + + )  
cout < < i ;
```

The output of the above program is

A B C D

In-Text Questions 9.2

- 1. What is the difference between while and do-while loops.
 - 2. Write a while loop that displays numbers 2, 4, 6, 8 12.
 - 3. Write a do-while loop that displays numbers 1, 3, 5, 7 11.
 - 4. Write a for-loop that displays numbers from 10 to 20.
 - 5. How many times is the following loop executed ?
 - a) `int s = 0, i = 0;`
-

```
while ( i + + < 5)
    s + = i;
```

```
b) int s = 0 ; i = 0;
    do
        s + = i ;
    while ( i < 5 ) ;
```

6. What will be the output of the following program ?

```
a) #include <iostream.h>
    void main ( )
    {
    long number = 7866372, result = 0 ;
    do
        {
        result * = 10 ;
        int digit = number % 10 ;
        result + = digit ;
        number / = 10 ;
        }
    while (number) ;
    cout << "output =" << result << endl;
    }
```

```
b) void main ( )
    {
    int i = 0, a = 0, b = 0, c = 0, f = 0 ;
    while ( i < = 5)
        {
        switch (i + +)
        {
        case 1 :
        case 2 : + + a ;
        case 3 :
        case 4 : + + b ;
```

```
        case 5 : + + c ;
        default : + + f ;
    }
    cout << a << "\n" ;
    cout << b << "\n" ;
    cout << c << "\n" ;
    cout << f << "\n" ;
}
```

7. What will be the value of counter after the following program is executed ?

```
void main ( )
{
    int counter ;
    int digit = 0 ;
    counter = 1 ;
    while (digit <= 10)
    {
        + + counter ;
        + + digit ;
    }
    cout << counter ;
}
```

a) 9 b) 10 c) 11 d) 12

8. What will be the value of sum after the following program is executed ?

```
void main ( )
{
    int sum, i ;
    sum = 1 ;
    i = 9 ;
    do
    {
        i = i - 1 ;
        sum = 2 * sum ;
    }
```

```
    }  
    while (i > 9) ;  
    cout << sum ;  
    }
```

- a) 1 b) 2 c) 9 d) 0.5

9. What is the following program doing ?

```
void main ( )  
{  
int digit = 0 ;  
do  
{  
cout << digit + + << "\n" ;  
}  
while (digit <= 10);  
}
```

- a) Display integers from 1 to 10
b) Display integers from 0 to 10
c) Adding 10 integers
d) None of the above

10. What is the final value of digit ?

```
void main ( )  
{  
int digit ;  
for (digit = 0 ; digit <= 9 ; digit + + )  
cout << digit << "\n" ;  
digit = 5 * digit ;  
-- digit ;  
}
```

- a) 49 b) -1 c) 47 d) 46
-

9.8 Jump Statements

The jump statements unconditionally transfer program control within a function.

- a) goto statement
- b) break statement
- c) continue statement

goto statement

syntax of goto statement

```
    goto pgr;  
    |  
    |  
    |  
pgr :
```

pgr is known as label. It is a user defined identifier. After the execution of goto statement, the control transfers to the line after label pgr.

Note : It is not a good programming to use goto statement in a program.

Break statement

syntax of break statement

The break statement can be used in a switch statement and in any of the loops. It causes program execution to pass to the next statement following the switch or the loop.


```
    while (condition)  
    {  
    statement 1;  
    if (condition)  
    break ;  
    statement 2;  
    }  
    → statement 3;
```

The break statement skips rest of the loop and goes out of the loop.

continue statement

The continue statement is used in loops and causes a program to skip the rest of the body of the loop.

```
while (condition) ←  
{  
  statement 1;  
  If (condition)  
  continue ;  
  statement 2;  
}  
statement 3;
```



The continue statement skips rest of the loop body and starts a new iteration.

9.9 exit () function

The execution of a program can be stopped at any point with exit () and a status code can be informed to the calling program. The general format is

```
exit (code) ;
```

where code is an integer value. The code has a value 0 for correct execution. The value of the code varies depending upon the operating system. It requires a process.h header file.

In-Text Questions 9.3

1. The goto statement causes control to go to
 - (i) an operation
 - (ii) a label
 - (iii) a variable
 - (iv) a function

 2. The break statement causes an exit
 - i) only from the innermost loop
-

- ii) only from the innermost switch
 - iii) from all loops and switches
 - iv) from the innermost loop or switch
3. A continue statement within a loop causes control to go to _____.
4. Which header file must be included in the program for using the exit () function.
5. Rewrite the following program segment using goto statement i = 100 ;
- ```
while (i)
cout << i -- ;
cout << "\n Thank you" ;
```
6. Rewrite the following program segment using a while loop.
- ```
i = 2 ;
start :
cout << i ;
i + = 2 ;
if ( i < 51) goto start ;
cout << "\n Thank you" ;
```
7. What will be the output of following program ?
- ```
#include < iostream.h>
void main ()
int x = 10, count = 0 ;
while (x)
{
x - - ;
if (x == 4)
continue ;
count + + ;
}
cout << "\n" <<count ;
}
```
-

```
8. #include <iostream.h>
void main ()
{
char ch = 'A' ;
while (ch <= 'F')
{
switch (ch)
{
case 'A' :
case 'B' :
case 'C' :
case 'D' : ch ++ ; continue ;
case 'E' :
case 'F' : ch ++ ;
}
putchar (ch) ;
}
}
```

What will be the output of the above program ?

- a) ABCDEF    b) FG    c) EFG    d) None of the above
- 

## 9.10 What you have learnt

In this lesson you learnt the statements which control the execution of a program. Now you should be in a position to write small programs in which flow control statements are used. You have learnt the meaning of opening braces '{' and closing braces '}'.

---

## 9.11 Terminal Questions

---

1. Write a program to display the following menu and accept a choice number. If an invalid choice is entered, appropriate error message must be displayed. Otherwise the choice number entered must be displayed.

MENU

---



1. Create a data file
2. Display a data file
3. Edit a data file
4. Exit

Choice :

2. Write a program that will accept a mark and assign letter grade according to the following table.

| Grade | Mark                 |
|-------|----------------------|
| A     | $\geq 90$            |
| B     | $\geq 80$ but $< 90$ |
| C     | $\geq 70$ but $< 80$ |
| D     | $\geq 60$ but $< 70$ |
| F     | $< 60$               |

3. Write a program that will print sum of N integers entered by the user.
4. Write a program to generate the members of fibonacci series upto 500.
5. Write a program to reverse a given number.
6. Write a program that will print the table of any number upto any number.

input :

Enter the table of which number : 4

Enter upto which number : 5

Output

$4 \times 1 = 4$

$4 \times 2 = 8$

$4 \times 3 = 12$

$4 \times 4 = 16$

$4 \times 5 = 20$

Continue (Y/N) ?

7. Write a program that will print the factorial of any number.

input : 5

Output :  $5 \times 4 \times 3 \times 2 \times 1 = 120$

---

8. Write a program to print all the tables between 2 and 20 upto 10.
9. Write a program that will find pythagorean triplets between 1 to 100. (Pythagorean triplets are such that the square of one number is equal to the sum of the squares of the other two numbers).

Example : 3, 4, 5

$$5^2 = 3^2 + 4^2$$

$$25 = 9 + 16 = 25$$

10. Determine the roots of the quadratic equation

$$ax^2 + bx + c = 0$$

using the formula  $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

11. Write a program to find out the sum of the series

$$1 + x + x^2 + x^3 + \dots + x^n$$

12. Write a program that will take a number N as input and print the factorial of all numbers from 1 to N as follows :

| Number | Factorial |
|--------|-----------|
| 1      | 1         |
| 2      | 2         |
| 3      | 6         |
|        |           |
|        |           |
|        |           |
| N      | N!        |

13. Write a program that will find out whether the given number is even or odd. If odd number then whether it is prime or not.
14. Write a program that will ask the user to enter N integers. Find out the following:
- Total number of positive integers
  - Total number of negative integers
  - Total number of integers equal zero

## 9.12 Feedback to In-Text Questions

---

### In-text Questions 9.1

1. Compound statement - Group of statements enclosed by a pair of braces ( { } ).
  2. Selection statement - statement that allows us to choose a set of instructions for execution, depending upon the expression. Examples If, If-else switch.
  3. If none of the constant present in case labels is equal to the value of the variable or expression in switch statement, default label is executed.
  4.
    - a) 

```
#include <iostream.h>
void main ()
{
 int a, b, max;
 cin >> b >> a;
 if (a > b) Max = a;
}
```
    - b) 

```
#include <iostream.h>
void main ()
{
 int x, y;
 cin >> x;
 for (y = 0 ; y < 10 ; y + +)
 if (x == y)
 cout << y + x;
 else
 cout << y;
}
```
  5.
    - a)
      - (i) x is not 10
      - (ii) x is 10
    - b)
      - (i) Leap
      - (ii) No output
-

- c) (i) Leap
- (ii) Not century year

## 6. Switch (code)

```
{
case 'a' : cout << "season is summer" ; break;
case 'r' : cout << "season is rainy"; break;
case 'w' : cout << "season is winter"; break ;
default : cout << "wrong code";
```

- 7. c)
- 8. c)
- 9. Hello
- 10. b)

**In-text Questions 9.2**

1. The while loop checks the condition first and then executes the loop whereas do-while loop executes the loop first and then checks the condition so it is executed at least once even if the condition is false.
  2. 

```
int i = 2 ;
while (i <= 12)
{
cout << i ;
i += 2 ;
}
```
  3. 

```
int i = 1 ;
do
{
cout << i ;
i += 2 ;
} while (i <= 11);
```
  4. 

```
for (int i = 10; i <= 20 ; i ++)
cout << i ;
```
-

5. a) 5 times  
b) endless
6. a) output = 2 7 3 6 6 8 7  
b) 2  
4  
5  
6
7. d)
8. b)
9. b)
10. a)

### In-text Questions 9.3

1. (ii) 2. (iv)
  3. the decision statement of loop
  4. process.h
  5. 

```
i = 100 ;
xyz : i -- ;
if i > 0 goto xyz ;
cout << "\n Thank you" ;
```
  6. 

```
i = 2 ;
while (i < 51)
{
cout << i ;
i += 2, ;
}
out << "\n Thank you" ;
```
  7. 9
  8. (b)
-

# 10

## Functions

### 10.1 Introduction

Functions in C ++ are available under C++ library header files. It provides declarations of the function prototypes, class definition, etc. For example,

```
double sqrt (float);
```

The above declaration specifies that the sqrt function takes only one argument of type float and return double type. To make use of predefined functions, the header file (that contains the functions prototypes) must be included in a program using a preprocessor directive # include.

### 10.2 Objectives

After going through this lesson, you would be able to:

- familiarize with the use of # include directive and its requirement in the program
- define functions included in math.h, ctype.h and string.h header files
- explain character handling functions which help in input/output operations
- define inline function and function with default value

### 10.3 # Include Directive

The # include directive instructs the compiler to read and include another file in the current file. The compiler compiles the entire code.

---

A header file may be included in one of two ways.

```
include <iostream.h>
```

or

```
include "iostream.h"
```

The header file in angle brackets <and> means that file reside in standard include directory. The header file in double quotes means that file reside in current directory.

## 10.4 Library Functions

C++ provides many built in functions that saves the programming time.

### Mathematical Functions

Mathematical functions like  $\sin ( )$ ,  $\cos ( )$ , etc. are defined in header file math.h. Some of the important mathematical functions are given in the following table.

Some standard mathematical functions

| Function           | Meaning                                              |
|--------------------|------------------------------------------------------|
| $\sin(x)$          | Sine of an angle $x$ (measured in radians)           |
| $\cos(x)$          | Cosine of an angle $x$ (measured in radians)         |
| $\tan(x)$          | Tangent of an angle $x$ (measured in radians)        |
| $\text{asin}(x)$   | $\text{Sin}^{-1}(x)$ where $x$ (measured in radians) |
| $\text{acos}(x)$   | $\text{Cos}^{-1}(x)$ where $x$ (measured in radians) |
| $\text{exp}(x)$    | Exponential function of $x$ ( $e^x$ )                |
| $\log(x)$          | logarithm of $x$                                     |
| $\log_{10}(x)$     | Logarithm of number $x$ to the base 10               |
| $\text{sqrt}(x)$   | Square root of $x$                                   |
| $\text{pow}(x, y)$ | $x$ raised to the power $y$                          |
| $\text{abs}(x)$    | Absolute value of integer number $x$                 |
| $\text{fabs}(x)$   | Absolute value of real number $x$                    |

---

## Character Functions

All the character functions require ctype.h header file. The following table lists the function.

| String     | Functions                                                                                                                    |
|------------|------------------------------------------------------------------------------------------------------------------------------|
| isalpha(c) | It returns True if c is a Letter (A to Z or a to z) otherwise False.                                                         |
| isdigit(c) | It returns True if c is a digit (0 through 9) otherwise False.                                                               |
| isalnum(c) | It returns True if c is a digit from 0 through 9 or an alphabetic character (either uppercase or lowercase) otherwise False. |
| islower(c) | It returns True if c is a lowercase letter otherwise False.                                                                  |
| isupper(c) | It returns True if c is an uppercase letter otherwise False.                                                                 |
| toupper(c) | It converts c to uppercase letter.                                                                                           |
| tolower(c) | It converts c to lowercase letter.                                                                                           |

## String Functions

The string functions are present in the string.h header file. Some string functions are given below:

|                                                                      |                                                                                                                   |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| strlen(S)                                                            | It gives the no. of characters including spaces present in a string S.                                            |
| strcat(S1, S2)                                                       | It concatenates the string S2 onto the end of the string S1. The string S1 must have enough locations to hold S2. |
| strcpy(S1, S2)                                                       | It copies character string S2 to string S1. The S1 must have enough storage locations to hold S2.                 |
| strcmp((S1, S2) = 0)<br>strcmp((S1, S2) > 0)<br>strcmp((S1, S2) < 0) | It compares S1 and S2 and finds out whether S1 equal to S2, S1 greater than S2 or S1 less than S2.                |



## Console I/O functions

The following are the list of functions

- (i) getchar ( )
- (ii) putchar ( )
- (iii) gets ( )
- (iv) puts ( )

The header file for above functions is stdio.h. The first two functions deal with single character and last two functions deal with string (group of characters).

### Getchar ( ) function

The getchar ( ) function returns a single character from a standard input device (keyboard). It takes no parameter and the returned value is the input character. The general form of the getchar ( ) function is :

A = getchar ( ) ;

The variable A is of the type character. It inputs a character from the keyboard and assign it to variable A.

### putchar ( ) function

The putchar ( ) function takes one argument, which is the character to be sent to output device. It also returns this character as a result. The general form of the putchar ( ) function is :

putchar (ch) ;

where ch is a variable of type character.

#### **Example 1**

```
include < iostream.h >
include < stdio.h>
void main ()
{ char ch;
 ch = getchar () ;
 putchar (ch);
}
```

---

The above program takes a character from the keyboard and prints it on the screen.

## gets ( ) function

The gets ( ) function gets a string terminated by a newline character from the standard input stream stdin. The gets ( ) replaces the newline by a null character (\0). It also allows input string to contain white space characters (spaces, tabs), gets return when it encounters a newline; everything upto the newline is copied into A.

