COMPUTER APPLICATION STUDY MATERIALS

(As per SCTE&VT,Odisha - Syllabus)

Prepared by :

Sri Rabindra Kumar Rout ,

Lecturer in Computer Application

GOVERNMENT POLYTECHNIC, KENDRAPARA

INDEX

CHAPTER - I (COMPUTER ORGANISATION) :

Applications of Computers / Uses of Computers in real life :-

Today's world is an information-rich world and it has become a necessity for everyone to know about Computers.

Every activity of human being is being controlled by Computers. It has become an essential commodity for human being. There is no such area where computer are not being used. We can find computers in a **Business organisation** for payroll, budget, sales analysis, financial management, managing employees database, maintenance of stocks etc. In **Banking** it is used for online accounting, deposits, overdraft, interest charges, ATM machines are making it even easier for customers to deal with banks. Computers can provide a lot of facilities for **Education** to the society. Computers are used for marketing such as advertisements and on-line shopping. Computers use in **Health care** system has become very very important and used for Patient Monitoring, suggesting proper diagnosis. Computers are widely used in Engineering design. In **Industries**, Robots are used for industrial process control and also used in specific areas where human life is being risky to work. In **Defence & scientific research**, missile control, satellite simulation, orbital analysis, weather forecasting can not be possible without the help of computers. Development in **Communication** without computer is just impossible in the areas like : E-mail, Chatting, FTP, Telnet, Video-conferencing etc. In **Government** sector some major fields in this category are : e-Governance, Budgets, Sales tax department, Income tax department, computerization of voters lists, Computerization of driving licensing system, Computerization of PAN card etc. **Travelling** has made easier to move from one place to another with the help of computers. Due to development of **multimedia** technology we can get image, video & audio combination on a computer just as we see in Television for entertainment purpose. Information is available to all over the countries through internet and the world has become a small family.

Gone are the days when highly qualified people like scientists or research scholars were able to use computers. But now-a-days a school going child of standard-I is using computer. Personal computers have been flooded through out the world and made information revolution.

Necessity of computers were felt during world war-II and scientists have developed first electronics computer called ENIAC which was huge size but to days computer are kept in a pocket called notebook which is much powerful than previous one.

Introduction to Computer ?

The word "computer" comes from the word compute which means to calculate derived from latin term computare. Computer is an electronics device that is designed to work on data and produce the information. It is a machine that accept data as input , process as per instructions or program and produce the result or output. It can not do any thing without set of instructions or program. Computer represents decimal numbers, alphabets or characters in strings of 0's and 1's i.e. in the form of binary digits or bits. Charles Babbage is called father of computer because he developed first mechanical computer called analytical engine which uses the concept of input, process and output.

Define Computer : Computer is an electronics device which accept the data as input, process it as per the instructions or Programs given and produce the desired output or result.

Basically three steps involved : INPUT ⟹ PROCESS ⟹ OUTPUT

Computer is a device which operates on data and instructions/programs..

Computer Organisation : Structure and behaviour of computer system. It's operational units and their interconnections.

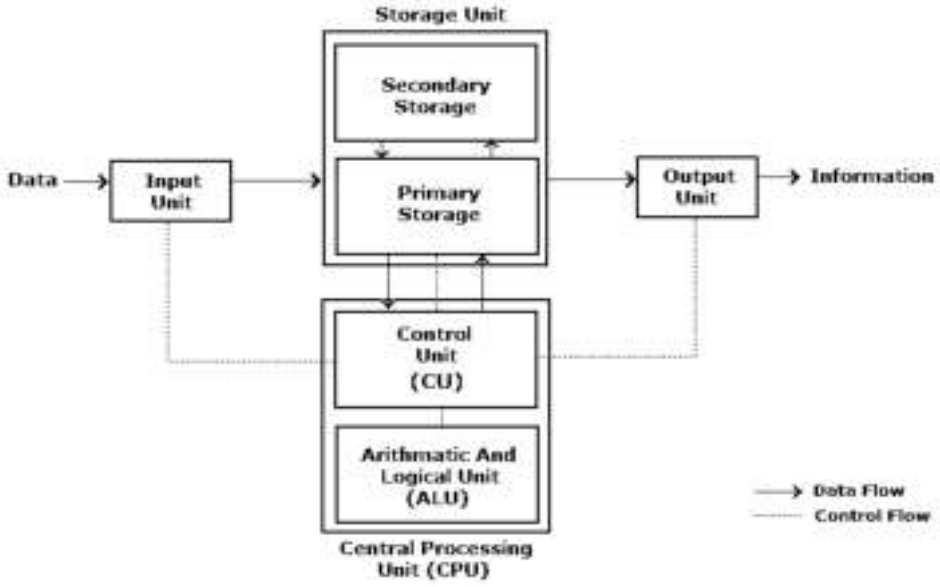
Computer Architecture : Design of hardware components of computer.

What is Data :- Data can be defined as representation of facts. It is raw material for data-processing.
Ex:- Names and address of the students in a class. It may be a text, picture or video etc.
Data is represented with the help of characters like alphabets (A-Z, a-z), digits (0-9) or special characters (+,-,/,*,<,>,= etc.).

What is information :- The data after processing becomes information. The names and address of the students in a class arranged in alphabetic order becomes information. So, it is a meaningful data or Information is the processed data on which decisions and actions are taken.

Data Processing :- Computers are also called data processing machines.

Data processing is the re-structuring or re-ordering of data by people or machine to increase their usefulness and add values for particular purpose. Data processing consists of basic steps input, processing and output. These three steps constitute the data processing cycle.



Any computer is having five units:- Input, Output, Memory, ALU & Control Unit. This concept was given by John Von Neuman. The program and data to be executed are stored in memory. The CPU fetches the instruction one-by-one, decode it and then execute it. This fetch-decode-execute concept was also given by Von Neuman .

Input :- It is the device which converts data & programs into format that is understood by Computer. The process of entering data and instructions into the computer is called inputting.
Input Unit : The following functions are performed by input unit.
It accept data or instructions from outside the world and convert it into computer acceptable form.
Ex: Keyboard, Mouse
Memory : The storage area of computer is called memory. Saving data and instructions to make them readily available for processing as and when required is called storing.
Storage Unit : The specific functions of storage unit are to hold data or instructions required for processing and also hold intermediate results or final result before or after processing. The storage unit of all computer is consists of 2 types i.e. primary storage and secondary storage.
Primary Storage : Also known as Main Memory. It is used to hold data, instructions or intermediate results of processing of the jobs. As soon as the computer system is switched off the information available in primary storage are disappeared because of that the materials made of this memory is semiconductor devices like germanium or silicon. It is also called semiconductor memory or volatile memory. It is limited capacity because it is expensive. Ex-RAM, ROM
Secondary Storage : It is also known as auxiliary storage /secondary memory/permanent memory/ non-volatile memory used to keep data or information permanently. It is cheaper.
Ex-Magnetic Disks, HDD, Pendrives, CD/DVD

Output :- It converts data or results into user understandable format. The process of producing useful information or results for the user is called outputting.
CPU :- It fetches instruction or data from memory and act as desired & then store it in to the memory then it sends to output devices. Performing arithmetic operations or logical operations on data to convert them into useful information is called Processing.
Arithmetic Logic Unit : All the arithmetic and logical comparison are performed by ALU.
Control Unit :- Controls & communicate with memory & Input / Output devices. Directing the manner and sequence in which all of the operations like inputting, storing, processing and outputting operations are performed is called controlling. Control Unit acts as a central nervous system for the other components of the computer system, it manages and co-ordinates the entire computer system.

System or Computer System is a group a computer vital parts integrated(Logical relationship among parts for the purpose of achieving some goal or objective.

Advantages of computer/ Characteristic of Computer :-
High Speed :-
Since the computer is an electronics device, it operates at a speed comparable to the speed of electron flow. Capable of performing calculation of very large amount of data in speed as measured in microsecond (10^{-6} second) , nanosecond(10^{-9} second) , and even the picosecond (10^{-12} second).

Accuracy :-
In addition to being very fast, computers are very accurate. The calculations are 100% error free.

Storage Capability :-
Memory is a very important characteristic of computers. A computer has much more storage capacity than human beings. It can store large amount of data. It can store any type of data such as images, videos, text, audio and many others.

Diligence :-
Unlike human beings, a computer is free from monotony, tiredness and lack of concentration .It can work continuously without creating any error. It can do repeated work with same speed and accuracy.

Versatility :-
A computer is a very versatile machine i.e it can perform different types of work. At one instance, it may be solving a complex scientific problem and the very next moment it may be playing a card game i.e. it can perform different types of work at the same time.

Reliability :-
A computer is 100% reliable machine. Modern electronic components have long lives. Computers are designed to make maintenance easy.

Automatic :-
Computer is an automatic machine. Automation means ability to perform the given task automatically. Once a program is given to computer i.e., stored in computer memory, the program and instruction can control the program execution without human interaction.

Reduction in Paper Work :-
The use of computers for data processing in an organization leads to reduction in paper work.

Reduction in Cost :-
Though the initial investment for installing a computer is high but it substantially reduces the cost of each of its transaction.

Disadvantages :-
No Intelligence :- A computer is a machine that has no intelligence to perform any task.
Each instruction has to be given to computer. A computer cannot take any decision on its own.
Dependency :- It is dependent on human being.
Environment :- The operating environment of computer should be dust free and suitable.

Comparison between Hardware and Software :-

Hardware:

1. Physical parts of the computer are called hardware.
   Ex- Keyboard, Mouse, Monitor etc. Hardware can not function without software.
2. You can touch, see and feel hardware.
3. Hardware is constructed using physical materials or components.
4. Computer is hardware, which operates under the control of a software.
5. If hardware is damaged, it is replaced with new one.
6. Hardware is not affected by computer viruses.
7. Hardware cannot be transferred from one place to another electronically through network.
8. User cannot make new duplicate copies of the hardware.

Software:

1. A set of instructions given to the computer to perform some specific task.
   Ex. Windows Operating System, MSDOS, Linux, Unix, MSOffice, Oracle, Pagemaker, Coreldraw
2. You cannot touch and feel software.
3. Software is developed by writing instructions in programming language.
4. The operations of computer are controlled through software.
5. If software is damaged or corrupted, its backup copy can be reinstalled.
6. Software is affected by computer viruses.
7. Software can be transferred from one lace to another electronically through network.
8. User can make many new duplicate copies of the software.
   Software cannot be utilized without hardware. A program is a sequence of instructions written to solve a particular problem.

Evolution of Computer :-

Evolution refers to historical development through which computer and technology have passed from time to time and started from ancient period to current state.
The technological development in the field of computer not only refers to improvement made to hardware but also improvement to software .
Increasing need for numerical calculations, storage of data and information etc. with minimum of mental and manual efforts led to invention of computers.

ABACUS : Abacus (Probably developed in China), used for calculations, is the earliest/first device way back around 5000 BC still being used in china and japan. Abacus is basically a rectangular wooden frame with beads stung on wire or rods. Arithmetic calculation are performed by manipulating the beads from one end to anther.

Napier's Bones : John Napier a Scottish mathematician in 1617 developed Logarithm, a tabular system of numbers through which many arithmetical calculations were simplified. It is device contains a set of rods made of bones and used for multiplication & division called Napier Bones.

William's Slide Rule : Based on the principle of logarithm, a calculating device was invented by William in 1620.

Pascaline / Pascal's Calculator : Blaise Pascal, a French mathematician invented the first adding machine in 1642 that could add and subtract. It was first device to perform additions and subtraction on whole numbers. Automobile Audometer is an example of device still used to calculate data.

Babbage's Difference Engine/Analytical Engine:- In 1822, Charles Babbage designed a prototype computer called Difference Engine that was 100 years ahead of it's time that would incorporate punched card as input, Process and output .

In 1833, Charles Babbage designed and worked on Analytical Engine. It was a general purpose computer designed to solve almost any type of problem. It contained most of the elements we find in modern digital computer systems. Hence, Charles Babbage is considered as **Father of modern computer**. Lady Augusta Ada Lovelace daughter of Lord Byron was the first lady programmer and in her name Ada programming language is named.

Punch Card System : In 1880, Dr. Herman Hollerith an American Statistician in US Census Bureau developed Punched Card (earliest form of computer input) for data processing by building a tabulating machine that could punch holes in cards and read them as well for storing and retrieving data. The system was very slow and card jams and data destruction were common problems. Now a days it is obsolete.
Joseph Jacquard invented punch cards in 1801. Punch card had holes punched in it. These were used by him to produce weaving pattern on the cloths.

In 1944, Howard Alken built MARK1, the first digital computer, at Harvard University. It lacked speed but had the ability to continuously perform complex arithmetic functions without frequent human intervention. It was partly electronic and partly mechanical machine.

In 1942, USA started to develop an electronic computer. In 1946, it could put to operation 'ENIAC' (Electronic Numerical Integrator and Calculator), made in University of Pennsylvania. John Mauchly and J.Presper Eckert were the two people involved in its development. This computer was made of 18,000 vacuum tubes. ENIAC could process the data at great speeds (though not comparable to today's computers). UNIVAC-1 was the first business oriented computer developed in 1952 used by US Bureau of Census.

Vacuum Tubes :- First electronics computers used vacuum tubes for circuitry & magnetic drum memory, Input was punched cards & paper tapes and output displayed on printouts. Example of Computers are ENIAC, EDSAC, EDVAC,UNIVAC
Transistors :- It is a semiconductor device used to amplify & switch electronics signals. It is made up of solid piece of semiconductor. Invented transistors during 1950's replace vacuum tubes and become smaller and cheaper.
Integrated circuits (IC) :- Combination of 1000's of transistors in a small Chip made up of semiconductor material such as silicon. Computer are very small and place on a table called desktop and made Personal Computer (PC) revolution because small and portable .
Very Large Scale Integration (VLSI) :- 100's of IC into a small chip called VLSI.
History and 'Generations of Computers' :-

A generation is the period of time over which the major change in technology has made in computer, mainly to distinguish between varying hardware technologies and software. Each generation has its own advantages and disadvantages.

There are totally five computer generations known till date.

First Generation Computers : - The period of first generation: (1940-1956)
Vacuum Tubes were used as basic components of the computer. Magnetic drum as main memory. In this generation batch operating system were used.
Major innovations or Technology : Vacuum tubes/Valves as main electronics components. Vacuum tubes/ Valve is an electronics device were used to amplify electronic signals.
Speed :- Speed was measured in mili-seconds i.e. 1/1000 sec. Fastest at that time.
Programming language : Instruction were written in Machine language or low level language using 0's and 1's or binary digits (BIT).
Input / Output devices:- Punched card, paper tape, magnetic drum
Disadvantages: Large & bulky and non portable; lot of heat generated and so air-conditioning compulsory; frequent hardware failures and maintenance. Developed many problems and required frequent maintenance. Performed at low speed, occupied large space, consumed more power and needed air conditioning and very expensive.
Example : ENIAC(Electronic Numerical Integrator and Calculator) computer developed by USA. EDVAC,EDSAC(Electronic Delayed Storage Automatic Computer) was another major development; developed at Cambridge University of England; first introduced in May, 1946.

Second Generation computers:- The period of second generation: (1956-1963)
Transistors were used as major component and Magnetic core as main memory and magnetic tapes, disks used to store bulk data.
Speed :- Faster than first generation i.e Speed measured in micro-seconds i.e. 1/100000 sec
Major innovations or technology : Solid State Devices or Transistors used in place of vacuum tubes Transistors were smaller, faster, less expensive , more reliable , much greater processing capacity and generated less heat compared to vacuum tubes. Transistors are made up of materials called semiconductor.
Programming Language :- Assembly Language
Example : UNIVAC (Universal Automatic computer) was the first commercial business computer
Example : PDP-8, PDP-11, IBM 1401, IBM 1410, IBM 3070
Advantages : Flaws of first generation computers removed; smaller in size; Faster and more reliable; commercial production easier and cheaper and consume less electricity. Symbolic or Assembly Language used as instructions.
Disadvantages : High cost, limited to special task . Air conditioning required; Manufacturing process involved manual assembly and commercial production of these computers were difficult.

Third Generation computers: The period of third generation: (1964-1971)
Used Integrated circuits (IC) in place of Transistors. Single IC can have many transistors.
Major innovations or technology :- Integrated Circuit (one chip contains 1000;s of transistors), improved secondary storage devices, new input/ output devices like high speed printers and secondary storage located outside computer. Several electronic components such as transistors, registers and capacitors were placed in a silicon chip. Integrated circuits contains about 10-20 components called Small Scale Integration (SSI) and later it become possible to integrate about 100 components on a single ship named Medium Scale Integration (MSI). In this generation, multiprocessing, multiprogramming and Time Sharing Operating System were developed.
Speed measured in nano-seconds. Faster than 2nd generation.Used LSIC (Large Scale Integrated Circuits) in place of transistors. IC was small and one IC could do the job of hundreds of transistors.
Development of software (called Operating System) for efficient use of computer.
Programming language :- High level programing language like FORTRAN, PASCAL
Example : Burroughs-5700, 6700, IBM-360/370 series, PDP-8, PDP-11, UNIVAC 1100

Advantages : Smaller in size; faster, more reliable, cheaper and portable; low maintenance cost, lesser power required and widely used for scientific and business application. High Level Languages used for writing instructions/programs.
Disadvantages : Required highly sophisticated technology for manufacturing ICs & difficult to maintain.
Fourth Generation Computers:- The period of fourth generation: 1971 to at present
Major innovations are LSI (Large Scale Integration) which is integrated over 30,000 electronic circuits on a single chip and then followed by VLSI (very Large Scale Integration) about 1 million electronic components on a single chip and further this technology leads to creation of Microprocessors. In this generation GUI, Mouse, and hand hold devices are used. Improved mass storage, Input/ Output devices. Price reduced and affordable and became a common equipment in small business establishments.
Advantages : Size reduced substantially; air conditioning not required; operations much faster; more general purpose; cheapest amongst all previous generations; desktops made available for business/personal use and portable. Faster and larger primary and secondary memory.
Disadvantages : They were not intelligent system.

Fifth Generation Computers :- The period of fifth generation: At present and beyond it. Artificial Inteligence (AI)
Major innovation : Based on ULSI (Ultra large scale integration) technology. These computers are completely based on the new concept of artificial intelligence. It uses parallel processing OS. Artificial intelligence touches areas like expert system, natural language, robotics etc. Tremendous processing power and speed. Support high degree of artificial intelligence (computers will be able to think and learn and act on their own to some extent) . Interact with humans in ordinary languages
Advantages : More user-friendly; reduced size; easy to operate; economical and faster; programming much easier and simpler; emphasis shifted to networks and client server technology.
Ex.- IBM notebook

Explain the classification/ types of computers.
Computers are classified into 3 broad categories basing on – type, purpose, capacity.

Types of Computer :
Digital computers : Digital computer is designed to process the mathematical operations like addition subtraction multiplication and division. A digital computer is designed to process the data in numerical form, its circuit performs mathematical operations of addition, subtraction, multiplication and division. The numbers operated on by a digital computer are expressed in binary system . It is a general purpose computer that works on digital or binary data. Digital computer is capable of solving problems by processing information in discrete form/fixed. It operates on data, including letters, and symbols, that are expressed in binary digits (bits) i.e., using only the two digits 0 and 1.

Analog computers : Analog computer represents the data as physical quantities and operates on the data by manipulating the quantities. It is designed to process data in which the variable quantities vary continuously used to measure physical quantities or that uses continuous values mostly used industries in process control activities. Work on analog data such as temperature, pressure, humidity, speed, voltage etc. A service station gasoline pump contains an analog processor that converts fuel flow measurement into quantities & price values.
It measures continuous changes in some physical quantity : Ex-the Speedometer of a car measures speed, the change of temperature. Analog computers are widely used for certain specialized engineering and scientific applications, for calculation and measurement of analog quantities. They are frequently used to control process such as those found in oil refinery where flow and temperature

measurements are important. Analog computers do not require any storage capability because they measure and compare quantities in a single operation. Output from an analog computer is generally in the form of readings on a series of dial (Speedometer of a car) or a graph on strip chart. Uses are limited.

Hybrid computers : Hybrid computer is combination of analog and digital computer i.e. having desired the features of analog and digital computers. It combines the best features of both types of computer i.e. has the speed of an analog computer and the memory and accuracy of digital computer. It accepts a continuously varying input , which is then converted into a set of discrete values for digital processing. Input is generally in the form of analog form like heat/pressure etc., measured by analog part of computer and then used by digital part for further operations. Ex:- For example a petrol pump contains a processor that converts fuel flow measurements into quantity and price values. In hospital, analog device measures a patient's heart function, temperature and other functions. These measurement are then converted into numbers and supplied to Digital computer that monitor the patient's vital signals. Computers used in factories for controlling manufacturing processes, launching a rocket etc.

Digital Computers are classified into 4 Types :
Computers/ sizes of computer can be broadly classified by their data processing capability.
i.e. 1) Amount of data can be stored in memory 2)Speed of internal operation
3)Types of software it support 4)Types of peripherals it support etc.

Microcomputer : A microcomputer is a small, relatively inexpensive computer with a microprocessor as its central processing unit (CPU). Portable, require minimum power and support various kind of OS. It includes a microprocessor, memory, and input/output (I/O) facilities. They are quite capable of handling small business applications. It is also a single user computer system which is similar to personal computer. Microcomputers are mainly known as Desktop PC used for single use and in a single location. Laptop are portable computers are having same configuration like desktops and Notebooks are also having the same capacity compact and lightweight and keyboard and screen are attached together. Personal Digital Assistance (PDA) is similar to computers used in the users hand.

Mini Computer :- A minicomputer is intermediate between a microcomputer and a mainframe in size, speed, and capacity, that can support time-sharing, and that is often dedicated to a single application. Minicomputers have been attached with multiple terminals and other devices that is capable of supporting hundreds of users simultaneously. It has large memory, Faster Input Output devices and support greater variety of software. Can connect upto 100 terminals of LAN and support wide range of applications. Word length – 32 bits. Ex - IBM, Burroughs.

Super-Mini/ Main Frame :- A mainframe computer is a very large computer capable of handling and processing very large amounts of data quickly i.e. intended for high volume of data processing. Larger capacity of memory, high speed operation than Mini Computers. It is a multi-user computer system which is capable of supporting hundreds of users simultaneously/ concurrently. Word length 64 bits. Used for Space application, University, Bankinmg, Airlines, Big Companies, WAN implementation.

Supercomputer :- A supercomputer is a computer that performs at highest operational rate. Performance of a supercomputer is measured in floating-point operations per second (FLOPS) instead of million instructions per second (MIPS) speed. A MFLOPS is equal to one million floating-point operations per second, and a GFLOPS is equal to one billion floating-point operations per

second. A TFLOPS is equal to one trillion floating-point operations per second. Very large size, most expensive, very high speed processing capacity. This is possible because of parallel processing technique which implements multiple processors to work in parallel manner. Generally desired to complex scientific jobs, weather forecasting, scientific simulations, nuclear energy research and analysis of geological data and is very expensive and has their own OS and programming language. Ex.- Cray, Param, Anupam

Micro Computers : divided as PC, PC-XT(Extended Technology), PC-AT (advance Technology)

PC (Personal Computer) : It is a general purpose microcomputer which can perform wide variety of tasks useful for individual user. It is small, relatively inexpensive. PCs are based on the microprocessor technology that enables manufacturers to put an entire CPU on one chip. Businesses use personal computers for word processing, accounting, desktop publishing, and for running spreadsheet and database management applications. At home, the most popular use for personal computers is playing games and surfing Internet. Although personal computers are designed as single-user systems, these systems are normally linked together to form a network. In terms of power, now-a-days High-end models PC offer the same computing power and graphics capability as Workstations/ Minicomputers.

Processor -Inter-8086/8088 Microprocessor
Memory – 640 KB RAM, Two 360 KB FDD, Numerical Processor -8087
System Bus- 8 bit data bus & 16 bit address bus
Clock Speed – 8 MHz
PC-AT
Processor- Intel 80386/80486/Pentium
Memory- 512 KB – 2 MB, FDD – 1.4 MB, HDD-1.2MB to 80 GB
System Bus – 32-64 bit
Clock speed – up 3 GHz
OS-MSDOS, Windows, Unix, Linux

Workstation : A workstation is a special computer designed for technical or scientific applications. Intended primarily to be used by one person at a time, they are commonly connected to a local area network and run multi-user operating systems. A workstation has more power. It can handle CAD, animation, data analysis, and photorealistic renderings, as well as video and audio creation and editing. The internal workings of a workstation are held to a higher standard than those of a PC.

Desktop : A *desktop computer* is designed to be used at a desk and seldom moved.

It consists of a large box called the *system unit* that contains most of the essential components. The monitor, keyboard, and mouse all plug into it using cables (or in some cases, using a wireless technology). Desktops offer a lot of computing power and performance for the price, and they're flexible because you can connect whatever monitor, keyboard, and mouse you want to it, as well as install additional storage drives, memory, and expansion cards that add new capabilities.

Notebook : A *notebook*, as its name implies, is a portable computer designed to fold up like a notebook for carrying and storage.

Its cover opens up to reveal a built-in screen, keyboard, and pointing device, which substitutes for a mouse. A notebook can run most of the same software as a desktop, and is similar to it in performance. Some people call it a laptop.

Notebooks allow you to take your computer with you almost anywhere. They're more expensive than desktops of the same level of performance, however, and not very customizable or upgradable.

Tablets : A *tablet* is a portable computer that consists of a touch-sensitive screen mounted on a tablet-size plastic frame with a small computer inside.

There is no keyboard or pointing device; a software-based keyboard pops up onscreen when needed, and your finger sliding on the screen serves as a pointing device. Tablets are extremely portable and convenient, but usually do not run desktop computer applications and have limited memory and storage capabilities.

Smart Phones : Smart phones have a lot in common with computers, but they lack the power and flexibility of larger computing devices.

Functions of each unit of Computer :

Input Unit / Input device :

The device through which computer receives information. Input devices are necessary to convert our data into a form which can be understood by the computer i.e this unit contains devices with the help of which we enter data into computer. This unit makes link between user and computer.

Following are few of the important input devices which are used in a computer:-

Keyboard	Mouse(Pointing Device)	Joy Stick	Light pen
Track Ball (Pointing Device)	Scanner	Graphic Tablet	Microphone
Magnetic Ink Card Reader (MICR)	Optical Character Reader(OCR)	Bar Code Reader	Optical Mark Reader (OMR)

Keyboard :-

Keyboard is the most common and user friendly input device. Data and programs are given to the computer using input devices. The Key board consists of a set of key switches. On pressing the key, the key switch is activated, the keyboard electronics determines the key pressed and sends an 8 bit scan code to computer. Determining which key is pressed & generating the relevant scan code for it is called Encoding. The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing additional functions. Keyboards are of two sizes 84 keys or 101/104 keys, but now keyboards with 109/110 are also available for Windows and Internet. Various Keyboards are PS2, Multimedia Keyboard, Wireless Keyboard, Internet Keyboard, Game Keyboards, Qwerty Keyboard used in Laptops. USB keyboard is a interface between PC and the device itself.

The keys on the keyboard are as basically divided into 3 parts :- Function keys, Numeric Keys and Alphanumeric keys.

SL	Keys	Description
1	Typing Keys	These keys include the letter keys (A-Z) and digit keys (0-9) which generally give same layout as that of typewriters.

2	Numeric Keypad	It is used to enter numeric data or cursor movement. Generally, it consists of a set of 17 keys that are laid out in the same configuration used by most adding machines and calculators.
3	Function Keys	The twelve function keys are present on the keyboard which are arranged in a row at the top of the keyboard. Each function key has unique meaning and is used for some specific purpose.
4	Control keys	These keys provide cursor and screen control. It includes four directional arrow keys. Control keys also include Home, End, Insert, Delete, Page Up, Page Down, Control(Ctrl), Alternate(Alt), Escape(Esc).
5	Special Purpose Keys	Keyboard also contains some special purpose keys such as Enter, Shift, Caps Lock, Num Lock, Space bar, Tab, and Print Screen.

Mouse :-

A mouse is a pointing device that detects two-dimensional X-Y co-ordinates and send corresponding signals to CPU. This motion is typically translated into the motion of a pointer on a display, which allows for fine control of a graphical user interface (GUI) applications like windows. Mouse can be used to control the position of cursor on screen, but it cannot be used to enter text into the computer. Its is Easy to use and not very expensive and moves the cursor faster than the arrow keys of keyboard. Mouse can be to draw pictures or point a certain instruction by select click/double click.

Types : Mechanical mouse(Hard rubber bulb may be 2/3 buttons with rotating switch), Optical mouse (Uses LED sensor technology, no need of pad, Cordless mouse/Wireless mouse (uses radio frequency to transmit signals, PS2 Mouse/USB Mouse has highest power usage. Mouse is designed to fit comfortably under the palm of our hand so that we can control movement of pointer on the screen and make selection from the screen by pressing the button of the mouse.

Joystick :- is a cursor control device used in Computer Games. A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. It is a also a pointing device with track ball which is used to move cursor position on a monitor screen. It is a stick having a spherical ball at its both lower and upper ends. The lower spherical ball moves in a socket. The joystick can be moved in all four directions.

The function of joystick is similar to that of a mouse. It is mainly used in Computer Aided Designing(CAD) and playing computer games. First used in aeroplane (to control movement) and also used in Cranes and JCB etc.