```
gets (A) ;
```

### Example 2

```
include < stdio.h >
include < iostream.h >
void main ()
{
char A [100];
cout << "Input a string";
gets (A);
puts (A);
}
```

## Some More Functions

### The getch ( ) and getche ( ) functions

The general form of the getch ( ) and getche ( ) is

```
ch = getche () ;
ch1 = getch () ;
```

ch and ch1 are the variables of type character. They take no argument and require the conio.h header file. On execution, the cursor blinks, the user must type a character. The value of the character returned from getche ( ) is assigned to ch. The getche ( ) function echoes the character to the screen. That's why there's an e in getche. Another function, getch ( ), is similar to getche ( ) but does not echo the character to the screen.

### Example 3

To count the no. of characters and words in a sentence entered by the user.

---

```
include <iostream.h >
include <conio.h >
void main ()
{
char ch;
int chcnt = 0, wdcnt = 1;
while ((ch = getch ()) != '\r')
{
if (ch == ' ')
wdcnt + + ;
else
chcnt + + ;
}
cout << "No. of characters" << chcnt << "\n"
cout << "No. of words" << wdcnt;
}
```

The above program counts the no. of characters and no. of words in a sentence terminated by return key. The '\r' reads the return key character.

---

### **In-Text Questions 10.1**

---

1. What is a header file ? What is its purpose ?
  2. Why it is necessary to include a header file in a program ?
  3. Name the header file that is required for the following :
    - (i) abs ( )
    - (ii) strcmp ( )
    - (iii) clrscr ( )
    - (iv) tolower ( )
  4. What is a library function ?
  5. Differentiate between getch ( ) and getch ( ) functions.
-

6. What will be the output of the following programs.

```
(a) # include < iostream.h >
 # include < stdio.h>
 void main ()
 {
 char name [100];
 puts ("enter your name ");
 gets (name);
 puts ("Hello");
 puts (name) ;
 }
```

Input : MOHIT

```
(b) # include < iostream.h>
 # include < stdio.h>
 void main ()
 {
 char c;
 c = getchar ();
 while (c != '.')
 {
 putchar (c);
 c = getchar ();
 }
```

Input : RUBY

7. Fill in the blanks :

- a) All header files have an extension .....
- b) Character functions are available in .....
- c) The function toupper (C) converts C to .....
- d) In strcpy (S1, S2) the string..... copies onto the string.....
- e) The function which counts the length of the string is .....
- f) The sin ( ) function is available in .....

8. State whether the following are True or False.
- a) The `isdigit ( )` function checks whether the entered character is digit or not.
  - b) All string functions are available in `stdio.h`.
  - c) The `iostream.h` contains all input/output functions.
  - d) The `tolower (C)` converts C to lowercase.
  - e) `strcat (S1, S2)` splits two strings into three strings.
  - f) `pow (x, y)` calculates y to the power x.
  - g) `cos (x)` returns the cosine of angle (x).
  - h) `tolower ( )` function is available in `ctype.h`.
  - i) The `log (x)` is available in `string.h`.
- 

## 10.5 User defined C ++ function

A C++ function is a grouping of program statements in a single unit. The `main ( )` is an essential function in all C ++ programs. If there is a single function in a program, it should be named `main ( )`. The `main ( )` function is the starting point for the execution of a program. The definition of `main ( )` would look like as follows :

```
main ()
{
// main program statements
}
```

In C ++, the `main ( )` returns a value of type `int` to the operating system. The function that have a return value should use the `return` statement for termination. The `main ( )` function, therefore, is defined as follows :

```
int main ()
{

return (0);
}
```

---

Since the return type of function is int by default, the keyword int in main ( ) header is optional.

Consider the program given below :

```
include < iostream. h >
void main ()
{
cout << "Computer Science \n";
}
```

It prints computer science on the screen. We can also write the above program in the following way:

```
include < iostream.h>
void message ();
void main ()
{
message () ;
}
void message ()
{
cout << "Computer Science\n";
}
```

The above program also prints computer science but it uses a function message for printing. In addition to functions main ( ) and message ( ), the program has an extra line after # include preprocessor directive. The line is void message ( ). This is known as function prototype.

### **10.5.1 Function Prototype**

The function prototype informs the compiler about the functions to be used in a program, the argument they take and the type of value they return. Functions which do not return any value are known as void functions.

### **10.5.2 Arguments to a function**

Sometimes the calling function supplies some values to the called function. These are known as parameters. The variables which supply the values to a calling function

---

called actual parameters. The variable which receive the value from called statement are termed formal parameters.

Consider the following example that evaluates the area of a circle.

```
include < iostream.h >
void area (float);
void main ()
{
 float radius;
 cin >> radius;
 area (radius) ;
}
void area (float r)
{
 cout << "The area of the circle is " << 3.1416 *r* r << "\n";
}
```

The statement `area (radius)`, that invokes the function, may be written as

```
area (6.0);
```

This will evaluate and display the area of a circle with radius 6.0

Here `radius` is called actual parameter and `r` is called formal parameter.

### 10.5.3 Return Type of a function

In the above program, area of a circle display in the function `area ( )` itself. If the area of a circle required in the calling function `main ( )`, the function `area ( )` may return the area evaluated to function `main ( )`. This is possible through the return statement. The program may be written as :

```
include < iostream. h >
float area (float);
void main ()
{
 float radius, y;
 cin >> radius ;
 y = area (radius)
```

---

```
 cout << "The area of the circle is " << y;
 }
 float area (float r)
 {
 return (3.1416 *r* r);
 }
```

In function prototype, void is replaced by float, this indicates that value returned by the function area to main function is float. In the calling function variable y is assigned the value returned by function area (. Thus, variable y will be assigned the area of a circle.

### 10.5.4 Global and Local variables

The variable declared in a program may broadly be classified into following two types;

- (i) Local variable
- (ii) Global variable

#### Local variable

A variable declared within the body of a function will be evaluated only within the function. The portion of the program in which a variable is retained in memory is known as the scope of the variable. The scope of the local variable is a function where it is defined. A variable may be local to function or compound statement.

#### Global variable

A variable that is declared outside any function is known as a global variable. The scope of such a variable extends till the end of the program. These variables are available to all functions which follow their declaration. So it should be defined at the beginning, before any function is defined. If in a program, variable a is declared as local as well as global.

Then        a represents the local variable

      :: a represents the global variable.

      The symbol :: is called scope resolution operator

#### Example 4

```
include < iostream.h >
```

---



```
int m = 2;
void main ()
{
int m = 15;
{
int m = 10 * :: m;
cout << "m = " << m << "\n";
cout << " :: m = " << :: m << "\n"
}
cout << "m = " << m << "\n";
cout << " : : m = " << : : m << "\n";
}
```

The output of the above program is

```
m= 20
:: m = 2
m = 15
: : m = 2
```

### **Variables and Storage class**

The storage class of a variable determines which parts of a program can access it and how long it stays in existence. The storage class can be classified as (i) Automatic (ii) Register (iii) Static (iv) External.

#### **Automatic variable**

All variables by default are auto , i.e., the declarations `int a` and `auto int a` are equivalent. Auto variables retain their scope till the end of the function in which they are defined. An automatic variable is not created until the function in which it is defined is called. When the function exits and control is returned to the calling program, the variables are destroyed and their values are lost. The name automatic is used because the variables are automatically created when a function is called and automatically destroyed when it returns.

#### **Example 5**

```
#include <iostream.h >
void sum (int);
void main ()
```

---

```
{
 int i;
 for (i = 1; i <=5; i++)
 sum (i);
}
void sum (int n)
{
 auto int s = 0;
 s = s + n;
 cout << s << "\n";
}
```

The output of the program is

```
1
2
3
4
5
```

### **Register variable**

A register declaration is an auto declaration. A register variable has all the characteristics of an auto variable. The difference is that register variable provides fast access as they are stored inside CPU registers rather than in memory. The register and auto can be applied to local variable.

### **Static variable**

The variable may be static automatic variable or static external variable. The later one is meaningful only in multifile programs. A static automatic variable has the visibility of a local variable but the lifetime of an external variable. Thus it is visible only inside the function in which it is defined, but it remains in existence for the life of the program. If in the previous program auto int s = 0 is changed to static int s = 0. The output of the program will be

```
1
3
6
10
15
```

---

### External variable

A large program may be written by a number of persons in different files. A variable declared global in one file will not be available to a function in another file. Such a variable, if required by functions in both the files, should be declared global in one file and at the same time declared external in the second file.

Example : extern int a;

Where extern is the keyword for external.

### 10.5.5 Calling of Function

The function can be called using either of the following methods:

- (i) call-by-value
- (ii) call-by-reference

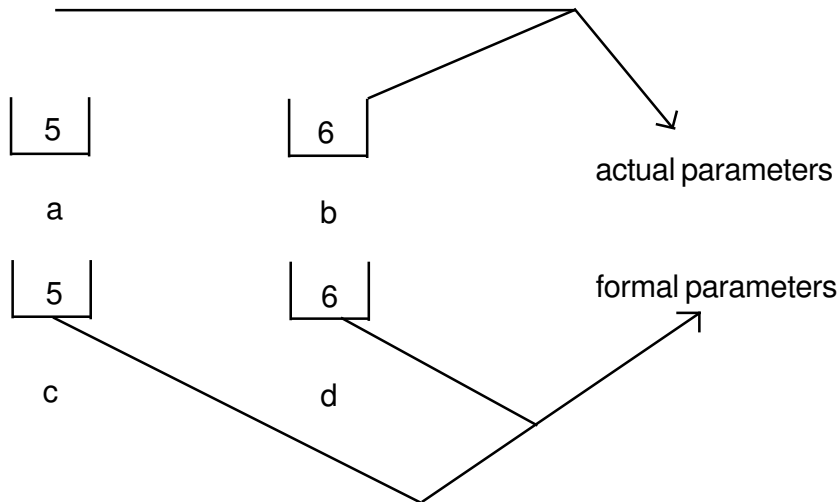
#### Call-by-value

Consider the following program which will swap the value of two variables using the method call-by-value.

```
#include <iostream.h>
void swap (int, int);
void main ()
{
 int a, b;
 cin >> a >> b;
 swap (a, b);
 cout << a << b << "\n";
}
void swap (int c, int d)
{
 int t;
 t = c;
 c = d;
 d = t;
 cout << c << d << "\n";
}
```

---

The variables a and b are called actual parameters. When calling the function swap ( ) the value of a is passed onto c and value of b is passed onto d. Here c and d are called formal parameters.



In the function swap ( ), the value are swapped but this change will not be implemented to actual parameters.

|        |   |   |
|--------|---|---|
| Input  | 5 | 6 |
| Output | 6 | 5 |
|        | 5 | 6 |

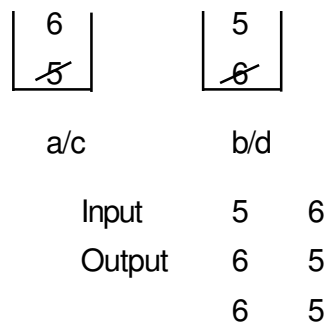
### Call-by-reference

Consider the following program which will swap the value of two variables using call-by-reference method.