Light Pen :- Light pen is a pointing device which is allows user to display or draw on the screen in similar way to a touch screen with greater accuracy. It is used to select a displayed menu item or draw pictures on the monitor screen. It consists of a photocell and an optical system placed in a small tube. When the tip of a light pen is moved over the monitor screen and pen button is pressed, its photocell sensing element detects the screen location and sends the corresponding signal to the CPU. It is a pen shape device used to select objects on a display screen used in CAD applications similar to touch screen.

Track Ball :-

Track ball is an input device/pointing device similar to mouse that is mostly used in notebook or laptop computer. The trackball is usually located in front of the keyboard toward the user. Essentially, the trackball is an upside-down mouse that rotates in place within a socket. Since the whole device is not moved, a track ball requires less space than a mouse. A bulb is on the top of the mouse and we rotate the bulb to move the cursor. The bulb is mounted on the socket.

Touch Screen :- In this device user only touches the monitor screen to input data into the computer. ATM machines are under this category.

Digitizer :- Converts analog information to digital form. It converts graphics and pictorial data into binary input used for fine works and drawing and image manipulation applications. A digitizer tablet (also known as a digitizer or graphics tablet) is a tool used to convert hand-drawn images into a format suitable for computer processing. Images are usually drawn onto a flat surface with a stylus and then appear on a computer monitor or screen.

.

Microphone :- Converts soundwaves into electrical signals or digital form.

Microphone is an input device to input sound that is then stored in digital form. The microphone is used for various applications like adding sound to a multimedia presentation or for mixing music.

Scanner :- It is a kind of input device which converts printed text, graphics, pictures into a digital form. Two types of scanner;- i) Optical Scanner (OCR, OMR, BCR) ii) MICR .

Optical devices are Barcode Reader, Scanner, OCR, OMR, MICR

Magnetic Ink Card Reader(MICR) :-

Special ink (ink with iron oxide particles), which is sensitive to magnetic fields, is used in the printing of certain characters on the original documents or cheques are processed by MICR for identification of Bank A/C Number, code etc.

MICR (magnetic ink character recognition) is a technology used to verify the legitimacy or originality of paper documents, especially cheques used in Banks. The bank's code number and cheque number are printed on the cheques with a special type of ink that contains particles of magnetic material that are machine readable. The main advantages of MICR is that it is fast and less error prone.

Optical Character Reader(OCR):- OCR (optical character recognition) is the peripheral device enabling letters, numbers, pictures or text and convert into digital form.

Bar Code Readers :- A barcode reader (or barcode scanner) is an electronic device that can read and output printed barcodes (data in form of light and dark lines and spacing between tgem) to computer. Bar coded data is generally used in labelling goods, numbering the books etc. It may be a hand held scanner or may be embedded in a stationary scanner. Bar Code Reader scans a bar code image, converts it into an alphanumeric value which is then fed to the computer.

Optical Mark Reader(OMR) :-

OMR is a special type of optical scanner used to recognize the type of mark made by pen or pencil. It is used where one out of a few alternatives is to be selected and marked. It is specially used for checking the answer sheets of examinations having multiple choice questions. OMR sheet feed into OMR scanner where light less reflect or no reflect and check the answer which has already feed into the machine.

Web Camera : A webcam is a video camera that captures images as data for computer. It is connected to computer through USB port.

Both Input and Output Devices :

Floppy Disk Drive, Hard Disk Drive, MODEM, Touch Screen, Audio Cards/Sound Cards, Network Cards, FAX (Scan and Print)

Floppy Disk Drive : Floppy disk drive is an input device which drives Floppy Disks.

OUTPUT UNIT / OUTPUT DEVICES :-

Output unit consists of devices with the help of which we get the information from computer. This unit is a link between computer and users. Output devices translate the computer's output into the form understandable by users.

Following are few of the important output devices which are used in a computer :-

- Monitors
- Graphic Plotter
- Printer

Monitors :-

Monitors, commonly called as Visual Display Unit (VDU), are the main output device of a computer. It forms images from tiny dots, called pixels that are arranged in a rectangular form. The sharpness of the image depends upon the number of pixels.

There are two kinds of viewing screen used for monitors. 1) CRT 2) Flat Monitor

Cathode-Ray Tube (CRT) - The CRT display is made up of small picture elements called pixels. The smaller the pixels, the better the image clarity, or resolution. Different types of CRT display adapter used in PC. Size and aspect ratio – proportional ratio of height X width of image.

MDA (Monochrome Display Adapter) 720 X 350 pixels, support text display only.
CGA (Color Graphics Adapter) 25 lines by 80 Col, 640 X 200 pixels, support graphics
EGA (Enhanced Graphics Adapter), VGA(Video Graphics Adapter,
SVGA(Super VGA) – 1280 X 1024 pixels.

Most screens are capable of displaying 80 characters of data horizontally and 25 lines vertically. There are some disadvantages of CRT:-Large in Size, High power consumption

Flat-Screen Monitor :-

CRT monitors are gradually replaced with Flat screen monitors such as LCD (Liquid Crystal Display) and TFT (Thin Film Transistor) display because it uses less power and less space. The flat-panel display refers to a class of video devices that have reduced volume, weight and power requirement in comparison to the CRT. You can hang them on walls or wear them on your wrists. Current uses of flat-panel displays include calculators, video games, monitors, laptop computer, graphics display. The flat-panel display is divided into two categories:-

Emissive Displays - The emissive displays are devices that convert electrical energy into light. Example are plasma panel and LED(Light-Emitting Diodes).

Non-Emissive Displays - The Non-emissive displays use optical effects to convert sunlight or light from some other source into graphics patterns. Example is LCD(Liquid-Crystal Device)

Softcopy : A softcopy is an output which is not produced on paper or some materials which can not be touched or carried to another place in physical form. These are temporary in nature and displayed in Monitor in the form of text or picture or video etc.

Hardcopy : It is the output in the physical form on a paper or any other material which can be touched and permanent in nature.

PRINTER :-

Printer is an output device, which is used to print information on paper.
There are two types of printers:

- Impact Printers
- Non-Impact Printers

Impact Printer :-

The impact printers print the characters by striking them on the ribbon which is then pressed on the paper.
Characteristics of Impact Printers are the following:-

- Very low consumable costs
- Very noisy
- Useful for bulk printing due to low cost
- There is physical contact with the paper to produce an image

These printers are of two types

- Character printers (Print a single character at a time)
- Line printers (Print a line at a time)

Character Printers :-

Character printers are the printers which print one character at a time.
These are further divided into two types:

- Dot Matrix Printer(DMP)
- Daisy Wheel

DOT MATRIX PRINTER :- 80 column (9 pin) or 132 column(24 pin)

In the market one of the most popular printers is Dot Matrix Printer. These printers are popular because of their ease of printing and economical price. Each character printed is in form of pattern of dots and head consists of a Matrix of Pins of size (5*7, 7*9, 9*7 or 9*9) which come out to form a character that is why it is called Dot Matrix Printer. Usually speed varies as 180 cps, 240 cps, 320 cps.

Advantages : - Inexpensive, Widely Used, Other language characters can be printed
Disadvantages :- Slow Speed, Poor Quality

DAISY WHEEL

Head is lying on a wheel and pins corresponding to characters are like petals of Daisy (flower name) that is why it is called Daisy Wheel Printer. These printers are generally used for word-processing in offices which require a few letters to be sent here and there with very nice quality.
Advantages :- More reliable than DMP, Better quality, The fonts of character can be easily changed
Disadvantages :- Slower than DMP, Noisy, More expensive than DMP

Line Printers

Line printers are the printers which print one line at a time.

These are of further two types :- 1) Drum Printer 2) Chain Printer

DRUM PRINTER :-

This printer is like a drum in shape so it is called drum printer. The surface of drum is divided into number of tracks. Total tracks are equal to size of paper i.e. for a paper width of 132 characters, drum will have 132 tracks. A character set is embossed on track. The different character sets available in the market are 48 character set, 64 and 96 characters set. One rotation of drum prints one line. Drum printers are fast in speed and can print 300 to 2000 lines per minute.

Advantages :- Very high speed

Disadvantages :- Very expensive, Characters fonts cannot be changed

CHAIN PRINTER :-

In this printer, chain of character sets are used so it is called Chain Printer. A standard character set may have 48, 64, or 96 characters.

Advantages :- Character fonts can easily be changed, Different languages can be used .

Disadvantages :- Noisy

Non-impact Printers / Page Printer :- Prints a page at a time.

Non-impact printers print the characters without using ribbon. These printers print a complete page at a time so they are also called as Page Printers.

These printers are of two types :- 1) Laser Printers 2) Inkjet Printers

Characteristics of Non-impact Printers :-

Faster than impact printers. They are not noisy. High quality. Support many fonts and different character size.

Laser Printers :-

These are non-impact page printers. They use laser lights to produce the dots needed to form the characters to be printed on a page.

ADVANTAGES :- Very high speed, Very high quality output, Give good graphics quality, Support many fonts and different character size

DISADVANTAGES :- Expensive. Cannot be used to produce multiple copies of a document in a single printing.

Inkjet Printers :-

Inkjet printers are non-impact character printers based on a relatively new technology. They print characters by spraying small drops of ink onto paper. Inkjet printers produce high quality output with presentable features.

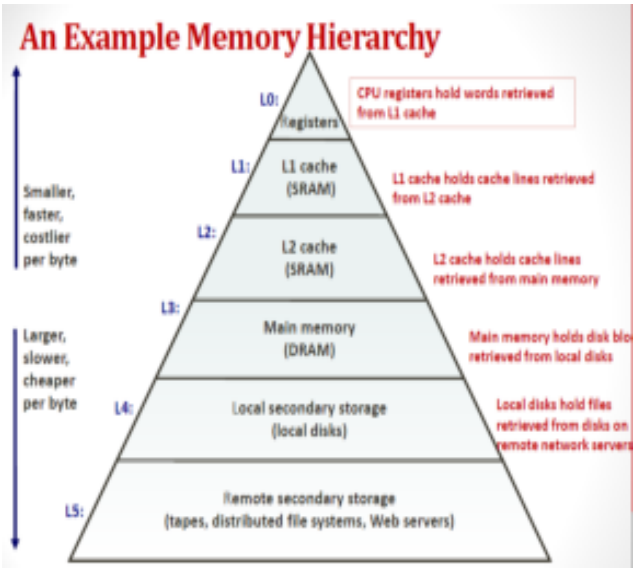
They make less noise because no hammering is done and these have many styles of printing modes available. Colour printing is also possible. Some models of Inkjet printers can produce multiple copies of printing also.

ADVANTAGES :- High quality printing, More reliable
DISADVANTAGES :- Expensive as cost per page is high, Slow as compared to laser printer

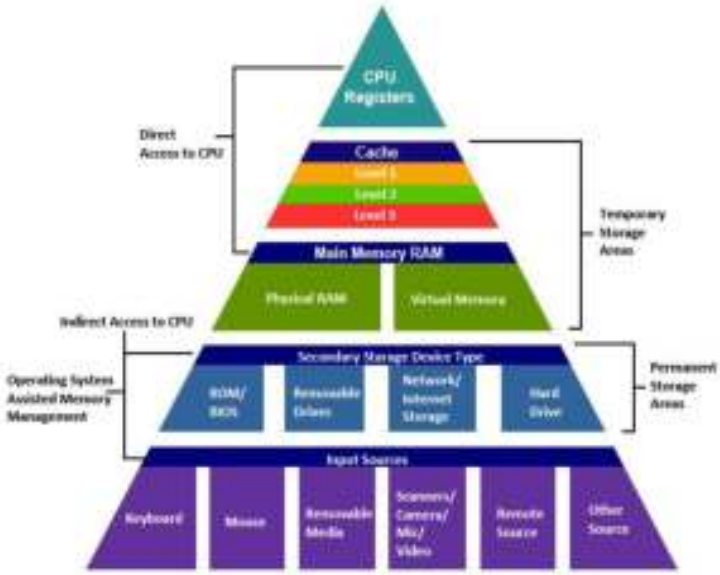
Plotters :-
It is a special purpose output device that draws pictures, graphs, charts, maps on paper based on commands from a computer. Plotters differ from printers in that they draw lines using a pen. As a result, they can produce continuous lines, whereas printers can only simulate lines by printing a closely spaced series of dots. Multicolor plotters use different-colored pens to draw different colors. There are two types of plotter i.e. Drum Plotter and Flat Bed Plotter.

LCD projector : -
An LCD projector is a type of video projector for displaying video, images or computer data on a screen or other flat surface. It is a modern equivalent of the slide projector or overhead projector.

Speaker : It is also known as voice recognition system. It produces sound. They receive audio input from the computer's sound card and produce audio output in the form of sound waves.



A



Memory : -
The function of memory is to store data and programs or results. A memory unit is collection of storage cells together with associated circuits needs to transfer information in & out of storage. The memory stores binary information in group of bits called words. Group of bits or words move in & out of storage. A memory word is a group of 1's and 0's and may represent number, an instruction code, one or more alphanumeric characters or any other binary coded information. A group of 8 bits is called a byte. Most computer memories use word whose number of bits is a multiple of 8. Thus a 16 bit word contains 2 byte and 32 bit word contains 4 byte capacity of memory. Thus a word is the minimum addressable unit of computer. The memory is divided into large number of small parts called cells. Each location or cell has a unique address which varies from zero to memory size minus one. For example if computer has 64k words, then this memory unit has 64 * 1024=65536 memory locations. The address of these locations varies from 0 to 65535. A device which can store a single bit (either 0 or 1) is called memory cell/flip flop. A group memory cell used to represent instruction or data to form a register (to store a word).

Register Memory : It is integrated inside the CPU. It consists of a number of Flip-Flops arranged in certain manner. It is small capacity. Used for storing data & instructions temporarily during execution of an instruction. Special purpose registers are MAR, MDR, IR, PC. It is fastest memory.

Cache Memory : It is a small memory situated between CPU and Main memory. The purpose of this memory is to hold/store frequently used instructions or data from the main memory during the execution process. This is a semiconductor memory which is having very low access time and hence is a fast memory.

Cache memory is a very high speed semiconductor memory which can speed up CPU. It acts as a buffer between the CPU and main memory. It is used to hold those parts of data and program which are most frequently used by CPU. The parts of data and programs are transferred from disk to cache memory by operating system, from where CPU can access them.

Advantages

The advantages of cache memory are as follows:

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed within a short period of time.
- It stores data for temporary use.

Disadvantages

The disadvantages of cache memory are as follows:

- Cache memory has limited capacity. • It is very expensive.

Memory is primarily of two types :-

- Primary Memory/Main Memory
- Secondary Memory

Primary Memory (Main Memory) :-

Primary memory holds only those data and instructions on which computer is currently working. It has limited capacity and data is lost when power is switched off because of semiconductor property. It is generally made up of semiconductor device. These memories are not as fast as registers. The data and instruction required to be processed reside in main memory.