```
include < iostream.h >
void swap (int &, int &);
void main ()
{
 int a , b;
 cin > > a > > b;
 swap (a, b);
 cout << a << b << "\n";
}
void swap (int &c, int &d)
```

```
{
int t;
t = c;
c = d;
d = t;
cout << c << d << "\n";
}
```

In this method, when the function is called, the address is transferred from actual to formal parameters. It means a and c are sharing the same memory address and b and d are sharing the same memory address.



So, any change in formal parameter will be implemented to actual parameter.

## 10.6 Inline Function

Functions save memory space because all the calls to the function cause the same code to be executed. The functions body need not be duplicated in memory. When the compiler sees a function call, it normally jumps to the function. At the end of the function, it jumps back to the statement following the call.

While the sequence of events may save memory space, it takes some extra time. To save execution time in short functions, inline function is used. Each time there is a function call, the actual code from the function is inserted instead of a jump to the function. The inline function is used only for shorter code.

### **Example 6**

```
#include <iostream.h >
inline int cube (int r)
{
```

---

```
 return (r*r*r);
 }
 void main ()
 {
 int number, y;
 cout << "Enter number";
 cin >> number;
 y = cube (number);
 cout << "The cube of the number is" << y;
 }
```

Some important points to be noted :

- (i) Function is made inline by putting a word inline in the beginning.
- (ii) Inline function should be declared before main ( ) function.
- (iii) It does not have function prototype.
- (iv) Only shorter code is used in inline function.
- (v) If longer code is made inline then compiler ignores the request and it will be executed as normal function.

## 10.7 Function with Default Arguments

C++ allows to call a function without specifying all its arguments. In such cases, the function assigns a default value to a parameter which does not have a matching argument in the function call. Default value are specified when the function is declared. The compiler knows from the prototype how many arguments a function uses for calling.

Example :

```
float result (int marks 1, int marks2, int marks 3 = 75);
```

A subsequent function call

```
Average = result (60, 70);
```

passes the value 60 to marks 1, 70 to marks2 and lets the function use default value of 75 for marks3.

---

The call `Average = result ( 60, 70, 80);`

passes the value 80 to marks 3.

The important point to note is that only the trailing argument can have default values.

Consider the following program:

```
#include <iostream.h>
void repchar (char = '=', int = 30);
void main ()
{
 repchar ();
 repchar ('*');
 repchar ('+', 45);
}
void repchar (char ch, int x)
{
 for (int i= 1; i <= x; i + +)
 cout << ch;
 cout << "\n";
}
```

The default argument follows an equal sign, which is placed directly after the type name. In the above program, the function `repchar ( )` takes two arguments. It is called three times from `main ( )`.

- 1 First time it is called with no argument. The function assigns the default value '=' to `ch` and 30 to `x` and it print '=' symbol thirty times on the screen.
- 1 Second time it is called with one argument. The function assigns the default value 30 to `x` and it prints '\*' symbol thirty times on the screen.
- 1 Third time it is called with two argument. The actual parameters take priority over the default parameters. It prints '+' symbol forty-five times.

### **Function Overloading**

This means that some function name can be used to create functions that perform a variety of different tasks. This is known as function polymorphism in OOP. The function would perform different operations depending on the argument list in function call. The correct function to be invoked is determined by checking the number and

---

type of arguments. For example, an overloaded function `mpy ( )` handles different types of data as follows:

```
// Declaration
void mpy (int, int); // prototype 1
void mpy (int, float); // prototype 2
void mpy (float, int); // prototype 3
void mpy (float, float); // prototype 4

// Function calls
cout << mpy (3.5, 1.2); // uses prototype 4
cout << mpy (2, 5); // uses prototype 1
cout << mpy (5.2, 6); // uses prototype 3
cout << mpy (7, 6.2); // uses prototype 2
```

A function call first matches the prototype having the same number and type of arguments and then calls the appropriate function for execution. A best match must be unique.

The function selection involves the following steps:

- (i) The compiler tries to find an exact match in which types of actual arguments are the same and use that function.
- (ii) If an exact match is not found, the compiler uses the integral promotions to the actual arguments, such as

```
char to int
float to double
```

to find a match.

- (iii) When either of them fails, the compiler tries to use the built in conversions (the implicit conversions) to the actual arguments then uses the function whose match is unique. If the conversion is possible to have multi matches, then the compiler will generate an error message. Suppose, we use the following two functions:-

```
long abc (long n);
double abc (double x);
```

---



A function call such as  
abc (10)

will cause an error because int argument can be converted to either long or double, thereby creating an ambiguous situation.

- (iv) If all the steps fail, then the compiler will try the user defined conversions in combination with integral promotions and built in conversions to find a unique match.

---

### In-Text Questions 10.2

---

1. What are global and local variables?
  2. When do you define the function as inline?
  3. When do you use the keyword return while defining a function ?
  4. Construct function prototypes for descriptions given below :
    - (a) abc ( ) takes no arguments and has no return value.
    - (b) abc ( ) takes a float argument and returns an int.
    - (c) abc ( ) takes two double arguments and returns a double.
  5. A function printchar is defined as  
void printchar (char ch = '\*', int len = 40)  
{  
for (int x = 0; x < len ; x + + )  
cout << ch;  
cout << "\n";  
}  
  
How will you invoke the function printchar for following output:
    - (a) to print '\*' 40 times
    - (b) to print '\*' 20 times
    - (c) to print '=' 40 times
    - (d) to print '=' 30 times
  6. Given the function  
int threetimes (int x)
-

```
{
return x * 3 ;
}
```

Write a main function that is necessary to call this function.

7. What will be the output of the following programs ?

(a) `# include < iostream.h>`  
`void add (float);`  
`void main ( )`  
`{`  
`float x = 5.0;`  
`add (x);`  
`cout << x ;`  
`}`  
`void add (float x)`  
`{`  
`x = x + 2;`  
`}`

(b) `# include < iostream.h >`  
`void add (float);`  
`float x = 5.0;`  
`void main ( )`  
`{`  
`add (x);`  
`cout << x ;`  
`}`  
`void add (float x)`  
`{`  
`x = x + 2;`  
`}`

(c) `# include < iostream.h >`  
`float add (float);`  
`void main ( )`  
`{`  
`float x = 5.0;`

---

```
float y;
y = add (x);
cout << x << " " << y << "\n"
}
float add (float x)
{
return (x +2);
}
```

```
(d) #include < iostream.h >
void P1 (int &, int);
void main ()
{
int a = 5;
int b = 6;
cout << "output of the program \n";
cout << "values of a and b are " << a << b << "\n";
P1 (a, b);
cout << "values of a and b are" << a << b;
}
void P1 (int & a, int b)
{
a = 7 ;
b = 10 ;
}
```

```
(e) include < iostream.h >
int a, b;
void try2 (int x, int & y);
void main ()
{
int a = 10 ;
int b = 7 ;
try2 (a, b);
cout << a << b << "\n";
}
void try2 (int x, int & y)
```

---

```

{
b = x;
y = y + b;
x = y;
cout << x << y << b << "\n";
}

```

8. Fill in the blanks:

- (a) Any C++ program contains at least .....function(s).
- (b) After each function has done a job, control returns to .....
- (c) The declaration of a function in the calling program is known as a function .....
- (d) the function can be called in two ways ..... and .....
- (e) When the return type is not specified, the default return type of function is .....
- (f) To return the control back to the calling function, we use the keyword.....
- (g) In an ..... function, looping is not allowed.

9. State whether the following are true or false.

- (a) There is a limit on the number of functions that might be present in a C++ program.
- (b) A variable which is declared inside the function is called global variable.
- (c) When function is not returning anything, return type is void.
- (d) An inline function is preferred when its size is small.
- (e) In call by reference, any change made in the object inside the function will reflect in the value of the actual object.
- (f) A C++ program can have more than one function with the same name.
- (g) Inline is a keyword in C++
- (h) An inline function must be defined before it is called.
- (i) Any function in C++ can be declared as inline function.
- (j) It is necessary to specify the variable name in the function prototype.

## 10.8 What you have learnt

From this lesson you learnt the use of # include in a program, Also you learnt the use of some library functions included in the ctype.h, math.h, string.h and stdio.h in header files.

In this lesson we discussed about the definition of a function and different forms of a function calling statements. Also we discussed about the inline function and how function can be used. By now you should be in a position to write a program using the concept of function.

## 10.9 Terminal Questions

1. Write a program that will ask the user to enter a single character. Find out whether it is alphabetic, digit or special character.
  2. Write a program that will ask the user to enter his/her name. Convert all uppercase letter to lowercase letter and vice-versa.
  3. Write a program that will ask the user to enter a character. Check if it is alphabetic or not. If alphabetic, check whether it is in uppercase or lowercase.
  4. What is the difference between the following ?
    - (i) getchar ( ) & putchar ( )
    - (ii) gets ( ) & puts ( )
  5. Write a program that will count the number of characters in a sentence entered by user.
  6. Write a program that will count the number of words in a sentence entered by the user.
  7. Write a program that will count the number of A's, E's, I's, O's and U's in a sentence.
  8. Differentiate between the following :
    - (i) Actual and Formal parameters
    - (ii) Call by value and call by reference
  9. What do you mean by inline function ?
-

10. What are the advantages of function prototype in C++?
11. What is the main advantage of passing the arguments by reference ?
12. When do we need to use default arguments in a function?
13. What do you mean by overloading of a function?
14. Write a program in C++ to call a function which will return the cube of a given number using inline function.
15. Write a program in C++ that prints the largest of two number entered from the keyboard. Pass the two numbers to a function as argument, find the largest and return this value to a main function.

---

## 10.10 Feedback to In-Text Questions

---

### In-text Questions 10.1

1. Header file contains declaration of the function, prototype, class definition etc. It instructs the compiler to read and include another file as source code.
  2. Because some of the functions, class definition etc. used in a program are declared in header file.
  3. (i) math.h (ii) string.h (iii) conio.h (iv) ctype.h
  4. The build in function in C++ compiler is called library function.
  5. Both functions ask for a character from the user. The getche ( ) function echoes the character and getch ( ) function does not echo the character.
  6. (a) Enter your name  
MOHIT  
Hello  
MOHIT  
(b) RUBY
  7. (a) h  
(b) ctype.h
-

- (c) uppercase
  - (d) S2
  - (e) strlen ( )
  - (f) math.h
8. (a) T (b) F (c) T (d) T (e) F (f) F  
(g) T (h) T (i) F

### In-text Questions 10.2

1. Global variables are available to all the functions. Local variable is available only in the function where it is defined.
  2. If the function code is short, i.e., contains either one or two lines, it is termed as inline.
  3. If a function returns some value to the calling function, return statement is used.
  4. (a) void abc ( )  
(b) int abc (float);  
(c) double abc (double, double);
  5. (a) void printchar ( ) ;  
(b) void printchar ( '\*', 20);  
(c) void printchar ('=');  
(d) void printchar ('=', 30);
  6. void main ( )  
{  
int y, l;  
cout << "Enter value" ;  
cin >> y ;  
l = threetimes (y) ;  
cout << l;  
}
-

7. (a) 5  
(b) 5  
(c) 5 7  
(d) output of the program  
values of a and b are 5 6  
values of a and b are 7 6  
(e) 17 17 10  
10 17
8. (a) 1  
(b) statement after calling function statement  
(c) prototype  
(d) call by value  
call by reference  
(e) int  
(f) return  
(g) inline
9. (a) F (b) F (c) T (d) T (e) T  
(f) T (g) T (h) T (i) F (j) F
-



# 11

## Array

### 11.1 Introduction

The ordered collection of identical elements is called an array. Sometimes, it is required to store a large number of variables of the same type, one can use an array. For example:

```
int A[5];
```

The above statement declares an array A which reserves 5 memory locations for storing integer data type. These locations are A[0], A[1], A[2], A[3] and A[4]. The number within the square bracket is called index or subscript. The array can be one dimensional (one subscript) or two dimensional (two subscripts).

### 11.2 Objectives

After going through this lesson, you would be able to:

- explain the concept of an array
  - define, access and initialize array elements
  - perform common operations on arrays
  - explain two dimensional array
  - define, access and initialize two dimensional array elements
-

### 11.3 Initialization of One Dimensional Array

An array can be uninitialized along with declaration. For array initialization it is required to place the elements separated by commas enclosed within braces.

```
int A[5] = {1, 2, 3, 4, 5};
```

It is possible to leave the array size open. The compiler will count the array size.

```
int B [] = {6, 7, 8, 11, 10};
```

### 11.4 Initialization of String

An array of characters known as character string may be initialized by placing the string in double quotes.

```
Char A [20] = "Computer Science";
Char B [] = "Calcutta";
```

A string is a series of characters stored in consecutive bytes of memory. This implies that one can store a string in an array of characters, with each character kept in its own array element. The last character of every C++ string is Null character. Therefore

```
char city [] = {'c', 'a', 'l', 'c', 'u', 't', 't', 'a'};
```

is an array of characters while

```
char city [] = {'c', 'a', 'l', 'c', 'u', 't', 't', 'a', '\0'};
```

is a string because it is terminated by a null character.

### 11.5 Processing an Array

The various operations possible on arrays are :

- Traversal
  - Searching
  - Sorting
  - Insertion
  - Deletion
-

### Traversal

It means to access each location of an array, may be for display purpose. Consider a program which will read five values from the user and finds out the maximum value.

```
#include <iostream.h >
void main ()
{ int T, A[5], i;
 cout << "Enter five values";
 for (i = 0; i < 5; i ++)
 cin >> A [i];
 T = A [0];
 for (i = 1; i < 5; i ++)
 {
 if (T < A [i])
 T = A [i];
 }
 cout << "Maximum value" << T;
}
```

### Searching

This method finds out whether the data entered by the user is present in an array or not. There are two types of searching method.

- (i) Linear or Sequential search
- (ii) Binary search

#### Linear or sequential search

This method is slower, inefficient and works on unsorted list. If the data we are searching is not present in the list, we come to know at the end of the list.

```
// Linear search
#include <iostream.h >
void main ()
{
```

---

```

int A[5], l, data, flag = 0;
cout << "Enter five values";
for (l = 0; l < 5; l++)
cin >> A [l];
cout >> 'Enter data to be searched";
cin >> data;
for (l=0; l < 5; l++)
{
if (A[l] == data)
flag = 1;
}
if (flag == 1)
cout << "Data present";
else
cout << "Data not present";
}

```

## Binary search

This method requires the array to be either in ascending or descending order. This method calculates the mid location from initial and final locations and compares the data with the value present in mid location. Consider the case where the list is in ascending order. If data is less than a [mid] then data is present in the upper half otherwise data is present in the lower half. The value of either final or initial will be changed. The mid value is calculated again and searching continues till the data is found or initial is greater than final. The values of initial, final and mid for searching a value of **35** in an ascending order sorted list containing 9 elements is shown below:

|          |     |     |     |     |     |     |     |     |     |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|          | 10  | 15  | 18  | 20  | 27  | 30  | 35  | 40  | 45  |
| location | (0) | (1) | (2) | (3) | (4) | (5) | (6) | (7) | (8) |

Let the array have value:

| Initial | Final | Mid = $\frac{\text{Initial} + \text{Final}}{2}$ | A [Mid]              |
|---------|-------|-------------------------------------------------|----------------------|
| 0       | 8     | 4                                               | 27                   |
| 5       | 8     | 6                                               | 35                   |
|         |       |                                                 | Search is successful |

```
// Binary search
#include < iostream.h >
const int N = 9;
void main ()
{ int A[N], l, initial, final, mid, data;
 cout << "Enter nine values in ascending order";
 for (l = 0; l < N; l ++)
 cin >> A [l];
 cout << "Enter data to be searched";
 cin >> data;
 initial = 0;
 final = N - 1;
 mid = (initial + final) / 2;
 While (initial <= final) && (A[mid] != data))
 {
 if (A [mid] > data)
 final = mid - 1;
 else
 initial = mid + 1;
 }
 if (A [mid] == data)
 cout << "data is present";
 if (initial > final)
 cout << "data not present in the list";
}
```

The advantage of Binary search is that each search cuts the list in to half. A list of 10,000 names can be searched in just 12 searches.

## Sorting

It is a method to arrange the list either in ascending or descending order.

### Bubble sort

Consider an array of five locations to be sorted in ascending order:

|       |    |   |   |   |   |
|-------|----|---|---|---|---|
|       | -1 | 9 | 5 | 0 | 2 |
| Index | 0  | 1 | 2 | 3 | 4 |

In this sorting method, A[0] is compared with A[1]. If A[0] is greater than A[1], the values are swapped. Then A[1] is compared with A[2], A[2] is compared with A[3], and A[3] is compared with A[4]. In all cases if the  $i^{\text{th}}$  location has value greater than  $i + 1$ , the values are swapped. The entire process is repeated  $N-1$  times where  $N$  is the number of data in an array.