It is divided into two categories RAM and ROM.

Characteristics of Main Memory :-

- These are semiconductor memories
- It is known as main memory.
- Usually volatile memory.
- Data is lost in case power is switched off.
- It is working memory of the computer.
- Faster than secondary memories.
- A computer cannot run without primary memory.

RAM(Random Access Memory) is the internal memory of the CPU for storing data, program and program result. It is read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased.

Access time in RAM is independent of the address that is, each storage location inside the memory is as easy to reach as other locations and takes the same amount of time. Data in the RAM can be accessed randomly but it is very expensive.

RAM is volatile, i.e. data stored in it is lost when we switch off the computer or if there is a power failure. Hence a backup uninterruptible power system(UPS) is often used with computers. RAM is small, both in terms of its physical size and in the amount of data it can hold.

RAM is of two types

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

Static RAM (SRAM)

The word static indicates that the memory retains its contents as long as power is being supplied. However, data is lost when the power gets down due to volatile nature. SRAM chips use a matrix of 6-transistors and no capacitors. Transistors do not require power to prevent leakage, so SRAM need not have to be refreshed on a regular basis.

Because of the extra space in the matrix, SRAM uses more chips than DRAM for the same amount of storage space, thus making the manufacturing costs higher. So SRAM is used as cache memory and has very fast access.

Characteristic of the Static RAM

- It has long life
- There is no need to refresh
- Faster
- Used as cache memory
- Large size
- Expensive
- High power consumption

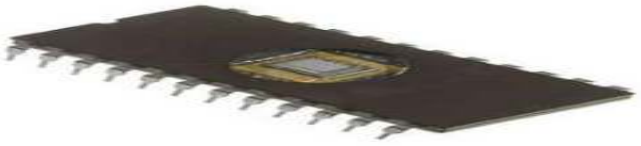
Dynamic RAM (DRAM)

DRAM, unlike SRAM, must be continually refreshed in order to maintain the data. Dynamic RAM losses its contents in a very short time even though power supply is on & has to be periodically re-written or refreshed. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second. DRAM is used for most system memory because it is cheap and small. All DRAMs are made up of memory cells which are composed of one capacitor and one transistor.

Characteristics of the Dynamic RAM

- It has short data lifetime
- Need to be refreshed continuously
- Slower as compared to SRAM
- Used as RAM
- Lesser in size
- Less expensive
- Less power consumption

ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture. A ROM, stores such instructions that are required to start a computer. This operation is referred to as bootstrap. ROM chips are not only used in the computer but also in other electronic items like washing machine and microwave oven.



Following are the various types of ROM

PROM (Programmable Read only Memory)

PROM is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM program. Inside the PROM chip there are small fuses which are burnt open during programming. It can be programmed only once and is not erasable.

EPROM(Erasable and Programmable Read Only Memory)

The EPROM can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes. Usually, an EPROM eraser achieves this function. During programming, an electrical charge is trapped in an insulated gate region. The charge is retained for more than ten years because the charge has no leakage path. For erasing this charge, ultra-violet light is passed through a quartz crystal window(lid). This exposure to ultra-violet light dissipates the charge. During normal use the quartz lid is sealed with a sticker.

EEPROM(Electrically Erasable and Programmable Read Only Memory)

The EEPROM is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (milli second). In EEPROM, any location can be selectively erased and programmed. EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of re-programming is flexible but slow.

Advantages of ROM

The advantages of ROM are as follows:

- Non-volatile in nature
- These cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- More reliable than RAMs
- These are static and do not require refreshing
- Its contents are always known and can be verified

Secondary Memory :-

Secondary Memory is known as external memory or non-volatile or auxiliary memory. Since the RAM is not permanent, a computer system requires secondary memory to hold data and programs permanently. The CPU can not directly access data and programs of secondary memory. They first need to transfer into RAM before being used. Most widely used secondary storage devices are magnetic tapes, magnetic disk (Hard disk, floppy disk), optical disk (CD, DVD) and flash drive. These devices store data, information and programs permanently in the form of binary digits till we don't erase them. It is slower than main memory.

Characteristic of Secondary Memory

- These are magnetic and optical memories
- It is known as backup memory.
- It is non-volatile memory.
- Data is permanently stored even if power is switched off.
- It is used for storage of data in a computer.
- Computer may run without secondary memory.
- Slower than primary memories.

Secondary Memory :

a) **Hard Disk** :-Hard Disk is present inside the system unit. Hard disk is the main secondary storage device that stores large amount of data permanently. Hard disk is also called fixed disk. Hard disk stores operating system software of a computer, data and other applications software. Hard disk consists of many metal disks or platters coated with magnetic material that store data in tracks, which is further divided into sectors. The various capacities of Hard disks are available in the market. Nowadays hard disk of 1TB (Tera Byte) is available in the market.

b) **Floppy Disk** :- A floppy disk is a portable storage device that stores data permanently. The floppy disk drive is used to read and write data on a floppy disk. Floppy disk may be mini floppy disk or micro floppy disk. The storage capacity of a Micro Floppy disk is 1.44 MB (Mega Byte) and of a Mini floppy disk is 1.2 MB. Floppy disk gets damaged easily. So, nowadays it is not used. Floppy disk is a thin plastic disk coated with magnetic material for storage of data or information. Floppy disk has a hard outer covering. Floppy disk is inserted into the disk drive while saving information into it or copying files into it. Thus, the file can be transferred into other computer.

c) **Compact Disk/DVD** :- Compact Disk/Digital Video Disk stores huge amount of data permanently. The Compact Disk Drive (CD-ROM) is required to read data from a compact disk. A CD-RW drive is

used to write data on blank CDs and delete, modify or add data on an already written CD-RW (CD-Rewritable). DVD Drive (DVD-ROM) is required to read the data from DVD (Digital Versatile Disk). DVD Writer allows you to read as well as write data on DVDs. The capacity of a compact disk is 680 MB to 700MB and DVD is 7 GB. CD/DVD is portable disk that can be used for transferring data and programs from one computer to another computer. Since CDs and DVDs are reliable storage devices they are also used for keeping backup copy of data and programs.

d) **Pen Drive** :- Pen drive is also called Flash Drive. Pen drive is a portable device used for transferring data and programs from a computer to another computer. You can use a pen drive to copy data, songs, games and programs from one computer to another computer. Different storage capacities of pen drives (256 MB to 16 GB) are available in the market. USB Flash Drive 4GB

Units of Memory :

- the amount of data that can be stored in the storage unit.
- that in which storage capacity is expressed in terms of Bytes.

Following are the main memory storage units:-

Sr.No.	Unit	Description
1	Bit (Binary Digit)	A binary digit is logical 0 and 1 representing a passive or an active state of a component in an electric circuit.
2	Nibble	A group of 4 bits is called nibble.
3	Byte	A group of 8 bits is called byte. A byte is the smallest unit which can represent a data item or a character.
4	Word	A computer word, like a byte, is a group of fixed number of bits processed as a unit which varies from computer to computer but is fixed for each computer. The length of a computer word is called word-size or word length and it may be as small as 8 bits or may be as long as 96 bits. A computer stores the information in the form of computer words.

Few higher storage units are following

Sr.No.	Unit	Description
1	Kilobyte (KB)	1 KB = 1024 Bytes
2	Megabyte (MB)	1 MB = 1024 KB
3	GigaByte (GB)	1 GB = 1024 MB
4	TeraByte (TB	1 TB = 1024 GB
5	PetaByte (PB)	1 PB = 1024 TB

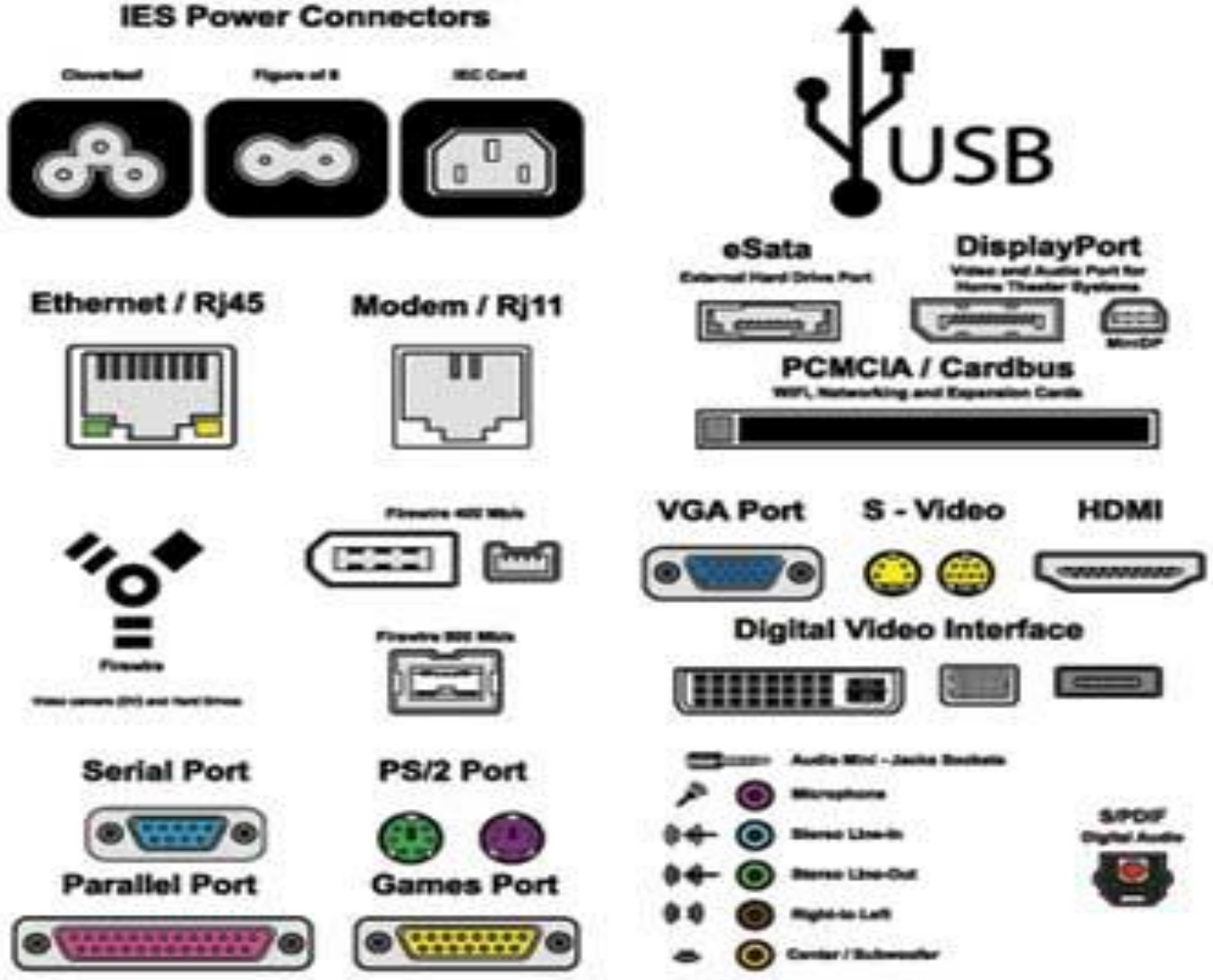
What is a Port?

A port is a physical docking point using which an external device can be connected to the computer. It can also be programmatic docking point through which information flows from a program to computer or over the internet.

Characteristics

A port has the following characteristics:

- External devices are connected to a computer using cables and ports.
- Ports are slots on the motherboard into which a cable of external device is plugged in.
- Examples of external devices attached via ports are mouse, keyboard, monitor, microphone, speakers etc.



Following are few important types of ports:

Serial Port

- Used for external modems and older computer mouse
- Two versions : 9 pin, 25 pin model
- Data travels at 115 kilobits per second

Parallel Port

- Used for scanners and printers
- Also called printer port
- 25 pin model
- Also known as IEEE 1284-compliant Centronics port

PS/2 Port

- Used for old computer keyboard and mouse
- Also called mouse port

- Most of the old computers provide two PS/2 port, each for mouse and keyboard
- Also known as IEEE 1284-compliant Centronics port

Universal Serial Bus (or USB) Port

- It can connect all kinds of external USB devices such as external hard disk, printer, scanner, mouse, keyboard etc.
- It was introduced in 1997.
- Most of the computers provide two USB ports as minimum.
- Data travels at 12 megabits per seconds
- USB compliant devices can get power from a USB port

VGA Port

- Connects monitor to a computer's video card.
- Has 15 holes.
- Similar to serial port connector but serial port connector has pins, it has holes.

Power Connector

- Three-pronged plug
- Connects to the computer's power cable that plugs into a power bar or wall socket

Firewire Port

- Transfers large amount of data at very fast speed.
- Connects camcorders and video equipment's to the computer
- Data travels at 400 to 800 megabits per seconds
- Invented by Apple
- Three variants : 4-Pin FireWire 400 connector, 6-Pin FireWire 400 connector and 9-Pin FireWire 800 connector

Modem Port

- Connects a PC's modem to the telephone network

Ethernet Port

- Connects to a network and high speed Internet.
- Connect network cable to a computer.
- This port resides on an Ethernet Card.
- Data travels at 10 megabits to 1000 megabits per seconds depending upon the network bandwidth.

Game Port

- Connect a joystick to a PC
- Now replaced by USB.

Digital Video Interface, DVI port

- Connects Flat panel LCD monitor to the computer's high end video graphic cards.
- Very popular among video card manufacturers.

Sockets

- Connect microphone, speakers to sound card of the computer

CHAPTER – II (COMPUTER SOFTWARE) :

Study of computer consists of study of hardware (physical) and study of software (logical). The software is nothing but the instructions or programs by which the hardware runs i.e without software hardware has no meaning. Software instructs or control the hardware i.e. what to do and how to do it.

Computer Software is classified into 2 types : Systems software and Application software

System software :- System Software is a set of programs designed to control or manage the operations of computer hardware and its resources i.e. enables the user to interact with the hardware components. It acts as an interface between user and the hardware. System Software is machine dependent.
System Software is a collection of system programs that perform a variety of functions such as :- File Management, Resource Management, I/O Device Management, Storage/Data Management, Memory Management etc
Some common System Software are :- Operating System, Language Processor/Compilers, Device Drivers, Utility Programs .
Operating System is a System control programs that controls the execution of programs , manage the storage & processing resources of the computer , perform other management & monitoring function.
Utilities or Libraries is a System support programs that Provides routine service functions to the other computer programs & computer users.
Compilers and Interpreters is a System development programs that assists in the creation of application Programs.
Device drivers are system software which generally comes along with a peripheral device and makes easy communication between computer and devices.
Utility Programs are capable of interacting with the computer hardware for various purpose. Generally for system maintenance activity. Ex:- antivirus, DM, Data compression utility

Application software :- It is a set of Programs that is designed to solve a specific task or job .
Ex :- Inventory, Payroll, Hospital management, Banking application, Ticket Booking, Library etc. The Application Software is machine independent.