```
include < iostream.h >
const int N = 5;
void main ()
{
 int A [N], i, j, T;
 cout << "Enter values";
 for (i = 0; i < N; i ++)
 cin >> A [i];
 // sorting
 for (i = 0; i < N - 1; i ++)
 for (j = 0; j < N - 1 - i; j ++)
 if (A[j] > A [j + 1])
 {
 T = A [j];
 A [j] = A [j + 1];
 A [j + 1] = T;
 }
 cout << "Sorted array is";
 for (i = 0; i < N; i ++)
 cout << A [i];
}
```

### Selection sort

Consider an array having N elements to be sorted in ascending order. Initially, first element is compared with others so that it holds the smallest value. In the next pass, second element is compared with others so that it holds the smallest value. This procedure is repeated for the entire array.

```
include < iostream.h >
const int N = 5;
void main ()
{
 int A [N], l, j, T;
 cout << "Enter values";
 for (l = 0; l < N; l ++)
 cin >> A [l];
 // sorting
 for (l = 0; l < N - 1; l ++)
 for (j = l; j < N; j ++)
 if (A [l] > A [j])
 {
 T = A [l];
 A [l] = A [j];
 A [j] = T;
 }
 // printing the sorted data
 for (l = 0; l < N; l ++)
 cout << A [l];
}
```

### Insertion

It means addition of a data item in the middle or at the end of the array. If data is to be added after a given data item then the location of the data item is first determined by applying search procedure and then the insertion procedure is implemented.

```
include < iostream.h >
void main ()
{
 int x [20] ;
```

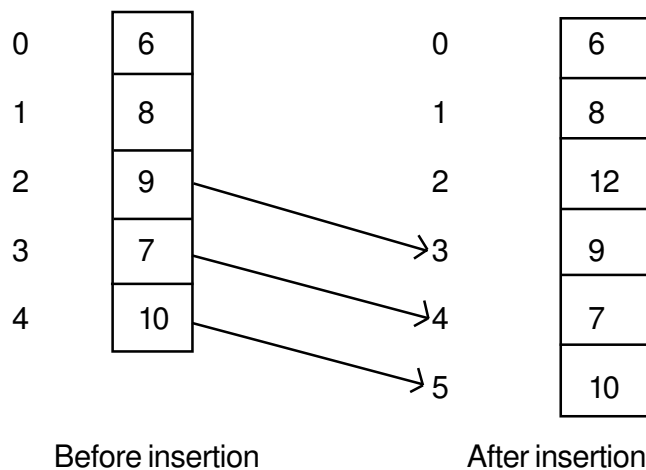
---

```

int l, loc, n, data;
cout << "Enter the no. of elements";
cin >> n;
for (l = 0; l < n; l ++)
{
cout << "Enter the array element";
cin >> x [l];
}
cout << "Enter the location after which data is to be inserted";
cin >> loc;
for (l = n - 1; l >= loc; l --)
x [l + 1] = x [l];
cout << "enter the new data to be added";
cin >> x [loc];
n ++ ;
cout << "Array elements after insertion";
for (l = 0; l < n; l ++)
cout << x [l];
}

```

Let data 12 to be inserted at location 2



## Deletion

It means removal of a data. First the location of the item to be deleted is determined by applying an appropriate search procedure and then the value present at particular





---

## In-Text Questions 11.1

---

1. What is an array ?
  2. What do you mean by index of an array?
  3. The elements of a five element array are numbered from ..... to .....
  4. Write a statement that defines one dimensional array called student of type int that holds 5 data elements.
  5. In the following array declarations find the number of elements in each array and total no. of bytes required to store each array.
    - (a) int A [ 20 ];
    - (b) char z [ 18 ] ;
  6. What will be the output of the following programs:
    - (a) 

```
#include < iostream.h >
void main ()
{
int a [5], t ;
for (i= 0; i < 5; i ++)
a [i] = 5 * i;
for (i = 0; i < 5; i ++)
cout << a [i];
}
```
    - (b) 

```
#include < iostream.h >
void main ()
{
char title [] = "Computer";
for (int i = 0; title [i]; i ++)
cout << "\n" << (title + i);
}
```
- 
-

## 11.6 Two Dimensional Array

It has two subscript or index, first for row and second for column. For example:

```
int A [5] [4];
```

The above has five rows and four columns. The total number of elements are 20. The subscripts of 20 elements are:

|             |             |             |             |
|-------------|-------------|-------------|-------------|
| A[ 0 ][ 0 ] | A[ 0 ][ 1 ] | A[ 0 ][ 2 ] | A[ 0 ][ 3 ] |
| A[ 1 ][ 0 ] | A[ 1 ][ 1 ] | A[ 1 ][ 2 ] | A[ 1 ][ 3 ] |
| A[ 2 ][ 0 ] | A[ 2 ][ 1 ] | A[ 2 ][ 2 ] | A[ 2 ][ 3 ] |
| A[ 3 ][ 0 ] | A[ 3 ][ 1 ] | A[ 3 ][ 2 ] | A[ 3 ][ 3 ] |
| A[ 4 ][ 0 ] | A[ 4 ][ 1 ] | A[ 4 ][ 2 ] | A[ 4 ][ 3 ] |

### Initialization of Two Dimensional Array

The initialization is done at the time of declaration of an array.

For example:

```
int A [2] [4] = {1, 2, 3, 4, 5, 6, 7, 8};
```

For more clarity

```
int A [2] [4] = { {1, 2, 3, 4 }, {5, 6, 7, 8 } };
```

The above data can be grouped. The inner braces are ignored by the compiler.

For two dimensional array, two loops are required. The following program finds out the maximum value stored in two dimensional array.

```
#include <iostream.h >
const int M = 5;
void main ()
{ int A [M] [M], i, j, T;
 cout >> "\n Enter Array Elements";
 for (i = 0; i <= M - 1; i ++)
 for (j = 0; j <= M - 1; j ++)
 cin >> A [i] [j];
```

---

```
T = A [0] [0];
for (i = 0; i <= M - 1; i + +)
for (j = 0; j <= M - 1; j + +)
{
 If (T < A [i] [j]
 T = A [i] [j];
}
cout << "Largest value" << T << "\n";
}
```

---

## In-Text Questions 11.2

---

1. Define a two dimensional array A having 2 rows, 3 columns and stores int data type.
  2. How many elements are there in the array defined in question 1 above ?
  3. How many bytes the above declaration (see question 1) will take in the memory?
  4. Consider the following array declarations. Find the number of elements in each array and total no. of bytes required to each array.
    - (a) int x [ 5 ] [ 10 ];
    - (b) long y [ 5 ] [ 10 ];
- 

## 11.7 What you have learnt

In this lesson we learnt how to define, access and assign values to one dimensional and two dimensional array. We have discussed the operations one can perform with one dimensional array.

## 11.8 Terminal Questions

1. What is the need of an array?
  2. What do you mean by searching?
  3. Give one advantage and one disadvantage of Linear and Binary search?
  4. What do you mean by sorting?
-

5. Which method is better: Bubble or Selection sorting? Why ?
6. Write a program that will read 20 float values in a one dimensional array and find out the following:
  - (i) Number of values greater than zero.
  - (ii) Number of values equal to zero.
  - (iii) Number of values less than zero.
7. Write a program that will find out the sum of two diagonals of two dimensional array of A [N] [N].
8. Write a program that will find out whether the data entered by the user present in a one dimensional array of 10 elements using:
  - (i) Linear search
  - (ii) Binary search
9. Write a program that will sort the given ten numbers in descending order using:
  - (i) Bubble sort
  - (ii) Selection sort

---

## 11.9 Feedback to In-Text Questions

---

### In-text Questions 11.1

1. Array is a collection of the same data type.
  2. The index designates the position in array ordering.
  3. 0, 4
  4. `int student [ 5 ];`
  5. (a) Number of elements - 20  
Total number of bytes - 40  
(b) Number of elements - 18  
Total number of bytes - 18
-

6. (a) 0 5 10 15 20
- (b) COMPUTER  
OMPUTER  
MPUTER  
PUTER  
UTER  
TER  
ER  
R

### In-text Questions 11.2

1. int A [ 2 ] [ 3 ];      2. 6      3. 12
4. (a) Number of elements - 50  
Total number of bytes - 100
- (b) Number of elements - 50  
Total number of bytes - 200
-

# 12

## Structure, Typedef & Enumerated Data Type

### 12.1 Introduction

Sometimes it is required to store information, which can be of same or different types. For this purpose, structure statement is used. This lesson contains the three important statements of C++ language.

### 12.2 Objectives

After going through this lesson, you would be able to:

- define variable, initialize and access members of the structure
- explain the concept of nested structure
- explain typedef statement in a program
- define enum statement and use it

### 12.3 Structure

A structure is a collection of simple variables which can be of same or different types. The data items in a structure are called the members of the structure. Consider the following example:

```
include < iostream.h >
```

---

```
struct student
{
 char name [20];
 int marks;
};

void main ()
{
 student S1, S3;
 cin >> S1. name >> S1. marks;
 cout << S1. name << S1. marks;
 student S2 = {"ANKIT", 32};
 cout << S2. name << S2. marks;
 S3 = S2;
 cout << S3. name << S3. marks;
}
```

### 12.3.1 Variable of the structure

S1 and S2 are called variable of the structure student. They are given enough space in memory to hold the members of the structure (22 bytes each for above program).

### 12.3.2 Accessing of data members

The accessing of data members is done by using the following format:

structure variable. member name

### 12.3.3 Initialization of structure variable

Initialization is done at the time of declaration of a variable. For example:

Student S2 = {"ANKITA", 32};

### 12.3.4 Structure variable in Assignment Statements

The statement

S3 = S2;

assigns the value of each member of S2 to the corresponding member of S3. Since a large structure can have many members, such an assignment statement requires the compiler to do a considerable amount of work.

---



Note that one structure variable can be assigned to another only when they are of the same structure type, otherwise compiler will give an error.

### 12.3.5 Structure within Structure

This is called nesting of structure. Consider the following program in which date field is a structure.

```
include < iostream.h >
include < stdio.h >
struct today
{
int month;
int date;
int year;
};
struct biodata
{
char name [20];
today date_of_birth;
char add [30];
};
void main ()
{
biodata b;
cout << "Enter name";
gets (b.name);
cout << "Enter birth month";
cin >> b.date_of_birth. month;
cout << "Enter birth date";
cin >> b.date_of_birth.date;
cout << "Enter birth year";
cin >> b.date_of_birth.year;
cout << "Enter address";

gets (b.add);

cout << "Name" << b.name << "\n";
cout << "Month" <<b.date_of_birth.month << "\n";
```

---

```
cout << "Date" << b.date_of_birth.date << "\n";
cout << "Year" << b.date_of_birth.year << "\n";
cout << "Address" << b.add << "\n";
}
```

### 12.3.6 Accessing nested structure members

As the structure is nested inside another, we must apply the dot operator twice to access the structure members:

```
cin >> b.date_of_birth.month;
```

### 12.3.7 Initializing Nested Structure

It is initialized at the time of declaration of structure variable. For example:

```
biodata bd = {"MOHIT", {3, 21, 1991}, "Rohini"};
```

The inner braces are required just to separate the values, although it is ignored by the compiler.

---

## In-Text Questions 12.1

---

1. What is a structure?
  2. How is structure different from array?
  3. The variable of the structure is given space in the memory. True or False?
  4. What is the syntax of accessing data members of the structure?
  5. The initialization of a structure is done at the time of assigning data type. True or False?
  6. What is nesting of structure?
  7. Write a structure specification that includes three variables all of type int called hrs, mins and secs. Call this structure time.
  8. Write a statement that sets the hrs member of the time 2 structure variable equal to 11.
  9. If you have three variables defined to be a type struct time and this structure contains three int members, how many bytes of memory do the variables use together?
-

10. Write a C++ nested structure for the record given below. Use appropriate data type.

| Name | Class | Roll | Address  |
|------|-------|------|----------|
|      |       |      | House No |
|      |       |      | Street   |
|      |       |      | City     |
|      |       |      | State    |
|      |       |      | PIN Code |

Declare a variable of the structure. Assign the value of each variable in a nested structure.

11. Write statements in the following program segment that will assign values to all the fields of a structure.

```
struct date
{
int dd, mm, yy;
};
struct donor
{
char N [20];
date dob;
};
void main ()
{
donor person;

}
```

12. What will be the output of the following programs.