Operating System :- An operating system (OS) is software that manages computer hardware resources and provides common services for computer programs. Application programs usually require an operating system to function. It acts as a interface between Computer and User.

USER > Application Package > Compilers/Interpreters > Operating System > Hardware

Functions of the Operating System :
CPU Management/Processor Management/Scheduling, Peripheral Management, Memory Management (automatic allocation), Data Management, Resource/Device Management, File Management/ Information Management.

Resource Management : Keeping track of all available resources, Allocation of resources to the various requesting jobs, Deallocation of resources.

Process Management : Keeping track of all active process and allocation of processors to various active process, Creation of child processes and termination of processes, assigning and changing priority of the process, block and allocating processes , delaying a process.

Memory Management : Keeping track of the available memory, Allocating the memory to different process and de-allocation old memory from process.

Device Management : Allocation of devices to process and deallocation of devices.

TYPES OF OPERATING SYSTEM : -

OS is broadly classified into two groups :- (a) Single User OS (b) Multi User OS

Single User OS :- This allows one user to work on the compute at a time. This OS is designed to run either one job at a time or support multiple tasks at the same time. Example:- MSDOS, Windows98, XP.

Multiple User OS :- This allows multiple users to work together on a single computer. This OS allows users to take advantage of the computer resources simultaneously. Example :- UNIX, LINUX.

Some typical Operating Systems are as follows :-

(a) Batch Operating System :-
Jobs are submitted on cards or tape to an operator who batches jobs together sequentially When the entire computer is dedicated to a single program/job at any given time, the system is said to be operating in batch processing mode. Batch processing system support processing of jobs one at a time. In this mode, each program is fully completed before the system switches over to next. Jobs are submitted in batch. It is the responsibility of the OS to schedule the job in a queue and assign system resources one after another. CPU remains idle during the time of program call for input and output operation. It has been found that for typical commercial application, CPU efficiency may be as low as 35%.

(b) Multiprogramming System :-
The draw back of batch processing is that it does not provide optimum CPU utilization and multiprogramming utilises maximum CPU time by running multiple programs/jobs simultaneously. In this mode OS switches from one job to another as soon as an interrupt occurs due to input/output operation. Multi programming helps maximizing CPU efficiency by reducing CPU idle time and It increases resource utilization. The OS picks one of the program and start executing it and during the execution it may need input or output operation to complete. Maximize Processor use.
Multiprogramming is a basic form of parallel processing in which several programs are run at the same time on a uniprocessor. Since there is only one processor, there can be no true simultaneous execution of different programs. Instead, the operating system executes part of one program, then part of another, and so on. To the user it appears that all programs are executing at the same time.

(c) Multiprocessing System :-
Generally a computer system has a single processor means a computer have just one CPU for processing the instruction. But if we are running multiple jobs , then we will decrease the CPU speed. For increasing the speed of the processing , we use multiprocessing. This system works with 2 or more CPU within a single Operating System, if one CPU fails then another CPU is used for providing backup to first CPU. Multiprocessing OS enables to execute several programs at a time.

(d) Time Sharing OS :-
Time Sharing allows a large number of users at various terminals to simultaneously use a centrally located computer. This OS makes time slice the CPU time and distribute among the multiple users sitting at various terminals. Each user is allocated resources for a particular slot. The switching between users is so fast that each user think the he or she is getting the system resources continuously. Here CPU execute the program at a time but each program is given a time say 2 mili second after which CPU switches over to next user in a cyclic fashion.

In computing, time-sharing is the sharing of a computing resource among many users by means of multiprogramming and multi-tasking. Minimize response time.

(e) Real Time Operating System (RTOS) :-

A real-time operating system is a multitasking operating system that aims at executing real-time applications. Real-time operating systems often use specialized scheduling algorithms so that they can achieve a deterministic nature or behavior. The main objective of real-time operating systems is their quick and predictable response to events. They have an event-driven or timesharing design and often aspects of both. An event-driven system switches between tasks based on their priorities/logic or external events while time-sharing operating systems switch tasks based on clock interrupts i.e. time in which it is performed.

Hard Real Time System :- Time is fixed and we can not change any moments of the time of processing. CPU will process data as we enter data.

Soft Real Time System :- Some moments can be change means after giving command to the CPU, CPU performs operation after a microsecond.

(f) Network Operating System ;-

It is the software that includes special functions for connecting computers and devices into a local area network. It controls a network and its message traffic and control access multiple users to network resources such as files and provides for certain administrative functions including security. Ex:- Windows NT, Netware, Unix, Mac OS, Xenix

(g) Distributed Operating System :- Distributed means data is stored and processed on multiple locations. Distributed means a it is a network collection of computers that are connected each other. Distribute the computation among several physically separated processors.

Loosely coupled system – each processor has its own local memory; processors communicate with one another through various communications lines, such as high-speed buses or telephone lines.

Advantages of distributed systems :- Resources Sharing , Computation speed up – load sharing Reliability and fault tolerance, Communications

Requires networking infrastructure - Local area networks (LAN) or Wide area networks (WAN)

Features of DOS, Windows and UNIX OS :-

The operating system is a system software used for management of computer hardware and application software. It is a interactive interface between user and the computer.

DOS : -

MS-DOS stands for Microsoft Disk Operating System, the first OS installed in PCs. The main function of DOS are to manage disk files , allocate system resources according to requirement. It is a single user single tasking Operating System. Tim Paterson developed this operating system in 1980. The IBM (International Business Machine) released first PC (Personal Computer) in 1981. MS-DOS version 1.0 was used as operating system in IBM-PC and become talk of town in overnight.

The father of PC Operating System is Gary Kildall of Digital Research designed successful operating System called CP/ M. The selling of CP / M is more than 600,000 copies proves its popularity. The Microsoft Disk Operating System or MS-DOS was based on QDOS, the Quick and Dirty Operating System written by Tim Peterson of Seattle Computer Products, for their prototype Intel 8086 based computer. QDOS was based on Gray Kildall's CP/M.

It has three essential files and many command files. These essential files are: IO.SYS, MSDOS.SYS, and COMMAND.COM files are loaded into internal memory after POST (Power on Self Test) is executed. POST is a routine program that is part of the ROMBIOS when we switch on the PC.

The heart of MS-DOS : responsible for DOS Prompt C:\>

- IO.SYS: This let DOS communicate with the hardware through the BIOS (Basic Input / Output System). MSDOS.SYS: This is a DOS kernel.
- COMMAND.COM: This is where all the DOS commands are stored and interpreted.

- COFIG.SYS: Hardware configuration information is stored here.
- AUTOEXEC.BAT: All the programs that are supposed to run at start-up are called here.

BOOTING :- The booting is a process of loading system files into main memory (RAM). There are two types of booting.

Commands Processor : Command.Com file along with above IS.SYS, MSDOS.SYS files causes the prompt C:\> to be displayed in the monitor or screen and view and interpret commands given are (a) Internal command like Date, Time, CLS, DIR, REN, DEL, ERASE TYPE, COPY MD CD PROMPT PATH VOL VER RD and (b) External commands are command files they reside externally in the disk and used as and when required are CHKDSK, DISKCOPY, DISKCOMP, TREE, ATTRIB, FORMAT, FDISK, SCANDISK, DELTEREE etc.

WINDOWS :-

It is an operating system, extension of MS-DOS with user friendly GUI based OS and several facilities to control memory, hardware, text, graphics, audio, video, internet connection etc.
Desktop display area that represents a kind of objects one might find documents. Icon is a small graphical symbol which represents a particular executable program. Folder is like directory of the DOS where we create various files under it. Taskbar normally represents at the buttom of the screen contains the active application programs. Title bar normally represents at the top of the window contains the title of the window. Menu bar presents just below the title bar of the window. A window contains a vertical and horizontal scroll bar for movement of the screen. AT the top right corner there are 3 buttons known as minimise(application is closed but still running, maximise(application in full screen) and close button.
Accessory contains frequently needed tools. Control panel contains various options for system maintenance/settings, adding or removing programs. My computer contains the entire secondary storage. Recycle Bin in contains all the deleted files.

Version	Comments
Windows 1.0	This operating system with user interface is a notification of MSDOS. The nifty mouse is used to click on desired program to open. It was first called interface manager, but then changed in to the more appealing Windows. Windows 1.0 lunched in November 1985.
Windows 2.0	It was released in 1987 to take advantage of the awesome processing power of the Intel 286 processor. The first version of Microsoft Word and Excel are introduced in this version.
Windows 3.0	It was released in May 1990. It came with a prettier 16-color interface, and new technological bells and whistles that let it make better use of the memory. In 1991, Microsoft brought multimedia support for Windows 3.0, called
	Multimedia Extensions 1.0. It gave Windows support for CD-ROM drives and sound cards. It also contained a basic CD player application for Windows
Windows 3.1	It was released in April 1992. It was equipped with big, comprehensive API (Application Program Interface), which simplified the task of creating user interface and let them focus more time on developing the core functionality of software. In 1993, windows for Workgroup 3.1 were released, which added support for networking, file and printer sharing. It also added Microsoft Mail Program to send and receive over the network.

Windows NT	In 1988, Microsoft had been developing, Windows NT; the NT stands for new technology. This was a whole new kernel, built for data and application security. It is a robust, pre-emptive, multithreaded, multi-tasking, 32-bits operating system with symmetric multiprocessing support.
Windows 98	It released in 1995. It is equipped with advance technology like AGP (Accelerated Graphics Port), MMX (multimedia Extension), USB (Universal serial Bus), DVD (Digital Video Disk) etc. Its most visible
Windows 2000	Its interface is similar to interface for windows 98. It has new security protocol with an encryption facility to authenticate users logging in to the network. It supports 32 FAT file system along with NTFC (New Technology File System), making it easier for users to upgrade for Windows 98. It has quite robust hybrid kernel architecture, make it more stable version.
Windows XP	Here, XP Stands For eXPerience. It bought together the robust Kernel of windows 2000 and all the friendless and multimedia support of windows Me, and painted on a new face for it. Apart from the merger of Windows 2000 and me, Windows Xp also added new features to enhance its performance. The first of These was its ability to work even in low- memory conditions without crashing, using a technique called Memory Throttling, Usually, Windows likes to do many things at once, but when memory falls short, it will throttle its memory access, doing fewer at a time. This shows the system down considerably, but prevents it form crashing.

UNIX Operating System :-

The UNIX is a powerful, flexible, multi-user, portable and machine independent Operating system with GUI and several utilities. Ken Thompson and Dennis Ritchie wrote C compiler under UNIX in 1969 at AT&T Bell Labs. In 1973, Thompson and Ritchie rewrote the UNIX Kernel using C language. It is based on MULTICS operating system. Its first user was Bell patent department. XENIX, VENIX, MICRONIX, LINUX, UNIXWARE-7 etc are version of UNIX operating system.

The UNIX operating system is made up from three parts:

(a) Kernel: It is a hub of operating system dedicated for memory management, file management and communication within system.

(b) Shell: It is an interface between kernel and users. When a user logs in, the login program matches the username and password, and then starts shell. The shell is a command line interpreter (CLI) of UNIX.

(c)Tools & Applications :Program or command is used to accomplish specific tasks. When one command is terminated, the shell displays prompt % to accept next command for execution. It provides additional functionality to OS

Some UNIX Commands:

To display files of current directory: Ls

Example: %Ls

To make directory: Mkdir<destination>

Example: %Mkdir unixst

To change directory: cd

Example: %cd

To copy file: cp<source><destination>

Example: 5cp /student /example /bio.txt

In UNIX, the dot means the current directory.

To move file: mv<file1><file2>

Example: 5mv bio.txt hello/

To remove or delete directory: rm or rmdir<directory>
Example: %rm unixstd
To display contents of file: Cat<file1>
Example: %cat bio.txt
To clean monitor: clear Example: %clear

Programming Languages :-

Computer can understand binary data, but it is difficult to write programs using 0's and 1's. Therefore, Programming Language invented. There are 3 types of Programming Language :- (a) Machine Language (b)Assembly Language (c) High Level Language.

(a) Machine Language :- Also known as low level language, Programs written in binary form.

Advantages : Data does not require an interpreter for execution. Execution is very fast.

Disadvantages: Machine dependent. The Programmer need to have the idea about the computer hardware. Programs are prone to error. Difficult to understand and modify. Not user friendly.

(b) Assembly language :- Programs are written in pneumonic codes. Numeric address location can be addressed by using alphanumeric variables. Each instruction consists of 2 parts i.e. opcode and operand. Opcode tell the computer what to do. Operand gives the address of operand on which the operation will be carried out. But program written in ALL has to go through a translator called Assembler which translate ALL to machine language format.

Advantages : easier to understand. Easier for locating errors and debugging.

Disadvantages : Machine dependent. Knowledge of hardware architecture is required.

(c) High Level Language :- A high level language is a programming language , where a programmer write program in English like language. It is user friendly. But a program written in HLL has to go through a translator called Interpreter/Compiler , where it is interpreted into corresponding binary format.
Advantages : Machine independent. Easier to learn and use. Gives minimum errors. Better documentation. Easy to modify and maintain.

Compiler : It is software which translate the high level language programs into machine level language. A compiler works with 3rd generation language such as Java, C & Other High Level Languages like COBOL, FORTRAN, PASCAL etc.

Characteristics : Spends a lot of time for analysing and processing the program. The resulting executable format is some forms of machine specific binary form. The computer hardware /interprets the resulting code. Program execution is very fast.

Interpreter : The language processor converts a high level language into machine language line by line as well as execute it. If there is error in the program, translation and execution stops and error is reported for debugging. The conversion and execution resumes only after the error is reflected. This is slow process as it consumes high memory as the interpreter is required to execute every time the program is required for run.

Comparison between DOS and Windows :-		
Windows		DOS
It is Graphics User Interface S/W		It is non-GUI based S/W i.e Character User Interface
Powerful and faster		Less powerful and slower
It provides inbuilt commands in various applications and accessories which can be used on a mouse click		All commands have to be types at the DOS Command prompt
Virtually all DOS commands are available in Windows		Not applicable
It provides multitasking environment		It is a Single User and Single tasking OS
It supports Graphics		It does not support graphics

Comparison UNIX & DOS:-		
UNIX		DOS
Powerful OS		Less powerful OS
Support multitasking & multiprogramming environment and hierarchical file system		Does not support multitasking and multiprogramming and hierarchical file system
Support visual display and graphics		Does not support graphics
Support networking of computers		Does not Support networking of computers
Secure		Not Safety
Support concept of Kernel and Shell		Does not Support concept of Kernel and Shell

Computer Virus :-

A computer virus is a malware program or malicious software/computer threat/spyware, browser hijacking is used to disrupt the computer operation. When virus program is executed it replicates by inserting copies of itself into other computer programs, data files, or the boot sector of the hard drive. When this replication succeeds, the affected areas are then said to be "infected". Viruses may corrupt or delete data on a computer. Virus are most easily spread by attachment in email messages or using other files which might have been virus affected. The harmful activities are stealing hard disk space or CPU time, accessing private information, corrupting data, displaying political or humorous messages on the user's screen, spamming their contacts, or logging their keystrokes. However, not all viruses carry a destructive payload or attempt to hide themselves—the defining characteristic of viruses is that they are self replicating computer programs which install themselves without the user's consent.