```
(a) #include <iostream.h>
void main ()
{
 struct rollno
 {
 int roll;
 };
 rollno r;
 r.roll = 100;
 cout << "roll = " << r.roll;
}
```

```
(b) #include <iostream.h>
void main ()
{
 struct student
 {
 int roll;
 int age;
 };
 student S1 = { 100, 15 };
 student S2 = S1;
 if (S1 == S2)
 cout << "The structures are equal";
 else
 cout << "The structures are not equal";
}
```

---

## 12.4 Typedef

It is used to define new data type for an existing data type. It provides an alternative name for standard data type. It is used for self documenting the code by allowing descriptive names. (for better understanding) for the standard data type.

---

The general format is:

```
typedef existing datatype new datatype;
```

For example:

```
typedef float real;
```

Now, in a program one can use datatype real instead of float.

Therefore, the following statement is valid:

```
real amount;
```

---

## In-Text Questions 12.2

---

1. What is the function of typedef statement?
2. Why typedef statement is used in a program?
3. What is the meaning of the following statement?

```
typedef int integer;
```

---

## 12.5 Enumerated Data Type

It has the following features:

- It is user defined.
- It works if you know in advance a finite list of values that a data type can take.
- The list cannot be input by the user or output on the screen.

**For example:**

```
enum months { jan, feb, mar, apr, may};
```

```
enum days { sun, mon, tue, wed, thu };
```

```
enum toys { cycle, bicycle, scooter };
```

The enum specifier defines the set of all names that will be permissible values of the type called members which are stored internally as integer constant. The first name was given the integer value 0, the second value 1 and so on.

---

**For example:**

```
jan = 0, feb = 1, mar = 2, apr = 3, may = 4
```

The ordering can be altered by using an equal sign and value.

```
enum months { jan = 1, feb, mar, apr, may };
```

Here jan = 1, feb = 2, mar = 3, apr = 4, may = 5

The value of the next element in the list is previous value plus one.

**For example:**

```
#include <iostream.h >
enum months { jan, feb, mar, apr, may };
void main ()
{
months m1, m2;
m1 = jan;
m2 = apr;
int diff = m2 - m1;
cout << "Months between" << diff << "\n";
if (m1 > m2)
cout << "m2 comes before m1";
}
```

The output of the above program is

```
Months between 3
```

because the members of the enumerated series is stored as integer constant in the memory.

---

### **In-Text Questions 12.3**

---

1. An enumerated type is considered as an integer type internally. True or False?
2. What is the meaning of following statement?

```
enum days { sun, mon, tue };
```

---

3. Can the enum data input by the user?
  4. (a) Write a statement that declares an enumerated data type called months with the values  
jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec  
(b) Define two variables m1 and m2 and assign them the values jan and mar respectively.
  5. The first three members of an enum type are normally represented by the values ....., ..... & .....
  6. Write a statement that declares an enumerated data type called colour with the values pink, red, green, and blue. Give these four names the integer values 33, 45, 58 and 78.
- 

## 12.6 What you have learnt

In this lesson, you learnt about three statements: struct, typedef and enum. Now you should be able to use these statements in C++ program.

## 12.7 Terminal Questions

1. Write a C++ nested structure for the record given below. Use appropriate data type.

|      |          |               |
|------|----------|---------------|
| Name | Roll No. | Date of Birth |
|      |          | Day           |
|      |          | Month         |
|      |          | Year          |

Assign value to all the members of the structure.

2. What will be the output of the following program?

```
#include <iostream.h >
enum days { sun, mon, tue, wed, thu, fri, sat };
void main ()
{
days today;
```

---

```
for (today = sun; today <= sat; today ++)
 cout << "\n today is " << today;
}
```

3. State the reason why

```
enum boolean {false, true};
is better than
enum boolean {true, false};
```

4. What is the advantage of typedef statement?

5. A phone number, such as 786-6370, can be thought of having two parts; the exchange code and the number. Write a program in C++ that uses a structure phone to store these two parts. Call the structure phone. Create two structure variable of type phone. Initialize one and have the user input a number for the other one. Then display both the numbers.

---

## 12.8 Feedback to In-Text Questions

---

### In-text Questions 12.1

1. Structure is a collection of simple variables which can be of same or different types.
  2. An array can hold variables of same data types whereas structure can hold variables of same or different data types.
  3. True
  4. structure variable. member
  5. True
  6. The structure within a structure is called nesting of structure.
  7. struct time  
{  
int hrs;  
int mins;  
int secs;  
};
-



```
8. time time2;
 time2. hrs = 11;

9. 18

10. struct add
 {
 char houseno [5];
 char street [10];
 char city [10];
 char state [10];
 char zip [10];
 };
 struct biodata
 {
 char name [20];
 char class [5];
 into roll;
 add address [30];
 };
 biodata data;
 cin >> data. name;
 cin >> data. class;
 cin >> data. roll;
 cin >> data. address. houseno;
 cin >> data. address. street;
 cin >> data. address. city;
 cin >> data. address. state;
 cin >> data. address. zip;

11. cin >> person. N;
 cin >> person. dob. dd;
 cin >> person. dob. mm;
 cin >> person. dob. yy;
```

---

12. (a) 100
- (b) The structures are equal

### **In-Text Questions 12.2**

1. To define new data type of existing data type.
2. To make the program readable.
3. Here instead of int data type, integer can be used in a program.

### **In-text Questions 12.3**

1. True
  2. sun = 0, mon = 1, tue = 2
  3. No
  4. (a) enum months  
{ jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec }
  - (b) months m1, m2;  
m1 = jan;  
m2 = mar;
  5. 0, 1, 2
  6. enum colour {pink = 33, red = 45, green = 58, blue = 78 };
-

# 13

## Classes & Objects with Constructors/Destructors

### 13.1 Introduction

In object oriented programming, the emphasis is on data rather than function. Class is a way that binds the data & function together. Constructor is a specially designed class and destructor returns the memory addresses back to the system.

### 13.2 Objectives

After going through this lesson, you would be able to

- define class and object
- access the members of the class
- learn about three visibility modes: public, private and protected
- familiarize with constructor and its types
- define constructor with default arguments
- use destructor

### 13.3 Class

A class is way to bind the data and its associated functions together. It allows data

---

functions to be hidden, if necessary from external use. A class specification has two parts.

- (i) Class declaration
- (ii) Class function definitions

The general format of a class declaration is

```
class class-name
{
private :
 Variable declarations;
 Function declarations;
public:
 Variable declarations;
 Function declarations;
protected :
 Variable declarations;
 Function declarations;
};
```

The keyword `class` is followed by the name of the class. The body of the class is enclosed between braces and terminated by semi-colon. The class body contains the declaration of variables and functions. These are collectively called members. The variables declared inside the class are called data members. The functions are known as member functions. The keywords `public`, `private` and `protected` are called visibility modes.

The data member and member functions present in `private` and `protected` mode can be accessed by the member function in `public` mode. the `private` and `protected` modes are exactly the same except that `protected` can be inherited (covered later) but `private` mode cannot be inherited.

Therefore, for the class concept, `private` mode is used.

The use of keyword `private` is optional. By default, the members of a class are `private`. If the labels are missing, members are `private` by default. Such a class is completely hidden from outside world and does not serve any function.

---

### **Example 1**

```
include < iostream.h >
class student
{
private :
 char name [80];
 int rn ;
 float marks ;
private :
 void getdata () ;
 void putdata () ;
};
void student :: getdata ()
{
 cin >> name >>rn >> marks ;
}
void student :: putdata ()
{
 cout << name << rn << marks ;
}
void main ()
{
 student st ;
 st. getdata () ;
 st. putdata () ;
}
```

### **13.3.1 Creating Objects**

An object is an instance of a class and it can be created by using class name.

```
student st ;
```

Object can be created when a class is defined by placing their names immediately after the closing brace.

---

```
class student
{
} x, y, z ;
```

The above definition would create the object x, y and z of type student.

### 13.3.2 Accessing class member

Through object, data member and member function present in public can be accessed. The general format is :

Object name . data member ;

Object name . member function ;

The dot operator is called the class member access operator.

### 13.3.3 Member function definition

Member function can be defined in two ways:

- (i) Inside the class
  - (ii) Outside the class
- (i) Inside the class : When a member function is defined inside a class, it is considered to be inline by default. If a member function is very small then it should be defined inside the class.

The class declaration of previous program will be as follow:

```
class student
{
 char name [20];
 int rn ;
 float marks ;
public :
 void getdata ()
 {
 cin >> name >> m >> marks ;
 }
 void putdata ()
```

---

```
 {
 cout << name << rn << marks :
 }
};
```

- (ii) Outside the class : When a function has larger code then it should be defined outside the class declaration. The prototype of such functions, however, must be provided in the class definition. The operator ‘:.’ known as scope resolution operator is used to associate member functions to their corresponding class.

The format is :

```
return _type class_name :: function_name
```

### 13.3.4 Nesting of member functions

A member function can be called by using its name inside another member function of the same class. This is known as nesting of member functions. The following program illustrates this concept.

```
class geatest
{
 int x, y, z ;
 public :
 void getdata () ;
 void display () ;
 int largest () ;
};
int geatest :: largest ()
{ int T ;
 if (x > y)
 T = x ;
 else
 T = y ;
 if (T > z)
 return (T) ;
 else
 return (z) ;
}
```

---

```
void greatest : : getdata ()
{
 cout << "Enter values of x, y, z" << "\n";
 cin >> x >> y >> z ;
}
void greatest : : display ()
{
 cout << "largest value" << largest () << "\n";
}
void main ()
{
 greatest A;
 A. getdata () ;
 A. display () ;
}
```

### 13.3.5 Memory Allocation for objects

The member function are created and placed in the memory space only once when they are defined in class specification. All the objects belonging to that class use the same member functions. Space for member variable is allocated separately for each object because member variable holds different value for different objects.

### 13.3.6 Array of object

Consider the following program.

```
class emp
{
 char name [30] ;
 int empno ;
public :
 void getdata () ;
 void putdata () ;
};
void main ()
{
 emp e [10] ;
```

---



```
for (i = 0; i < 10; i ++)
e [i]. getdata () ;
for (i = 0; i < 10; i ++)
e [i] .putdata () ;
```

In the above program, the ten objects are created, namely, e [ 0 ], e [ 1 ] -----  
e [ 9 ].

The statement e [ i ]. getdata ( )  
will get the data of the ith element of the array e.

---

### In-Text Questions 13.1

---

1. What is a class ?
  2. Define object.
  3. What are the three visibility modes?
  4. What is the default mode to the members of a class?
  5. How are public members of a class different from private members of a class?
  6. Write a class specifier that creates a class called student with one private data member called rn, of type int, and two public member function : putdata ( ) and getdata ( ).
  7. Fill in the blanks:
    - (a) All members of a class are ..... by default.
    - (b) A function declared and defined inside the class definition by default is .....
    - (c) To define member function outside the class ..... operator is used.
    - (d) Operator :: is known as ..... operator.
  8. Whether the following are True or False.
    - (a) A class contains only private data members.
    - (b) A class member is accessed using dot operator.
-

- (c) The class declaration must end with a semicolon.
  - (d) Scope resolution operator always be used for definition of member functions outside the class declaration.
- 

## **13.4 Constructor**

A constructor is a special member function that initializes the objects of its class. It is special because its name is the same as the class name. It is invoked automatically whenever an object is created. It is called constructor because it constructs the values of data members of the class. It does not have any return type, not even void.

A constructor is declared and defined as follows :

```
class student
{
 int rn;
 int total ;
 public
 student ()
 {
 rn = 0 ; total = 0 ;
 }
};
```

The declaration

```
student st ;
```

Invokes the constructor, student ( ) and assign rn = 0 and total = 0.

### **13.4.1 Default Constructor**

A constructor that accepts no parameter is called default constructor. If no such constructor is defined, then the compiler supplies a default constructor. In that case, it is called nothing-to-do constructor.

### **13.4.2 Parameterized Constructors**

The constructors that can take arguments are called parameterized constructors.

---

```
class student
{
 int rn, total ;
 public
 student (int x, int y)
 {
 rn = x ; total = y ;
 }
};
```

When the object is created, we must supply arguments to the constructor function. This can be done in two ways:

- By calling the function explicitly
- By calling the function implicitly

The first call is implemented as follows :

```
student S1 = student (1, 70) ;
```

The second call is implemented as follows :

```
student S1 (1, 70) ;
```

The second method is used very often as it is shorter.

### 13.4.3 Copy Constructor

A copy constructor takes a reference to an object of the same class as itself as an argument. Consider the following program segment:

```
class student
{
 int rn, total ;
 public :
 student (int x, int y)
 {
 rn = x ; total = y ;
 }
 student (Student & i)
 {
```

---

```
rn = i. rn ;
total = i. total ;
}
};
```

The above program has both parameterized and copy constructor. The statement

```
student S1 (1, 75) ;
```

calls the parameterized constructor and assigns 1 to rn and 75 to total. The statement

```
student S2 (S1) ;
```

uses copy constructor and initializes an object S2 from another object S1. Another form of the statement is

```
student S2 = S1 ;
```

The process of initialization through a copy constructor is known as copy initialization.

Note that the statement

```
S2 = S1 ;
```

does not invoke the copy constructor. However, it simply assigns the value of S1 to S2, member by member.

### **13.5 Constructor with default arguments**

The constructor can be declared with default argument.

For example :

```
student (int rn, int total = 0) ;
```

Here the default value of total is zero.

Then the statement

```
student S1 (2) ;
```

assigns the value 2 to rn and 0 to total.

However, the statement

```
student S2 (3, 75) ;
```

---

assigns 3 to rn and 75 to total. In this case actual parameter takes the priority over default parameter. All default values should be on the right side.

Consider the following statement

```
A (int = 0) ;
```

It has only one argument. It can be called in two ways.

```
A B ;
A B (5) ;
```

In the first statement, no parameter is supplied. In the second statement, one parameter is supplied. When no parameter is supplied, it becomes a default constructor. When both the forms are used in a class (default constructor and constructor with one default argument), it causes ambiguity for a statement such as

```
A B
```

(whether to call A ( ) or A ( int = 0 )

## 13.6 Destructor

It is used to destroy the objects that have been created by a constructor. The destructor is a member function whose name is the same as the class name but is preceded by a tilde. For example the destructor of the class student can be defined as

```
- student () ;
```

It never takes any argument nor does it return any value. It will be invoked by the compiler upon exit from the program ( or function or block) to clean the storage. It is a good practice to declare destructor in a program because it releases memory space for future use.

---

### In-Text Questions 13.2

---

1. What is a constructor?
  2. What is a default constructor?
  3. If l1 and l2 are the objects of the same class, what does the following statement do ?  
l1 = l2;
-

4. A constructor is executed automatically when the object is created True or False?
  5. What is a destructor?
  6. Fill in the blanks:
    - (a) A constructor name is the same as .....
    - (b) A constructor is executed automatically when an object is .....
    - (c) Destructor is executed ..... at the end of the program
    - (d) A class can have any number of constructors but only ..... destructor at a time
    - (e) The name of the destructor function is same as that of class but proceed with a symbol.....
    - (f) The constructor which accepts value from outside is called ..... constructor.
  7. State whether the following are true or False.
    - (a) A constructor name is not the same as class name.
    - (b) In a class, you can have more than one constructor with the same name.
    - (c) Constructor does not return any value.
    - (d) A constructor that accepts no parameter is known as default constructor.
    - (e) Destructor never takes any arguments.
    - (f) Destructor is called automatically at the end of compound statement, function or main program.
    - (g) Constructor initialize the data members of a class.
- 

## **13.7 What you have learnt**

In this lesson, you learnt about class, object, constructor and its type and destructor. Now you should be in a position to write a program using destructor. Also, you learnt why destructor is included in a program.

---

### 13.8 Terminal Questions

1. How does a class accomplish data hiding ?
2. How does C++ structure differ from C++ class ?
3. What do you understand by visibility modes in class derivations ? What are these modes.
4. Define a class Teacher with the following specifications :

Private members :

|                |               |
|----------------|---------------|
| Name           | 20 characters |
| Subject        | 10 characters |
| Basic, DA, HRA | float         |
| Salary         | float         |

Calculate ( ) function computes the salary and returns it. Salary is sum of Basic, DA and HRA

Public members :

Readdata ( ) function accepts the data values and invokes the calculate function.

Displaydata ( ) function prints the data on the screen.

5. Define a class worker with the following specifications :

Private members of class worker

|              |                                            |
|--------------|--------------------------------------------|
| wno          | integer                                    |
| wname        | 25 characters                              |
| hrwk, wgrate | float (hour worked and wage rate per hour) |
| totwage      | float (hrwk * wgrate )                     |
| calcwg ( )   | A function to find                         |

hrwk \* wgrate with float return type.

Public members of class workder

---

In\_data ( ) a function to accept values for wno, wname, hrwk, wgrate and invoke calcwg ( ) to calculate netpay.

Out\_data ( ) a function to display all the data members on the screen

6. What are the special properties of a constructor function?
7. What is parameterized constructor?
8. What is copy constructor ?
9. What is the importance of destructors ?

---

## **13.9 Feedback to In-Text Questions**

---

### **In-text Questions 13.1**

1. Class is a way to bind data and function together.
  2. The instance (variable) of the class is called the object.
  3. public, private, protected
  4. private
  5. A public member of the class can be accessed by the object whereas private members can be accessed by the member function of public.
  6. 

```
class student
{
 int rn ;
 public :
 void getdata () ;
 void putdata () ;
};
```
  7.
    - a) private
    - b) inline
    - c) ::
    - d) Scope resolution
-



- 8. a) F
- b) T
- c) T
- d) T

### **In-text Questions 13.2**

- 1. A special function whose name is same as the class name is called constructor. It does not have any return type. It is executed automatically when an object is created.
  - 2. A constructor with no parameter is called default constructor.
  - 3. It assigns all the members of object l2 to l1 one by one.
  - 4. True
  - 5. It is a function whose name is same as the class name but preceded by the tilde sign. It is executed automatically at the end of block or function. It releases the memory which can be used to store some other data or instruction.
  - 6.
    - a) class name
    - b) created
    - c) automatically
    - d) one
    - e) tilde (~)
    - f) parameterized
  - 7.
    - a) F
    - b) T
    - c) T
    - d) T
    - e) T
    - f) T
    - g) T
-

# 14

## Inheritance Extending Classes

### 14.1 Introduction

This lesson discusses about inheritance, the capability of one class to inherit properties from another class as a child inherits some properties from his/her parents. The most important advantage of inheritance is code reusability. Once a base class is written and debugged, it can be used in various situations without having to redefine it or rewrite it. Reusing existing code saves time, money and efforts of writing the code again. Without redefining the old class, you can add new properties to desired class and redefine an inherited class member function.

### 14.2 Objectives

After going through this lesson, you would be able to

- explain the concept of inheritance
  - describe the five forms of inheritance
  - define three types of inheritance
  - explain all three visibility modes
  - describe the concept of abstract class & virtual class
-

### 14.3 Need for Inheritance

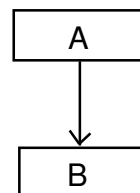
Inheritance is one of the important concepts of object-oriented language. There are several reasons why this concept was introduced in object oriented language. Some major reasons are:

- (i) The capability to express the inheritance relationship which ensures the closeness with the real world model.
- (ii) Idea of reusability, i.e., the new class can use some of the features of old class.
- (iii) Transitive nature of inheritance, i.e., it can be passed on further.

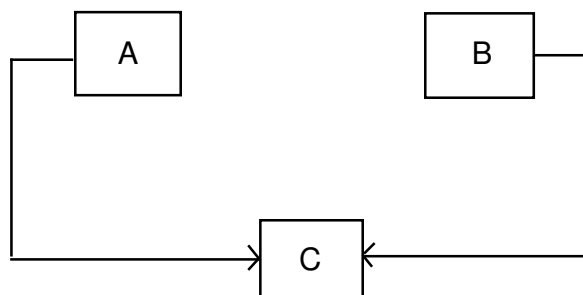
### 14.4 Different Forms of Inheritance

The mechanism of deriving a new class from an old one is called inheritance (or derivation). The old class is referred to as the base class and new one is called the derived class. There are various forms of inheritance.

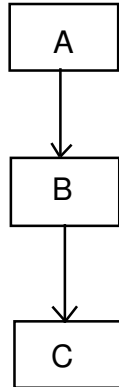
- (i) Single inheritance —> A derived class with only one base class is called single inheritance.



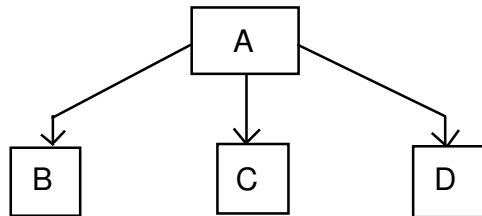
- (ii) Multiple inheritance —> A derived class with several base classes is called multiple inheritance.



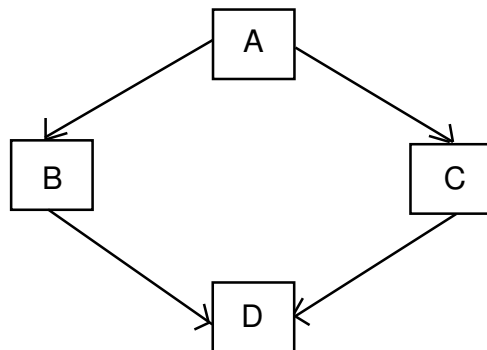
- (iii) Multilevel inheritance —> The mechanism of deriving a class from another derived class is called multilevel inheritance.
-



- (iv) Hierarchical inheritance → One class may be inherited by more than one classes. This process is known as hierarchical inheritance.



- (v) Hybrid inheritance → It is a combination of hierarchical and multiple inheritance.



## 14.5 Defining Derived Class

A derived class is defined by specifying its relationship with the base class using visibility mode.

The general form of defining a derived class is:

```
class derived_class : visibility_mode base_class
{

 _____ // members of derived class.
};
```

The colon indicates that the `derived_class` is derived (inherits some property) from `base_class`.

The base class(es) name(s) follow(s) the colon (:). The names of all the base classes of a derived class follow : (colon) and are separated by comma. The visibility-mode can be either private or public or protected. If no visibility mode is specified, then by default the visibility mode is considered as private.

Following are some examples of derived class definitions:

```
class Marksheet : public student // public derivation
{
 // members of derived class
};
class Marksheet : private student // private derivation
 // members of derived class
};
class Marksheet : protected student // protected derivation
{
 // members of protected class
};
```

In the above definitions, Marksheet is the derived class of student base class. The visibility mode public indicates that student is a public base class. Similarly, the visibility modes private or protected indicates that student is private base class or protected base class respectively.

---

When we say that members of a class are inheritable, it means that the derived class can access them directly. However, the derived class has access privilege only to the non-private members of the base class. Although the private members of the base class cannot be accessed directly, yet the objects of derived class are able to access them through the non-private inherited members.

## **14.6 Multiple Inheritance**

As we know that a subclass inheriting from multiple base classes is known as Multiple inheritance. The Syntax of defining a derived class is given below :

```
class derived_classname : mode baseclass1 , mode baseclass2
{
// members of derived class
};
```

### **Example 1**

```
class marks : public semester1 , private semester2
{
// members
};
```

## **14.7 Visibility Modes**

It can be public, private or protected. The private data of base class cannot be inherited.

- (i) If inheritance is done in public mode, public members of the base class become the public members of derived class and protected members of base class become the protected members of derived class.
- (ii) In inheritances is done in a private mode, public and protected members of base class become the private members of derived class.
- (iii) If inheritance is done in a protected mode, public and protected members of base class become the protected members of derived class.

The following table shows the three types of inheritance:

---

| Base class<br>Access specifier | Derived class    |                  |                  |
|--------------------------------|------------------|------------------|------------------|
|                                | public           | private          | protected        |
| public                         | public           | private          | protected        |
| private                        | Not<br>inherited | Not<br>inherited | Not<br>inherited |
| protected                      | protected        | protected        | Protected        |

### The Public Visibility mode

The following example and figure illustrate the public derivation in classes.

```
class student
{
private :
 int x;
 void getdata ();
public:
 int y;
 void putdata ();
protected:
 int z;
 void check ();
};
class marks : public student
{
private :
 int a ;
 void readdata ();
public :
 int b;
 void writedata ();
protected :
```

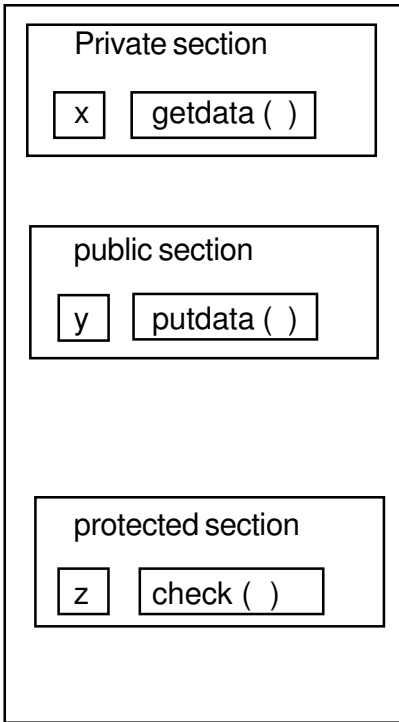
---

```

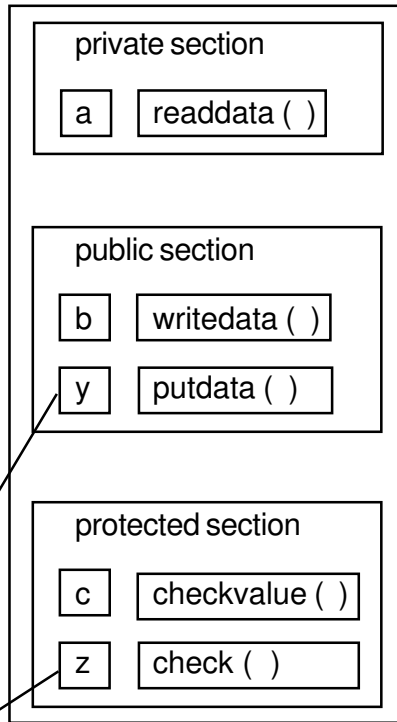
int c;
void checkvalue ();
};

```

class student



class marks



Inherited from base class student

**Fig. 14.1 public derivation of a class**

The public derivation does not change the access specifiers for inherited members in the derived class. The private data of base class student cannot be inherited.

### The Private Visibility Mode

We are using the same example, but the derivation is done in private mode.