Computer viruses currently cause billions of dollars worth of economic damage each year, due to causing systems failure, wasting computer resources, corrupting data, increasing maintenance costs, etc. In response, free, opensource antivirus tools have been developed, and a multi-billion dollar industry of antivirus software vendors has cropped up, selling virus protection to users of various operating systems of which Android and Windows are among the most victimized.

Different Types of computer virus are :-

There are different types of viruses which can be classified according to their origin, techniques, types of files they infect, where they hide, the kind of damage they cause, the type of operating system, or platform they attack..

Boot Sector Virus :- It affects boot sector or Master Boot Record (MBR) of the HardDisk, FloppyDisk, External HardDisk etc. They replace the boot record program (which is responsible for loading OS in memory) by copying it elsewhere on the disk or overwriting it. Therefore the boot virus load into memory while Booting. Example :- Polyboot, AntiEXE, Michalangelo, Disk Killer, Stoned, Pingpong etc.

Memory Resident Virus :- These viruses fix themselves in the computer memory and get activated whenever the OS runs and infects all the files that are then opened. This type of virus hides in the RAM and stays there even after the malicious code is executed. It gets control over the system memory and allocate memory blocks through which it runs its own code, and executes the code when any function is executed. It can corrupt files and programs that are opened, closed, copied, renamed, etc. Examples: Randex, CMJ, Meve, and MrKlunky

Direct Action Viruses:- The main purpose of this virus is to replicate and take action when it is executed. When a specific condition is met, the virus will go into action and infect files in the

directory or folder that are specified in the AUTOEXEC.BAT file path. This batch file is always located in the root directory of the hard disk and carries out certain operations when the computer is booted. It also infects the external devices like pen drives or hard disks by copying itself on them. The viruses keep changing their location into new files whenever the code is executed, but are generally found in the hard disk's root directory. It can corrupt files. Basically, it is a file-infector virus. Examples: Vienna virus

Overwrite Viruses:- A virus of this kind is characterized by the fact that it deletes the information contained in the files that it infects, rendering them partially or totally useless once they have been infected. The virus replaces the file content. However, it does not change the file size.
Examples:Way,Trj.Reboot,Trivial.88.D

Macro Virus :- Macro viruses infect files that are created using certain applications or programs that contain macros, like .doc, .xls, .ppt, .mdb, etc. These mini programs make it possible to automate series of operations so that they are performed as a single action, thereby saving the user from having to carry them out one by one. These viruses automatically infect the file that contains macros, and also infects the templates and documents that the file contains. It is referred to as a type of e-mail virus. These hide in documents that are shared via e-mail or networks.
Examples: Relax,Melissa.A,Bablas,O97M/Y2K

Directory viruses (also called Cluster Virus/File System Virus) : It infect the directory of your computer by changing the path that indicates the location of a file. When you execute a program file with an extension .EXE or .COM that has been infected by a virus, you are unknowingly running the virus program, while the original file and program is previously moved by the virus. Once infected, it becomes impossible to locate the original files. It is usually located in only one location of the disk, but infects the entire program in the directory. Examples:Dir-2virus

Companion viruses can be considered as a type of file infector virus, like resident or direct action types. They are known as companion viruses because once they get into the system they 'accompany' the other files that already exist. In other words, to carry out their infection routines, companion viruses can wait in memory until a program is run (resident virus), or act immediately by making copies of themselves (direct action virus).Hideout: These generally use the same filename and create a different extension of it. For example: If there is a file "Me.exe", the virus creates another file named "Me.com" and hides in the new file. When the system calls the filename "Me", the ".com" file gets executed (as ".com" has higher priority than ".exe"), thus infecting the system.
Examples: Stator, Asimov.1539 and Terrax.1069

FAT Virus :- The file allocation table (FAT) is the part of a disk used to store all the information about the location of files, available space, unusable space, etc.. FAT virus attacks the FAT section and may damage crucial information. It can be especially dangerous as it prevents access to certain sections of the disk where important files are stored. Damage caused can result in loss of information from individual files or even entire directories.Ex:Link VirusProtection.

Worm Virus :-A worm is a program very similar to a virus; it has the ability to self-replicate and can lead to negative effects on your system. But they can be detected and eliminated by an antivirus software. Hideout: These generally spread through emails and networks. They do not infect files or damage them, but they replicate so fast that the entire network may collapse.
Examples: PSWBugbear.B, Lovgate.F, Trile.C, Sobig.D, Mapson Protection: Install an updated version of antivirus.

Trojans Virus : These virus is a malicious code i.e Trojans or Trojan horses, which unlike viruses, do not reproduce by infecting other files, nor do they self-replicate like worms. In fact,

it is a program which disguises itself as a useful program or application. Viruses copy files in your computer (when their carrier program is executed) that can damage your data, and even delete it. The attacker can also program the trojans in such a manner that the information in our computer is accessible to them.

Logic Bombs :- These are not considered viruses because they do not replicate. They are not even programs in their own right, but rather camouflaged segments of other programs. They are only executed when a certain predefined condition is met. Their objective is to destroy data on the computer once certain conditions have been met. Logic bombs go undetected until launched, the results can be destructive, and your entire data can be deleted.

Virus Symptoms :- Computer is giving problem during Booting, Restart automatically, Hanging when user wants to execute a program, Displaying some unusual figures/symbols, Performing some operation automatically, Giving some absurd messages, Harddisk is accessed many times.

Detection and Prevention of Virus :-

Following steps may be taken for Virus Detection and Prevention.

1. Do not open any files attached to an email from an unknown, suspicious or untrustworthy source.
2. Do not open any files attached to an email unless you know what it is, even if it appears to come from a dear friend or someone you know. Some viruses can replicate themselves and spread through email. Better be safe than sorry and confirm that they really sent it.
3. Delete chain emails and junk email. Do not forward or reply to any to them. These types of email are considered spam, which is unsolicited, intrusive mail that clogs up the network.
4. Exercise caution when downloading files from the Internet. Ensure that the source is a legitimate one. Verify that an anti-virus program checks the files on the download site. If you're uncertain, don't download the file at all.
5. Update your anti-virus software regularly. Thousands of viruses are discovered each month, so you'll want to be protected.
6. Back up your files on a regular basis. If a virus destroys your files, at least you can replace them with your back-up copy. You should store your backup copy in a separate location from your work files, one that is preferably not on your computer.
7. Use only legal software.
8. Use Antivirus software
9. Be careful when searching on the internet, the links that come up from your search engine may contain a virus. Never go to sites that sound suspicious.
10. Due to the popularity of the social networking websites such as Facebook, Twitter, Whatsapp virus makers target them more than any other website. Online gaming and gambling websites also are high risk websites. It’s best to avoid these kinds of websites altogether.
11. If you see any suspicious pop-ups appear on your screen, do not click on them. If you do, it is very likely you will infect your computer. Instead use the following keyboard command, which will allow you to close the pop-up, without having the click on it or infecting yourself. The keyboard command is ALT + F4.

If that fails, then shut down the computer.

CHAPTER-III : DATA COMMUNICATION AND NETWORK.

Data communication : Exchange of data between two or more devices via transmission media such as wire or cable.

Data communication system has 5 components : (Components of Network)

1) **Message** : It is the information/data to be communicated. Ex. Text, number, pictures, audios, videos. ***e**lectronic **mail** or email is* the transmission of messages over communications networks

2) **Sender** : It is the device that sends data messages. It can be computers, telephones handset etc

3) **Receiver** : It is the device which receives message . It can be computer, telephone, TV set etc

4) **Transmission Medium** : It is the Physical path by which messages travel from sender to receiver. Ex:- Twisted pair wire, Co-axial cable, Fibre optics, radio wave etc

5) **Protocol** : It is a set of rules that govern data communication. Without protocol two devices may be connected but can not communicate and without it no communication is possible. The guidelines are to regulate access method, regulate physical topologies, types of cabling, speed of data transfer.

Data representation : Text, Number, Image, Audio, Video etc is represented in collection of 0's and 1's and each set of bits is called Code.(32 bits/ASCII Code)

Modes of Data Transmission /Data Flow :- Communication between two devices can be Simplex, Half-duplex, Full-duplex. Ex:- Radio wave, Satellite(Microwave), Infrared(TV Remote),Laser

Simplex : Communication is Uni-directional : Sender sends and receiver receives only No reply.
Ex: Radio, Laser etc. and inside computer, Keyboard, Monitors are Simplex devices.

Half-duplex : Each station can both transmit & receive but not at the same time.
Ex: Walkie-talkies, radios, police wireless.

Full-duplex : In full-duplex mode both stations can transmit & receive simultaneously. Ex. Telephone Network

Networking : An interconnected collection of autonomous computers. Autonomous means no computer in the network can start or stop or control others. A network is a set of devices (**nodes**) connected by communication links. Nodes can be Computers, Printers or other devices that can send receive data. Most networks use distributed processing in which a single task is divided into multiple computers instead of a single big machine.

A computer network is defined as interconnected collection of autonomous computers. Computers are said to be connected if they are able to exchange information. Connection is physically established through cables, lasers, microwaves, fibre optics and communication satellite. Autonomous computer means any computer in the network can not forcibly start or stop or control another computer in the network. It is the process of linking two or more computing devices for the purpose of sharing data and other H/W and S/W resources.

A network is a combination of hardware and software that sends data from one location to another. The hardware consists of the physical equipment that carries signals from one point of the network to another. The software consists of instruction sets that make possible the services that we expect from a network.

COMPUTER NETWORK is a system of interconnected computers and peripheral devices to exchange data. Each device in the network is called a Node and each Node has a unique address

Objectives of Computer Network :

Resource Sharing : A huge task is divided into small modules and distributed to various computers present in the network. This technique reduces the load on individual computer but finishes the whole task at a time. This technique is known as parallel processing of job. The goal is to make all programs, data and hardware available to everyone in the network without concern to physical location of the resource and the user.

High Reliability :- This is achieved by replicating the file on two or more machines, so in case due to hardware failure we can use another system to attend ongoing job.

Economical :- Helped the organisation to save money.

Information Sharing(www) i.e services on internet and Improves Communication (Messages can be sent) and security. Disadvantages :- Costs more, System is complex to run, If network fails it creates a havoc.

Applications of Network:

Information services – A www site offering technical specification for a new product of information service.

Electronics Data Exchange – EDI allows business information such as purchase, inventory transferred without using paper.

Teleconferencing – It allows conference to occur without the participants being in the same location. It includes Text conferencing (Participants communicate with their keyboards and computer monitor) Video Conferencing (Participants at a number of locations communicate simultaneously over the phone (talk) , Video Conferencing (Participants can see as well as talk to one another.

Marketing (Online reservation), Financial services (ATM)

Electronics messaging- E-mail - Transfer of messages between users.

Directory services- allow list of files stored in central location to speed up world wide search operation. Engines are Goole, Yahoo

Marketing and sales – teleshopping, online reservation

Financial services – Electronics fund transfer by banks

Manufacturing – CAD/CAM

Cellular Telephone – Wireless Phone connection even while travelling over large distance Cable TV

Types of Data Transmissions :-

Serial and Parallel Transmission :

Transmission modes refers to number of bits that can be transmitted simultaneously over communication channel. Speed of Transmission is Baud rate i.e BPS (Bits per second), KBPS, MBPS,

Digital data transmission can occur in two basic modes: serial or parallel. Data within a computer system is transmitted via parallel mode on **buses** with the width of the parallel bus matched to the word size of the computer system. Data between computer systems is usually transmitted in **bit serial mode** Consequently, it is necessary to make a parallel-to-serial conversion at a computer **interface** when sending data from a computer system into a network and a **serial-to-parallel** conversion at a computer interface when receiving information from a network. The type of transmission mode used may also depend upon distance and required data rate.

Parallel Transmission :

Within a computing or communication device, the distances between different sub-units are too short. Thus, it is normal practice to transfer data between subunits using a separate wire to carry each bit of data. There are multiple wires connecting each sub-unit and data is exchanged using a parallel transfer mode. This mode of operation results in minimal delays in transferring each word.

•In parallel transmission, all the bits of data are transmitted simultaneously on separate communication lines.

•In order to transmit n bits, n wires or lines are used. Thus each bit has its own line.

•All n bits of one group are transmitted with each clock pulse from one device to another i.e. multiple bits are sent with each clock pulse.

•Parallel transmission is used for short distance communication.

In parallel transmission, multiple **bits** (usually 8 bits or a byte/character) are sent simultaneously on different channels (wires, frequency channels) within the same cable, or radio path, and **synchronized** to a clock. Parallel devices have a wider data bus than serial devices and can therefore transfer data in words of one or more bytes at a time. As a result, there is a speedup in parallel transmission bit rate over serial transmission bit rate. However, this speedup is a tradeoff versus cost since multiple wires cost more than a single wire, and as a parallel cable gets longer, the synchronization timing between multiple channels becomes more sensitive to distance. The timing for parallel transmission is provided by a constant clocking signal sent over a separate wire within the parallel cable; thus parallel transmission is considered **synchronous** .

Examples of parallel communication systems

Chanel between CPU & Memory is parallel

EX:-.Computer peripheral buses: ISA, ATA, SCSI, PCI and Front side bus, and the
- once-ubiquitous IEEE-1284 / Centronics "printer port"
- Laboratory Instrumentation bus IEEE-488

Serial Transmission

In serial transmission, bits are sent **sequentially** on the same channel (wire) which reduces costs for wire but also slows the speed of transmission. Also, for serial transmission, some overhead time is needed since bits must be assembled and sent as a unit and then disassembled at the receiver.
Serial transmission can be either synchronous or **asynchronous** . In synchronous transmission, groups of bits are combined into frames and frames are sent continuously with or without data to be transmitted. In asynchronous transmission, groups of bits are sent as independent units with start/stop flags and no data link synchronization, to allow for arbitrary size gaps between frames. However, start/stop bits maintain physical bit level synchronization once detected.
Ex:- Serial **transmitting** data is sending one bit at a time, include **Serial** ATA, **Serial** SCSI, Ethernet cable plugged into Ethernet ports, the Display Data Channel using previously reserved pins of the VGA connector or the DVI port or the HDMI port. serial communication architectures :-
- RS-232 (low-speed, implemented by Serial Ports)

Ex:-Universal Serial Bus (moderate-speed, for connecting computers to peripherals) • Ethernet

Types of Serial Transmission :- Asynchronous & Synchronous
- There are two types of serial transmission - synchronous and asynchronous both these transmissions use '**Bit synchronization**'
- Bit Synchronization is a function that is required to determine when the beginning and end of the data transmission occurs.
- Bit synchronization helps the receiving computer to know when data begin and end during a transmission. Therefore bit synchronization provides timing control.

Asynchronous Transmission
- Asynchronous transmission sends only one character at a time where a character is either a letter of the alphabet or number or control character *i.e.* it sends one byte of data at a time.
- Bit synchronization between two devices is made possible using start bit and stop bit.
- Start bit indicates the beginning of data *i.e.* alerts the receiver to the arrival of new group of bits. A start bit usually 0 is added to the beginning of each byte.
- Stop bit indicates the end of data *i.e.* to let the receiver know that byte is finished, one or more additional bits are appended to the end of the byte. These bits, usually 1s are called stop bits.

- Addition of start and stop increase the number of data bits. Hence more bandwidth is consumed in asynchronous transmission.
- There is idle time between the transmissions of different data bytes. This idle time is also known as Gap
- The gap or idle time can be of varying intervals. This mechanism is called Asynchronous, because at byte level sender and receiver need not to be synchronized. But within each byte, receiver must be synchronized with the incoming bit stream.

Application of Asynchronous Transmission

Asynchronous transmission is well suited for keyboard type-terminals and paper tape devices. The advantage of this method is that it does not require any local storage at the terminal or the computer as transmission takes place character by character.

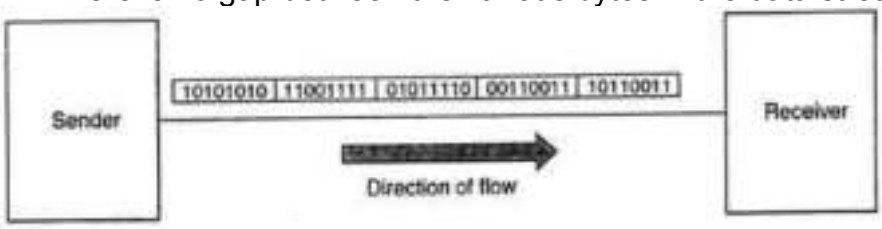
Asynchronous transmission is best suited to Internet traffic in which information is transmitted in short bursts. This type of transmission is used by modems.

Advantages of Asynchronous transmission

This method of data transmission is cheaper in cost as compared to synchronous *e.g.* If lines are short, asynchronous transmission is better, because line cost would be low and idle time will not be expensive.

Synchronous Transmission (Best suitable for communication between two divices)

- Synchronous transmission does not use start and stop bits.
- In this method bit stream is combined into longer frames that may contain multiple bytes.
- There is no gap between the various bytes in the data stream.



Synchronous Transmission

- In the absence of start & stop bits, bit synchronization is established between sender & receiver by *'timing'* the transmission of each bit.
- Synchronous transmission is used for high speed communication between computers.

Advantage of Synchronous transmission

- This method is faster as compared to asynchronous as there are no extra bits (start bit & stop bit) and also there is no gap between the individual data bytes.

Disadvantages of Synchronous transmission

- It is costly as compared to asynchronous method. It requires local buffer storage at the two ends of line to assemble blocks and it also requires accurately synchronized clocks at both ends. This lead to increase in the cost.
- The sender and receiver have to operate at the same clock frequency. This requires proper synchronization which makes the system complicated.

Sr. No.	Factor	Asynchronous	Synchronus
1.	Data send at one time	Usually 1 byte	Multiple bytes
2.	Start and Stop bit	Used	Not used
3.	Gap between Data units	Present	Not present
4.	Data transmission speed	Slow	Fast
5.	Cost	Low	High

Types of Network connection :

1) Peer-to-peer network 2) Centralised network 3) Client-Server network

Peer-to-peer :

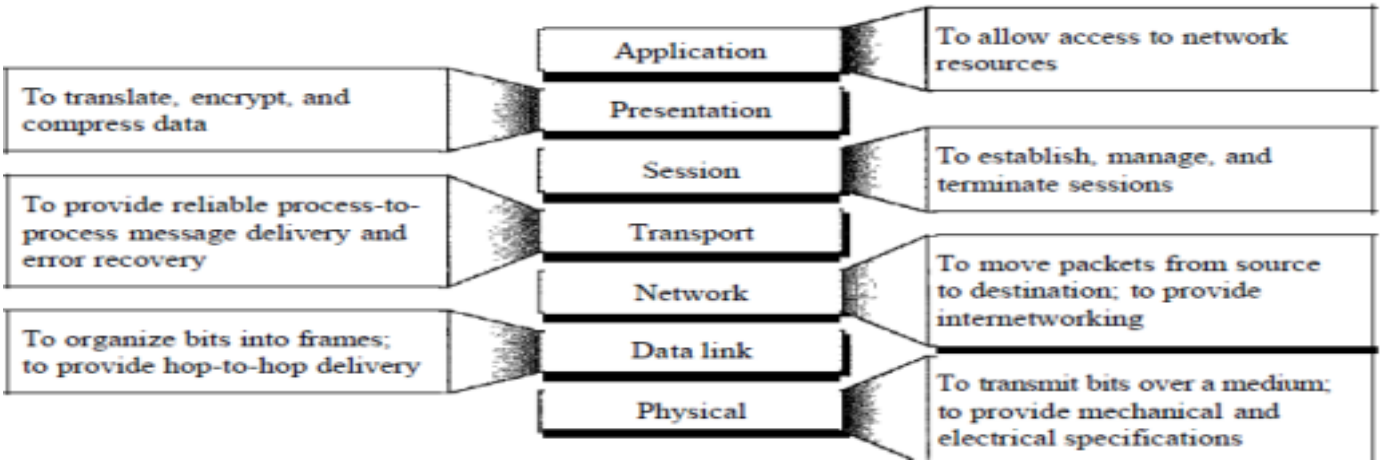
In this network each terminal is known as peer. Each peer in network can work independently. Each peer can access information from any other peer in the network. Commonly used in Micro-computers.

Centralised : It has one mainframe computer with many terminals. The terminals are depending on the mainframe computer (Server). The data transfer is based on request from terminals to mainframe. Here the mainframe computer has full control over the entire network.

Client-Server : It is most powerful & wide ranging network than peer-to-peer. It supports larger number of clients & link the multiple plat forms like DOS, Windows, Unix, Linux etc. Here higher capacity of system act as Server & will have full control over the network.

Server : Dedicated or Non-dedicated. Server.
Specialised Server : File server, Print server, Application Server, Mail server, Fax server etc.

Network Architecture (Design) : Network Model : (Network Architecture is a structured way & organised in series of layers.
Computer networks are created by different entities. Standards are needed so that these heterogeneous networks can be communicated with one another. The two best known standards are OSI model and Internet model.
OSI consists of 7 layers
Internet model consists of 5 layers



What is Protocol :
A protocol is a set of rules that governs the communications between computers on a network. In order for two computers to talk to each other, they must be speaking the same language. Many different types of network protocols and standards are required to ensure that your computer (no matter which operating system, network card, or application you are using) can communicate with another computer located on the next desk or half-way around the world.

One of the most common and known protocols is HTTP (Hyper Text Transfer Protocol), which is a protocol used to transmit data over the world wide web (Internet).

TCP/IP>Application Layer Eg. WWW, FTP, IRC, Email, telnet,>DATA
Transport Layer Eg. TCP, UDP > SEGMENT
Network Layer Eg. IP > PACKETS
Link Layer Eg. Ethernet, WiFi > Frames
Physical Layer Eg. Ethernet Cable, fiber-optics > Bits

The OSI (Open Systems Interconnection) Reference Model defines 7 Layers of Networking Protocol :-

OSI Layer	Name	Common Protocols
7	Application	HTTP | FTP | SMTP | DNS | Telnet Application layer are users, whether human or software, to access the network . It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services
6	Presentation	The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems. the relationship between the presentation layer and the application and session layers
5	Session	HTTP, FTP (Several protocols overlap the session, presentation, and application layers of networks) The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes, It establishes,

		maintains, and synchronizes the interaction among communicating systems
4	Transport	TCP | SPX (e-mail messages broken to transportation of the data packets)
3	Network	TCP/IP| IPX charge of routing network messages (data) from one computer to another, Whereas the network layer oversees source-to-destination delivery of individual packets Network layer is responsible for the source-to-destination delivery of a packet
2	Data Link	Ethernet Hardware elements, cables, repeaters, Nic, network addresses
1	Physical	The data link layer divides the stream of bits received from the network layer into manageable data units called frames. The Physical layer contains data in form of 0's and 1's

Network Topologies : The way in which network is connected physically. The pattern of interconnection of nodes in a network is called Topology. It is the Schematic description of the arrangement of the network.

1) Mesh 2)Star 3)Bus 4)Ring

A topology is a network's virtual shape or structure. This shape does not necessarily correspond to the actual physical layout of the devices on the network. For example, the computers on a home LAN may be arranged in a circle in a family room, but it would be highly unlikely to find a ring topology there. Network topologies are categorized into the following basic types:

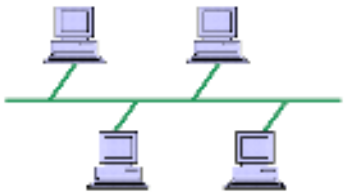
- bus
- ring
- star
- tree
- mesh

More complex networks can be built as hybrids of two or more of the above basic topologies.

Bus Topology :-

Adv : Easy installation

Dis : Difficult to find fault. A fault in the bus stops all transmission.



Bus networks (not to be confused with the system bus of a computer) use a common backbone to connect all devices. A single cable, the backbone functions as a shared communication medium that devices attach or tap into with an interface connector. A device wanting to communicate with another device on the network sends a broadcast message onto the wire that all other devices see, but only the intended recipient actually accepts and processes the message. Ethernet bus topologies are relatively easy to install and don't require much cabling compared to the alternatives. 10Base-2 ("ThinNet") and 10Base-5 ("ThickNet") both were popular Ethernet cabling options many years ago for bus topologies. However, bus networks work best with a limited number of devices. If more than a few dozen computers are added to a network bus, performance problems will likely result. In addition, if the backbone cable fails, the entire network effectively becomes unusable.

Ring Topology:

Adv: Dedicated point to point connection. Easy to install and reconfigure.

Dis: Unidirectional traffic. In a simple ring, a break in the ring can disable the entire network.

In a ring network, every device has exactly two neighbours for communication purposes. All messages travel through a ring in the same direction (either "clockwise" or "counter clock wise"). A failure in any cable or device breaks the loop and can take down the entire network. To implement a ring network, one typically uses FDDI(Fibre Distributed Data Interface), SONET, or Token Ring technology. Ring topologies are found in some office buildings or school campuses.

Star Topology:
Adv : Easy to install and re-configure. Less expensive. If one link fails, only that link is affected.
Dis: Depends upon one node. If the hub or central controller fails , nodes attached are disabled.

Many home networks use the star topology. A star network features a central connection point called a "hub" that may be a hub, switch or router. Devices typically connect to the hub with Unshielded Twisted Pair (UTP) Ethernet. Compared to the bus topology, a star network generally requires more cable, but a failure in any star network cable will only take down one computer's network access and not the entire LAN. (If the hub fails, however, the entire network also fails.)

Mesh Topology :
(Every device has a dedicated point-to-point link to every other device.
Adv : Eliminate traffic possible
Dis: Cable excess use, installation and reconnection is difficult & expensive.

Mesh topologies involve the concept of routes. Unlike each of the previous topologies, messages sent on a mesh network can take any of several possible paths from source to destination. (Recall that even in a ring, although two cable paths exist, messages can only travel in one direction.) Some WANs, most notably the Internet, employ mesh routing. A mesh network in which every device connects to every other is called a full mesh. As shown in the illustration below, partial mesh networks also exist in which some devices connect only indirectly to others.
Topologies remain an important part of network design theory. You can probably build a home or small business computer network without understanding the difference between a bus design and a star design, but becoming familiar with the standard topologies gives you a better understanding of important networking concepts like hubs, broadcasts, and routes.

Categories of Network: Types of Network :
LAN, WAN, MAN
Interconnection of Network (**Internetwork**) i.e. LAN, MAN and WAN connected to each other.
Most common types of computer networks in order of scale from less number of computers and geographical area coverage.

Local Area Network/LAN Metropolitan Area Network/MAN Wide Area Network/ WAN
A LAN connects network devices over a relatively short distance. A networked office building, school, or home usually contains a single LAN, though sometimes one building will contain a few small LANs (perhaps one per room), and occasionally a LAN will span a group of nearby buildings. In TCP/IP networking, a LAN is often but not always implemented as a single IP subnet.
In addition to operating in a limited space, LANs are also typically owned, controlled, and managed by a single person or organization. They also tend to use certain connectivity technologies, primarily Ethernet and Token Ring.
A LAN is a network that is used for communicating among computer devices, usually within an office building or home. A LAN IS ALWAYS A WIRED MEDIA & RATE OF DATA TRANSMISSION IS HIGHEER• LAN's enable the sharing of resources such as files or hardware devices that may be

needed by multiple users • Is limited in size, typically spanning a few hundred meters, and no more than a mile • Is fast, with speeds from 10 Mbps to 10 Gbps • Requires little wiring, typically a single cable connecting to each device • Has lower cost compared to MAN's or WAN's
LAN's can be either wired or wireless. Twisted pair, coax or fibre optic cable can be used in wired LAN's. • Every LAN uses a protocol – a set of rules that governs how packets are configured and transmitted. • Nodes in a LAN are linked together with a certain topology. These topologies include: – Bus – Ring – Star • LANs are capable of very high transmission rates (100s Mb/s to G b/s).
Adv : Speed • Cost • Security • E-mail • Resource Sharing
Dis : Expensive To Install • Requires Administrative Time • File Server May Fail • Cables May Break

A network covering a small geographic area, like a home, office, or building. Current LANs are most likely to be based on Ethernet technology. For example, a library may have a wired or wireless LAN for users to interconnect local devices (servers) and to connect to the internet. On a wired LAN, PCs in the library are typically connected by category 5 (Cat5) cable, running the IEEE 802.3 protocol through a system of interconnection devices and eventually connect to the internet. The cables to the servers are typically on Cat 5e enhanced cable, which will support IEEE 802.3 at 1 Gbit/s. A wireless LAN may exist using a different IEEE protocol, 802.11b or 802.11g. The staff computers (bright green in the figure) can get to the color printer, checkout records, and the academic network *and* the Internet. All user computers can get to the Internet and the card catalog. Each workgroup can get to its local printer. Note that the printers are not accessible from outside their workgroup. The defining characteristics of LANs, in contrast to WANs (wide area networks), include their higher data transfer rates, smaller geographic range, and lack of a need for leased telecommunication lines. Current Ethernet or other IEEE 802.3 LAN technologies operate at speeds up to 10 Gbit/s. This is the data transfer rate. IEEE has projects investigating the standardization of 100 Gbit/s, and possibly 40 Gbit/s.