```

Class student
{
// same as in previous example

```

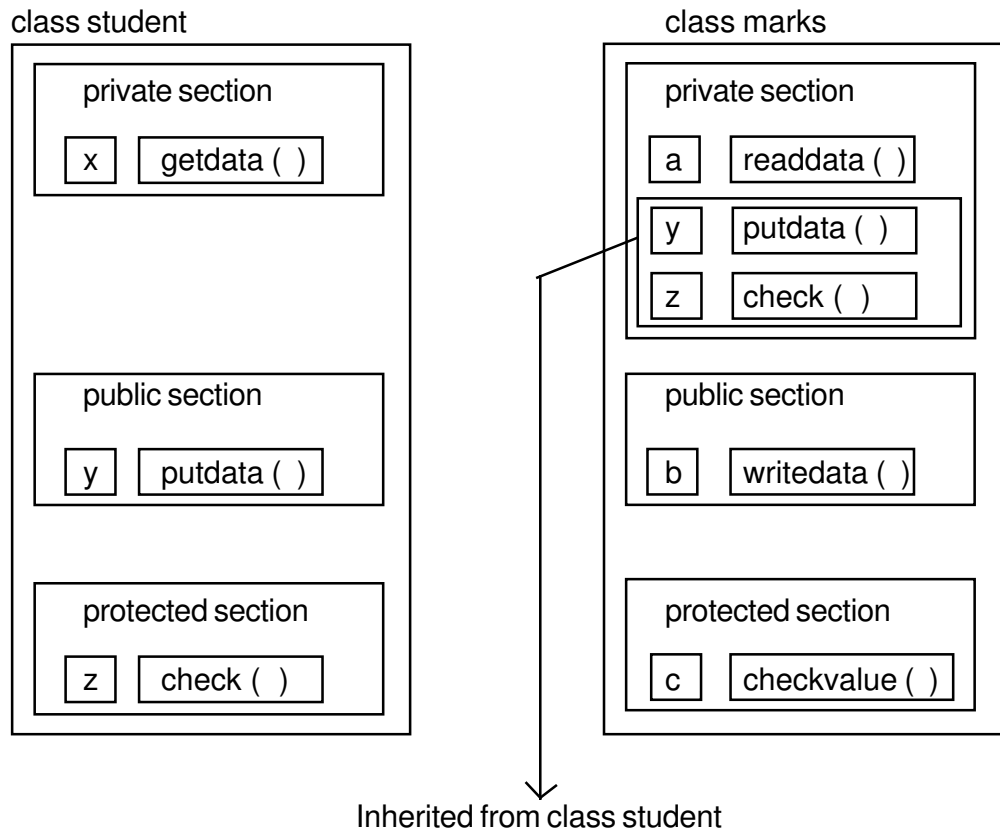


```

};
class marks : private student
{
 //
};

```

The following figure illustrates the private derivation in the classes.



**Fig 14.2 Private derivation of a class**

As it is clear from the figure that the data present in public and protected section of base class become the private members of derived class. The data in private section of base class cannot be inherited.

### The Protected visibility mode

We are using the same example but the derivation is done in protected mode.

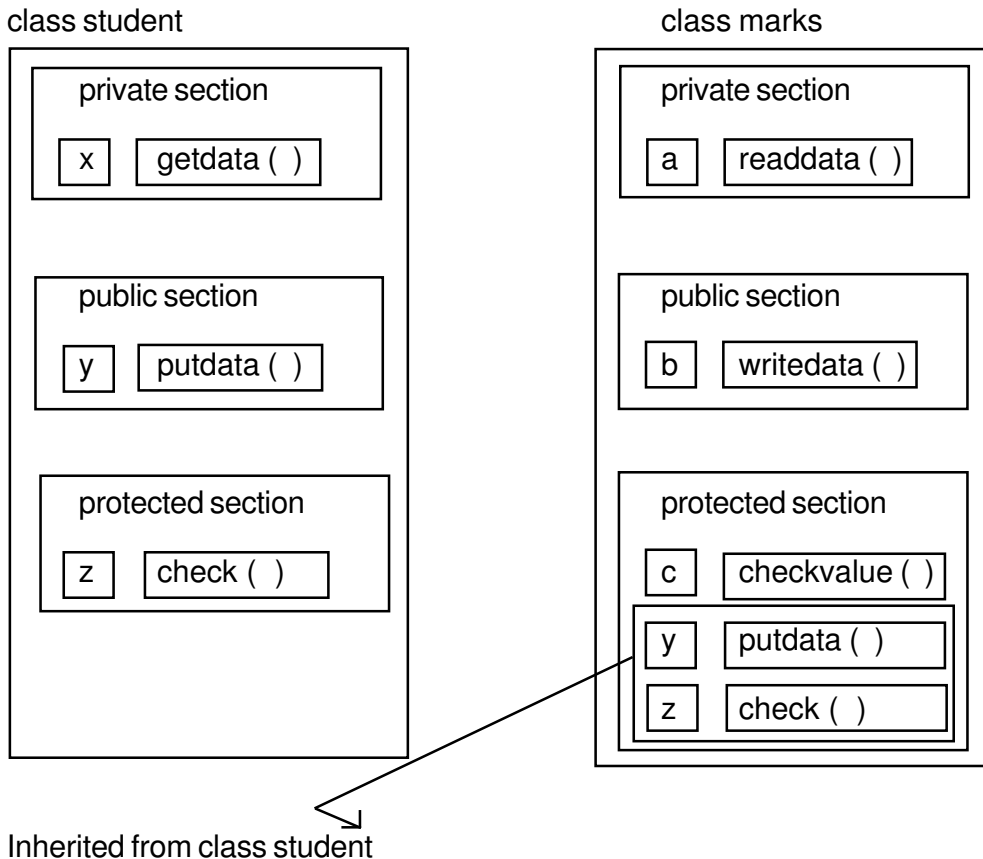
```

class student
{

```

```
// same as in previous example
};
class marks : protected student
{
};
```

The following figure illustrates the protected derivation in the classes.



**Fig. 14.3 Protected derivation of a class**

The data present in private section of base class cannot be inherited. The difference between private and protected section is that data present in protected section can be inherited. Otherwise both the section cannot be accessed by the object of the class.

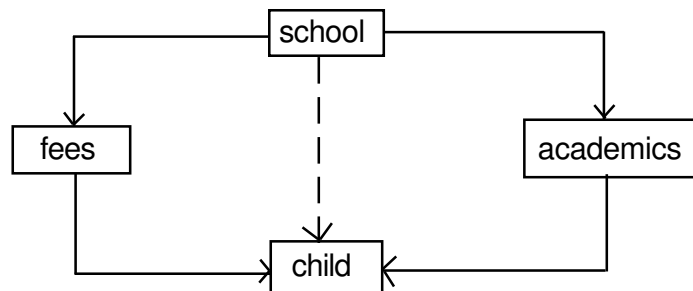
## 14.8 Abstract Class

An abstract class is one that is not used to create objects. An abstract class is designed only to act as a base class (to be inherited by other classes). It is a design

concept in program development and provides a base upon which other classes may be built. In the previous example, the class student is an abstract class since it was not used to create any object.

## 14.9 Virtual Base Class

Consider a situation where all the three kinds of inheritance, namely, multilevel, multiple and hierarchical instances are involved. Consider the following example illustrated in a figure.



**Fig. 14.4**

The child has two direct base classes fees and academics which themselves have a common base class 'school'. The child inherits the traits of 'school' via two separate paths. It can also inherit directly as shown by the broken line. The 'school' is sometimes referred to as indirect base class. All the public and protected members of 'school' are inherited into 'child' twice, first via 'fees' and again via 'academics'. This means, 'child' would have duplicate sets of the members inherited from 'school'. This introduces ambiguity and should be avoided.

The duplication of inherited members due to these multiple paths can be avoided by making the common base class as virtual class by declaring the base class as shown below :

```

class school
{

};
class fees : virtual public school
{

};

```

```
class academics : public virtual school
{

};
class child : public fees, public academics
{
 // only one copy of school will be inherited.
};
```

**Note** : you can write either virtual public or public virtual.

---

### **In-Text Questions 14.1**

---

1. What is inheritance? What are base class and derived class ?
  2. What are the different forms of inheritance?
  3. What are the three modes of inheritance?
  4. The private section of base class can be inherited. True or False?
  5. What is the difference between private and protected sections ?
  6. What is an abstract class?
  7. Fill in the blanks:
    - (a) The base class is also called .....
    - (b) The derived class can be derived from base class in ..... or ..... way.
    - (c) By default base class is visible as ..... mode in derived class.
    - (d) When a derived class is derived from more than one base class then the inheritance is called ..... inheritance.
  8. State whether the following are True or False.
    - (a) Inheritance means child receiving certain traits from parents.
    - (b) The default base class is visible as public mode in derived class.
-

- (c) When a derived class is derived from more than one base class then the inheritance is called hierarchical inheritance.
  - (d) The base class is called abstract class
  - (e) Private data of base class can be inherited
9. Consider the following class declaration and answer the questions given below:

```
class zoo
{
 char location [20];
 protected;
 int no_of_animals;
public;
 void inputdata (char, int);
 void outputdata ();
};
class animal : protected zoo
{
 int tail;
 protected;
 int legs;
public:
 void readdata (int, int) ;
 void writedata ();
};
class carnivorous : private animal
{
 int paw_size;
 public :
 void fetchdata (int);
 void displayed ();
};
```

- (a) Name the base class and derived class of the class animal.
  - (b) Name the data member(s) that can be accessed from function displayed ( ).
-

- (c) Name the data member(s) that can be accessed by an object of carnivorous class.
  - (d) Is the member function outputdata accessible to the objects of animal class ?
- 

## **14.10 What you have learnt**

In this lesson, you learnt about a very important aspect of object oriented programming i.e., inheritance. We discussed different forms and types of inheritance so that you should be able to use inheritance in C++ programming.

## **14.11 Terminal Questions**

1. What are the needs of inheritance?
2. What are the three modes of inheritance? Explain.
3. What is a virtual base class ? Explain it by taking an example.
4. Consider the following class declaration and answer the questions given below:

```
class vehicle
{
 int wheels;
 protected:
 int passenger;
public:
 void inputdata (int, int);
 void outputdata ();
};
class heavy_vehicle: protected vehicle
{
 int diesel_petrol;
 protected:
 int load;
public:
 void readdata (int, int) ;
 void writedata ();
```

---

```
};
class bus : private heavy_vehicle
{
 char make [20];
 public :
 void fetchdata (int);
 void displaydata () ;
};
```

- (i) Name the base class and derived class of the class heavy\_vehicle.
- (ii) Name the data member(s) that can be accessed from function displaydata.
- (iii) Name the data member(s) that can be accessed by an object of bus class?
- (iv) Is the member function outputdata accessible to the objects of heavy\_vehicle class?

---

## 14.12 Feedback to In-Text Questions

---

### In-text Questions 14.1

1. The mechanism of deriving a new class from an old one is called inheritance. The old class is referred to as the base class and new one is called the derived class.
  2.
    - (i) Single inheritance
    - (ii) Multiple inheritance
    - (iii) Multilevel inheritance
    - (iv) Hybrid inheritance
    - (v) Hierarchical inheritance
  3. private, public and protected
  4. False
-

5. The data in protected section can be inherited.
  6. The base class is called an abstract class.
  7.
    - (a) Abstract class
    - (b) public, private, protected
    - (c) private
    - (d) multiple
  8.
    - (a) T
    - (b) F
    - (c) F
    - (d) T
    - (e) F
  9.
    - (a) base class - zoo  
derived class - carnivorous
    - (b) legs, no-o-animals, paw\_size
    - (c) None
    - (d) No
-



# 15

## Pointer

### 15.1 Introduction

In C++, the programming with pointers is more powerful and it is used extensively. It saves the processing time. Pointer is a variable which holds the address of another variable. So, programming is concerned with the address, not with the direct value stored.

### 15.2 Objectives

After going through this lesson, you would be able to:

- use pointers in arrays
- define pointer variables in a structure and access data members through pointer
- define pointer objects in a class and access members through pointer

### 15.3 Pointer

A pointer is a variable that represents the location (rather than the value ) of a data item such as a variable or an array element. Pointers are used frequently in C++, as they have a number of useful applications. Consider the following example:

```
#include <iostream.h >
void main ()
{
```

---

```
int A = 5;
cout << & A;
int *ptr;
ptr = & A;
cout << *ptr;
}
```

If variable A in the above example has a value 5, then & A holds the address of memory cell A. The variable which holds the address is called pointer variable. int \*ptr means that a variable ptr is a pointer to a memory cell which can hold the int data type. \*ptr is used for the value stored in a memory cell pointed to by the pointer ptr. It is called de-referencing the pointer. The output of the above program is the address of memory cell A and value 5.

```
void *ptr;
```

Here ptr can point to any data type.

### 15.3.1 Pointer to Array

Consider the following declaration:

```
int A [5];
```

The name of the array A itself is a pointer which holds the address of zero location (&A[0]). It is a constant in a program, its value cannot be changed. The following program prints all the values of an array A.

```
include < iostream.h >
void main ()
{
 int [5] = { 20, 35, 25, 22, 27 };
 for (int i = 0; i < 5; i + +)
 cout << "\n" << A [i];
}
```

```
The output is 20
 35
 25
 22
 27
```

---

The above program can be written as pointer notation.

```
include < iostream.h >
void main ()
{
 int A [5] = { 20, 35, 25, 22, 27 }
 for (int i = 0; i < 5; i + +)
 cout << "\n" << *(A + i);
}
```

At one stage the value of i is 2. Therefore A + i is 2, i.e., two locations from the zero location. The \*(A+i) will print the data stored at that location.

### 15.3.2 Pointer to String constant

Consider the following example:

```
include < iostream.h >
void main ()
{
 char stu1 [] = "work as an array";
 char *stu2 = "work as a pointer";
 cout << stu 1;
 cout << stu2; // display both the strings
 stu1 ++; // it is a wrong statement
 stu2 + +;
 cout << stu2; // it prints "ork as a pointer"
}
```

stu 1 + + is a wrong statement because stu 1 is a pointer which holds the address of zero location ( & stu1 [0]). It is a constant in a program.

### 15.3.3 Pointer to Structure

Consider the following program

```
struct student
{
 char name [30];
```

---

```
 int rn;
 };
```

The statement

```
 student st;
```

declares st as the variable of the structure student.

The statement

```
 student *ptr;
```

declares a pointer variable ptr to a student.

That data members using ptr can be referred to as

```
 ptr -> name;
 ptr -> rn;
```

Another way of referring the data member is

```
 (*ptr) . name;
 (*ptr) . rn;
```

### 15.3.4 Pointer to Objects

Object pointers are useful in creating objects at run time. Object pointers can be used to access the public members of an object. Consider a class student defined as follows:

```
class student
{
 char name [20];
 int rn;
public :
 int marks;
 void getdata () ;
 void putdata () ;
};
void student :: getdata ()
{
```

---

```
 cin >> name;
 cin >> rn;
 cin >> marks;
 }
 void student :: putdata ()
 {
 cout << "Name" << << name << "\n";
 cout << "Roll no" << rn << "\n";
 cout << "Marks" << marks << "\n";
 }
```

Declare an object st and pointer ptr as follows:

```
student st;
student *ptr;
```

We can refer to the member functions and data member of student in two ways.

(i) using dot operator

```
st. marks = 90;
st. getdata ();
```

(ii) using arrow operator and object pointer

```
ptr -> marks = 90;
ptr -> getdata ();
```

another way to represent is (\*ptr)

```
(*ptr) . marks = 90;
(*ptr) . getdata ();
```

## 15.4 This Pointer

C++ uses a unique keyword called *this* to represent the object that invokes a member function. *This* is a pointer that points to the object for which *this* function was called. For example;

```
class ABC
{
```

---

```
 int rn;
public:
 void getdata ()
 {
 cin >> this -> rn;
 }
 void putdata ()
 {
 cout << this -> rn;
 };
void main ()
{
 ABC A, B;
 A . getdata ();
 A . putdata ();
 B . getdata ();
 B . putdata ();
}
```

When a `getdata ( )` or `putdata ( )` function is called through object A, *this* has the address of object A. Similarly, when a `getdata ( )` or `putdata ( )` function is called through object B, *this* has the address of object B.

---

## In-Text Questions 15.1

---

1. How is a pointer variable different from ordinary variable ?
  2. A pointer is :
    - (i) the address of a variable
    - (ii) an indication of the variable to be accessed next
    - (iii) a variable for string addresses
    - (iv) the data type of address variable
  3. If a pointer `ptr` points to a variable A, write a statement that represents the content of A but does not use its name.
  4. The expression `*ptr` can be said to
-

- (i) be a pointer to ptr
  - (ii) refer to the contents of ptr
  - (iii) refer to the value of the variable pointed to by ptr
  - (iv) dereference ptr
5. The expression `int*` can be said to
- (i) be a pointer pointing to variable `int`
  - (ii) refer to contents of `int`
  - (iii) be a pointer to a `int` type value.
6. Fill in the blanks:
- (a) A pointer variable stores the ..... of another variable
  - (b) The ..... operator is called address operator.
  - (c) To declare `t_age` pointer to integer, we should write .....
  - (d) If `A` is an integer variable than ..... gives its address.
  - (e) If `ptr` is a pointer to an integer variable, the number stored in it is given by .....
7. State whether the following are True or False.
- (a) A pointer is a address of the variable.
  - (b) Dereferencing operator (`*`) is a unary operator. It is different from the multiplication operator (`*`) which needs two operands.
  - (c) this pointer points to the objects that is currently used to invoke a function.
- 

## 15.5 What you have learnt

Having gone through the present lesson you learnt the use of pointer in variable array, structure variable and class object. A special pointer is available in C++ called this pointer. We have learnt the use of this special pointer also.

---

## 15.6 Terminal Questions

---

1. If `arr` is an array of integers, why is the expression `arr++` not legal?
-

2. What is the difference between `arr [4]` and `*(arr+4)`?
3. If a structure defined has pointer variable, how can it access the members of the structure ? Explain if by taking an example.
4. How a data member and member function present in public in class accessed through pointer object? Explain it by taking an example.
5. What is this pointer? Explain briefly?

---

## 15.7 Feedback to In-Text Questions

---

### In-text Questions 15.1

1. The pointer variable stores the address whereas ordinary variable stores the value.
  2. (i)
  3. `*ptr`
  4. (iv)
  5. (iii)
  6. (a) address  
(b) `&`  
(c) `int *t_age;`  
(d) `&A`  
(e) `*ptr`
  7. (a) T  
(b) T  
(c) T
-



# 16

## Files

### 16.1 Introduction

At times it is required to store data on hard disk or floppy disk in some application program. The data is stored in these devices using the concept of file.

### 16.2 Objectives

After going through this lesson, you would be able to:

- store data in a file
- access data record by record from the file
- move pointer within the file
- open or close file

### 16.3 File

A file is a collection of logically related records. A program usually requires two types of data communication.

#### (i) Writing data on the datafile:

The data flows from keyboard to memory and from memory to storage device.

---

keyboard → memory → hard disk/floppydisk

This is called output stream where stream is the flow of data and requires an `ofstream.h` header file.

**(ii) Reading data from datafile:**

The data flows from storage device to memory and from memory to output device, particularly monitor.

datafile → memory → output device (screen)

external storage

device

(hard disk/floppy)

This is called input stream and requires `ifstream.h` header file.

If both input stream and output stream are used in the same program then header file `fstream.h` is required.

If header file `fstream.h` is included in the program, there is no need to include `iostream.h` explicitly.

### 16.3.1 Opening a file

A file can be opened in two ways:

- (i) using constructor function of a class.
- (ii) using the member function `open ( )` of the class.

Opening a file using constructor function

The following statement opens the file `STU.DAT` in output mode, i.e., for writing data on the file.

```
ofstream outfile ("STU.DAT");
```

`ofstream` is a class available in the compiler file.

`outfile` is any user defined object.

The statements

---

```
outfile << "TOTAL MARKS" << "\n";
outfile << total << "\n";
```

are used for writing data on the file. The newline character is used for moving the pointer to the next line.

Similarly, the following statement

```
ifstream infile ("STU.DAT");
```

opens the file "STU.DAT" in input mode, i.e., for reading purpose

The statements

```
infile >> string;
infile >> number;
```

read the data from the data file.

The following program uses a single file for both writing and reading purposes. First, it takes the data form the keyboard and writes it to the file. After the writing is completed, the file is closed. The program again opens the same file, reads the information already written to it and displays it on the screen.

```
include < fstream.h >
void main ()
{ char name [30];
int rn, marks;
ofstream outfile ("INF");
cout << "Enter student name";
cin >> name;
cout << "Enter student roll number";
cin >> rn;
cout << "Enter student marks";
cin >> marks;
outfile << name << "\n";
outfile << rn << "\n";
outfile << marks << "\n";
outfile . close ();
```

---

```
ifstream infile ("INF");
infile >> name;
infile >> rn;
infile >> marks;
cout << "Name" << name << "\n";
cout << "Roll no" << rn << "\n";
cout << "Marks" << marks << "\n";
infile close ();
}
```

The output of the program would be:

```
Enter student name PARAM
Enter student roll number 20
Enter student marks 90
Name PARAM
Roll No. 20
Marks 90
```

### Opening a file using open ( ) function

The function open ( ) can be used to multiple files that use the same stream object. First a stream object is assigned to and then it is used to open the file in turn.

```
filestream_class stream_object;
stream_object . open ("filename");
```

For example :

```
ofstream outfile;
outfile . open ("ABC");

outfile . close ();
outfile . open ("XYZ");

outfile.close ();
```

### 16.3.2 Open ( ) Function

The open ( ) function has two parameters : filename and access mode. The general

---

format is:

```
stream_object . open ("filename", access mode);
```

The second argument specifies the mode in which the file is opened. The default values are taken for ifstream or ofstream functions. (the mode is not defined explicitly).

ios :: in for ifstream functions

ios :: out for ofstream functions

The file mode parameters can take one or more of the constants defined in the class ios. The following table shows the file mode parameters.

| Parameter        | Meaning                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| ios :: app       | It opens the file in output mode. The file pointer is at the end of file and it can add a record.                                                       |
| ios :: ate       | The file pointer is at the end of the file and it allows to add data or to modify the existing data anywhere in the file.                               |
| ios :: binary    | Binary file                                                                                                                                             |
| ios :: in        | It opens the file in input mode. The file pointer is at the top of the file and it is ready for reading.                                                |
| ios :: nocreate  | If file is already present, it opens the file otherwise open fails.                                                                                     |
| ios :: noreplace | If file is not present, it opens the file otherwise open statement fails.                                                                               |
| ios :: out       | It opens the file in output mode. The file pointer is at the end of the file. If it already has a data, the output mode erases the content of the file. |
| ios :: trunc     | It deletes the contents of the file if exist.                                                                                                           |

---

The mode can combine two or more parameters using bitwise OR operator.

**Example:**

```
outfile . open ("ABC", ios::in | ios::out | ios :: binary);
```

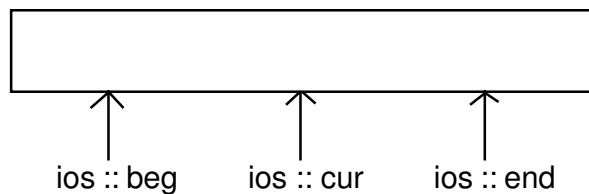
### 16.3.3 File Pointers

File has two associated pointers called input pointer (or get pointer) and output pointer (or put pointer). Each time an input or output operation takes place, the pointer moves automatically. There are two pointers.

`seekg ( )` It moves get pointer to a specified location.

`seekp ( )` It moves the put pointer to a specified location.

A file can be viewed as



`ios :: beg` means start of the file

`ios :: cur` means current position of the pointer

`ios :: end` means end of the file

The `seekg ( )` and `seekp ( )` statement has two parameters.

```
object . seekg (no. of bytes, reposition);
```

```
object . seekp (no. of bytes, reposition);
```

The reposition takes one of the above three constants defined in the `ios` class.

**Example 1**

```
infile.seekg (0, ios::beg);
```

It moves the pointer to the beginning of the file. In this case, the reposition `ios :: beg` is optional.

```
infile.seekg (100, ios::cur);
```

It moves the pointer 100 bytes forward from the current position.

```
infile.seekg (-200, ios::end);
```

It moves the pointer 200 bytes backward from the end of the file.

### 16.3.4 The tellg ( ) and tellp ( ) function

The tellg ( ) function gives the position of get pointer in terms of number of bytes. Similarly, tellp ( ) function gives the position of put pointer in terms of bytes.

#### **Example 2**

```
ifstream infile;
infile . open ("ABC", ios ::ate);
int B = infile . tellg ();
```

On execution of the above statements, the input pointer is moved to the end of the file and B gives the number of bytes in the file.

### 16.3.5 Write ( ) and read ( ) functions

The functions write ( ) and read ( ) has two parameters: address of the variable, size of the variable. The address of the variable must be cast to the type char\*. The general format is:

```
infile . read ((char*) & v, sizeof v);
outfile . write ((char*) & v, sizeof v);
where v is the variable.
```

### 16.3.6 Close ( ) function

The file should be closed at the end if it is opened either through constructor or open ( ) function. The General format is stream\_object.close ( );

The following example works with class object and does the following operations:

- Create a data file
  - Display a data file
  - Adding a new record
  - Modify the existing record
-

---

**Example 3**

```
include <fstream.h>
class student
{
 char name [30];
 int rn;
public:
 void getdata ();
 void putdata ();
};
void student :: getdata ()
{
 cout << "Enter student name";
 cin >> name;
 cout << "Enter roll number";
 cin >> rn;
}
void student :: putdata ()
{
 cout << "Student name" << name << "\n";
 cout << "Student roll number" << rn << "\n";
}
void main ()
{
 fstream file;
 file . open ("ABC", ios::in | ios::out | ios::binary);
 student st;
 // create a data file
 int i, n;
 cout << "How many record to enter";
 cin >> n;
 for (i = 1; i <= n, i ++)
 {
 st. getdata ();
 }
}
```

---



```
 file . write ((char*) & st, sizeof st);
 }
 // Display a data file
 file . seekg (0, ios::beg);
 while (file . read ((char*) & st, sizeof st))
 {
 st. putdata ();
 }
 file . clear () // To make the end of file mark false
 // To append record
 st . getdata ();
 file . write ((char*) & st, sizeof st);
 // To modify a record
 file.clear ();
 cout << "Enter record number";
 cin >> n;
 file . seekp ((n - 1)* sizeof st, ios::beg);
 st. getdata ();
 file.write ((char*) & st, sizeof st);
 // To close a file
 file . close ();
}
```

---

## In-Text Questions 16.1

---

1. What is file?
  2. What is the difference between ios::ate and ios::app modes?
  3. What are the two file pointers?
  4. The ios::out mode is ios::trunc mode by default. True or false.
  5. What is the significance of tellg ( ) function?
  6. Fill in the blanks:
    - (a) A..... is a collection of logically related records.
-

- 
- (b) The file open in ofstream is only available for .....
  - (c) The file open in ifstream is only available for .....
  - (d) We can open the file using ..... function.
  - (e) The mode app opens the file for .....
  - (f) The file open in output mode is ..... by default.
7. State whether the following are True or False.
- (a) A file is a collection of record.
  - (b) The file open in ofstream is available for writing.
  - (c) The output mode of opening a file deletes the contents, if present in the file.
  - (d) The close ( ) function is used to close a file.
  - (e) The statement  

```
outfile.write ((char*) &obj, sizeof obj);
```

writes only data in obj to outfile.
  - (f) The ios::ate mode allow us to write data at the end the file only.
  - (g) We can add data to an existing file by opening in append mode.
  - (h) The data written to a file with write ( ) function can be read with the get ( ) function.
- 

## 16.4 What you have learnt

In this lesson you learnt about the file programming which is a very important aspect of any language. We have provided you with examples that will help you in writing programs using files.

## 16.5 Terminal Questions

1. What are input and output stream?
  2. What is the difference between opening a file with constructor function and
-

opening a file with open () function?

3. What is the file access mode? Describe the various file modes.

4. A file consists of 5 records, each takes 100 bytes of storage:

fstream file;

file.seekg ( 0, ios::end);

N = file.tellg ( );

(i) What will be the datatype of N?

(ii) What will be the value of N?

5. Consider the following statements:

fstream file;

file.open ("ABC", ios::in | ios::out);

Write C++ statement(s) for the following:

(i) To move the pointer at the beginning of file.

(ii) To move the pointer at the end of file.

(iii) To find the total number of bytes.

(iv) To close the file.

6. Explain the functioning of the following:

fstream file;

(i) file.seekg (100, ios::cur);

(ii) file.seekg (-100, ios::end);

(iii) file.seekg ( 100, ios::beg);

7. The record consists of two fields: name and rollno. Write a program that will perform the following:

(i) create a data file of 5 records

(ii) display a data file

(iii) append a record

(iv) modify one of the records

---

## 16.6 Feedback to In-Text Questions

---

### In-text Questions 16.1

1. It is a group of logically related records.
  2. The ios::ate mode can add or modify an existing record whereas ios::app mode can only add records.
  3. put and get
  4. True
  5. It gives the position of get pointer in terms of bytes.
  6.
    - (a) file
    - (b) output
    - (c) input
    - (d) open
    - (e) output
    - (f) trunc
  7.
    - (a) T
    - (b) T
    - (c) T
    - (d) T
    - (e) T
    - (f) F
    - (g) T
    - (h) F
-