Metropolitan Area Network (MAN) :-
A metropolitan area network (MAN) is a large computer network that usually spread over a city or a large campus. • A MAN is optimized for a larger geographical area than a LAN, ranging from several blocks of buildings to entire cities. • A MAN might be owned and operated by a single organization, but it usually will be used by many individuals and organizations.
A MAN often acts as a high speed network to allow sharing of regional resources. • A MAN typically covers an area of between 5 and 50 km diameter. • Examples of MAN: Telephone Company network that provides a high speed DSL to customers and Cable TV network.

A Metropolitan Area Network is a network that connects two or more Local Area Networks or Campus Area Networks together but does not extend beyond the boundaries of the immediate town/city. Routers, switches and hubs are connected to create a Metropolitan Area Network.

Wide Area Network (WAN)
WAN covers a large geographic area such as country, continent or even whole of the world. • A WAN is two or more LANs connected together. The LANs can be many miles apart. • To cover great distances, WANs may transmit data over leased high-speed phone lines or wireless links such as satellites. A WAN IS ALWAYS IS WIRELESS MEDIA & RATE OF DATA TRANSMISSION IS LOWER THAN LAN.
Multiple LANs can be connected together using devices such as bridges, routers, or gateways, which enable them to share data. • The world's most popular WAN is the Internet.

WAN - Wide Area Network
As the term implies, a WAN spans a large physical distance.
A WAN is a geographically-dispersed collection of LANs. A network device called a router connects LANs to a WAN. In IP networking, the router maintains both a LAN address and a WAN address.
A WAN differs from a LAN in several important ways. Most WANs (like the Internet) are not owned by any one organization but rather exist under collective or distributed ownership and management. WANs tend to use technology like ATM, Frame Relay and X.25 for connectivity over the longer distances.

A WAN is a data communications network that covers a relatively broad geographic area (i.e. one city to another and one country to another country) and that often uses transmission facilities provided by common carriers, such as telephone companies. WAN technologies generally function at the lower three layers of the OSI reference model: the physical layer, the data link layer, and the network layer.

Networking Devices like Hub, Repeater, Switch, Bridge, Router, Gateway & NIC - Network Hardware

Transmission Media

The first thing to consider is how you plan to transmit data across the network. The **transmission media** of a computer network describes the material substances that carry energy waves, which include the data being transferred. The two main categories are wired, or guided, which uses physical cables, and wireless, or unguided, which uses electromagnetic waves that can travel through a vacuum or a medium, such as air. Wireless signals do not require a physical medium, such as cables. The most commonly used wired connections use twisted-pair cables, coaxial cables and fibre optic cables. Twisted-pair cables consist of individual copper wires that are twisted into pairs. The wires are wrapped in an insulation material. Twisted-pair cables are widely used for telephone service. A coaxial cable consists of a copper or aluminium wire wrapped inside an insulating layer. Most cable TV companies use coaxial cables. A fibre optic cable consists of a central fiberglass core surrounded by several layers of protective material. This type of cable transmits light rather than electronic signals. A light emitting diode (LED) or laser is used to create the light pulses. The transmission speed of a fibre optic cable is much faster compared to the other cables. Most networks built today use a fibre optic cable because of its superior speed, but coaxial cable is also very common. The most commonly used wireless connections use radio waves, microwaves and infrared waves. You obviously don't need cables for a wireless connection, but you will need other hardware for the transmission of wireless signals through the air.

Hubs, Repeaters, Bridges and Switches

A H**ub** is typically the least expensive, least intelligent, and least complicated of the networking device used to connect multiple computers/devices directly using cables. Each connection is called a Port. Its job is very simple – anything that comes in one port is sent out to the others i.e Hub distributes the data to the other ports in the network. Typically, a hub sends all the data it receives to all the other ports.

A **switch** does essentially what a hub does, but more efficiently. By paying attention to the traffic that comes across it, it can “learn” where particular addresses are. Initially, a switch knows nothing and simply sends on incoming messages to all ports: Even accepting that first message, however, the switch has learned something–it knows on which connection the sender of the message is located. Thus, when machine responds to the message, the switches only need to send that message out to the one connection: When a network contains a large number of devices, switches are needed instead of hubs to make sure the communications between devices does not slow down. Contrary to hubs, switches send the data it receives only to specific ports.

A R**epeater** is a networking device that receives electronic signals, cleans them and retransmits them or amplify at a higher power level. Signals transmitted over cable tend to degrade over long distances. Repeaters are needed so that the signal can travel longer distances.

Bridges are networking devices that divide up the network into different segments to manage the amount of traffic. This prevents unnecessary traffic from entering other parts of the network and reduces congestion. As a network becomes more complex, bridges make sure your network speed doesn't drop dramatically.

Routers are communication devices used to connect two different networks/protolols. A router sorts incoming data and distributes it to the correct destination. For example, if you have a network within a single office building, many different devices from within the network may access resources outside

the network. The best example of this would be the Internet. A router ensures that requests from within the network for information over the Internet are distributed to the correct computer within the network. The Internet itself uses numerous routers to direct all the traffic taking place. Such routers are typically very large and heavy-duty pieces of hardware, designed to handle huge amounts of data traffic. Routers can be used for wired connection, wireless connections or both. A router that provides a wireless connection is referred to as a 'wireless router.' While routers are used to connect different networks, they only work if the network protocols are the same.

Gateways interfaces networks that use and protocol conversion. A gateway is actually a node on a network that serves as a entrance to another network. You can think of a gateway as a router that includes protocol translators. The terms 'router' and 'gateway' are often used interchangeably, but it is important to remember that only gateways make it possible to connect networks using different protocols. Since the Internet and many other computer networks use the same TCP/IP protocols, routers are sometimes all that is needed for a particular network. However, any network that also includes a mainframe system will need a gateway since this type of network uses different communication protocols.

Modem is used to modulate and demodulate data signals. The term itself is a combination of the first two letters of modulator and the first three letters of demodulator. What does this really mean? While all computer data is digital, signals over certain types of connections are analog. A modem is used to encode digital information onto an analog carrier signal and to decode the transmitted information back to digital.

NIC or Network interface controller (**NIC**, also known as a **network interface card**, **network** adapter, LAN adapter, and by similar terms) is a **computer** hardware component that connects a **computer** to a **computer network**. Short for **Network Interface Card**, the **NIC** is also referred to as an **Ethernet card** and **network adapter**. It is an expansion card that enables a computer to connect to a network; such as a home network, or the Internet using an Ethernet cable with an RJ-45 connector. Due to the popularity and low cost of the Ethernet standard, most new computers have a network interface build directly into the motherboard.

What is the Internet ? The Internet or simply the Net is a interconnection of large or small networks around the Globe. It is often referred to as a network of networks. It is The Internet is a global network connecting millions of computers. It allows computer users to communicate with each other across distance and computer platforms. More than 190 countries are linked into exchanges of data, news and opinions. The number of Internet users represents nearly 40 percent of the world's population. The largest number of Internet users by country is China, followed by the United States and India. The number of websites with a unique hostname online exceeded 1 billion. The Internet began in 1969 as the U.S. Department of Defense's Advanced Research Project Agency (**ARPA**) to provide immediate communication within the Department in case of war. Computers were then installed at U.S. universities with defense related projects. As scholars began to go online, this network changed from military use to scientific use. As **ARPAnet** grew, administration of the system became distributed to a number of organizations, including the National Science Foundation (NSF). This shift of responsibility began the transformation of the science oriented ARPAnet into the commercially minded and funded Internet used by millions today.

The Internet acts as a pipeline to transport electronic messages from one network to another network. At the heart of most networks is a server, a fast computer with large amounts of memory and storage space. The server controls the communication of information between the devices attached to a network, such as computers, printers, or other servers.

An Internet Service Provider (ISP) allows the user access to the Internet through their server. Many teachers use a connection through a local university as their ISP because it is free. Other ISPs, such as America Online, telephone companies, or cable companies provide Internet access for their members. You can connect to the Internet through telephone lines, cable modems, cellphones and other mobile devices.

Internet Service Provider, refers to a company that provides Internet services, including personal and business access to the Internet. For a monthly fee, the service provider usually provides a software package, username, password and access phone number. Equipped with a modem, you can then log on to the Internet and browse the World Wide Web and USENET, and send and

receive e-mail. For broadband access you typically receive the broadband modem hardware or pay a monthly fee for this equipment that is added to your ISP account billing.

Who Owns the Internet?

No one actually owns the Internet, and no single person or organization controls the Internet in its entirety. The Internet is more of a concept than an actual tangible entity, and it relies on a physical infrastructure that connects networks to other networks.

Is Web and Internet the Same?

The Internet is **not** synonymous with *World Wide Web*. **The Internet is a massive network of networks**, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. **The World Wide Web, or simply Web**, is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet.

WWW : World Wide Web is a set of protocols that allows to access any document on the net through a naming system based on URL. Sir Tim Berners Lee invented World Wide Web in 1989.

The World Wide Web is a system of Internet servers that support specially formatted documents. The documents are formatted in a markup language called **HTML** (*HyperText Markup Language*) that supports links to other documents, as well as graphics, audio, and video files. This means you can jump from one document to another simply by clicking on hot spots. Not all Internet servers are part of the World Wide Web.

TCP/IP protocols : In order for a computer to communicate on the Internet, a set of rules or protocols computers must follow to exchange messages was developed. The two most important protocols allowing computers to transmit data on the Internet are Transmission Control Protocol (TCP) and Internet Protocol (IP). With these protocols, virtually all computers can communicate with each other. For instance, if a user is running Windows on a PC, he or she can communicate with iPhones.

DNS : Domain name system is an Internet address has four fields with numbers that are separated by periods or dots. This type of address is known as an IP address. Rather than have the user remember long strings of numbers, the Domain Name System (DNS) was developed to translate the numerical addresses into words. For example, the address fcit.usf.edu is really 131.247.120.10.

URLs :Addresses for web sites are called URLs (Uniform Resource Locators). Most of them begin with http (HyperText Transfer Protocol), followed by a colon and two slashes. For example, the URL for the Florida Center for Instructional Technology is http://fcit.usf.edu/ .

Some of the URL addresses include a directory path and a file name. Consequently, the addresses can become quite long. For example, the URL of a web page may be:

http://fcit.usf.edu/holocaust/default.htm. In this example, "default.htm" is the name of the file which is in a directory named "holocaust" on the FCIT server at the University of South Florida.

Top-level domain : Each part of a domain name contains certain information. The first field is the host name, identifying a single computer or organization. The last field is the top-level domain, describing the type of organization and occasionally country of origin associated with the address.

Domain names are :-
.com Commercial
.edu Educational

.gov US Government
.int Organization
.mil US Military
.net Networking Providers
.org Non-profit Organization

Domain name country codes include, but are not limited to:

.au Australia
.de Germany
.fr France
.nl Netherlands
.uk United Kingdom
.us United States

Why do I need a browser? : Once you have an account with an Internet service provider, you can access the Web through a browser, such as Safari or Microsoft Internet Explorer. The browser is the application responsible for allowing a user's computer to read and display web documents.

Hypertext Markup Language (HTML) is the language used to write web pages. A browser takes the HTML and translates it into the content you see on the screen. You will note your cursor turns into a pointing finger over some images or text on the page. This indicates a link to additional information and it can be either a link to additional web pages, email, newsgroups, audio, video, or any number of other exciting files.

For example, if you were to click on Florida Department of Education your browser would link to the Florida Department of Education home page and that web page would open in your screen.

What is Internet Explorer?

A Web browser that made its debut in 1995 as Microsoft's response to Netscape, one of the first graphical-based Web browsers and, at the time, the dominant browser in use with control of over 90 percent of the market.
Initially called ***M**icrosoft **I**nternet **E**xplorer* (MSIE), ***I**nternet **E**xplorer*(IE) has long held the title of most popular browser in use, despite strong competition from the likes of **Mozilla Firefox**, **Apple's Safari**, **Opera** and **Google Chrome**.

What is Internet address?

Everyone is acquainted with the usual type of address or URL (Uniform Resource Locator) used for Web sites. Things like “www.microsoft.com” have become as familiar to us as street addresses. The word “dotcom” has even become part of the language (and not just in English). However, the common form of addresses that we use, which contain letters and recognizable names, is for the convenience of humans only and is not actually the kind of Web addresses used by computers.
An Internet address uniquely identifies a node on the Internet. Internet address may also refer to the name or IP of a Website(URL). The term Internet address can also represent someone's e-mail address.

What is URL?

Abbreviation of ***U**niform **R**esource **L**ocator (URL)* it is the global address of documents and other resources on the World Wide Web.
The first part of the URL is called a *protocol identifier* and it indicates what protocol to use, and the second part is called *are source name* and it specifies the IP address or the domain name where the resource is located. The protocol identifier and the resource name are separated by a colon and two forward slashes. A **URL** is one type of Uniform Resource Identifier (URI); the generic term for all types of names and addresses that refer to objects on the World Wide Web. The term "Web address" is a synonym for a **URL** that uses the HTTP / HTTPS protocol.

For example, the two URLs below point to two different files at the domain *pcwebopedia.com.* The first specifies an executable file that should be fetched using the FTP protocol; the second specifies a Web page that should be fetched using the HTTP protocol:

What is domain name?

Domain names are used to identify one or more *IP addresses*. For example, the domain name *microsoft.com* represents about a dozen IP addresses. Domain names are used in URLs to identify particular Web pages. For example, in the URL *http://www.pcwebopedia.com/index.html,* the **domain name is*pcwebopedia.com.***

Every domain name has a suffix that indicates which top level domain (**TLD**) it belongs to. There are only a limited number of such domains. For example:

gov - Government agencies **edu** - Educational institutions **org** – Organizations (nonprofit)
mil – Military **com** - commercial business **net** - Network organizations

Because the Internet is based on IP addresses, not domain names, every Web server requires a Domain Name System (DNS) server to translate domain names into IP addresses.

What is Internet marketing?

Internet marketing, or online marketing, refers to advertising and marketing efforts that use the Web and email to drive direct sales via electronic commerce, in addition to sales leads from Web sites or emails. Internet marketing and online advertising efforts are typically used in conjunction with traditional types of advertising like radio, television, newspapers and magazines.

Chatting :- It is the most fantastic thing on the internet. It is like text phone. In chatting you type message on the screen , which is immediately received by the recipient then recipient can type message in response which is received back immediately.

Internet (Web) Conferencing :- It is a form of real time communication in which multiple computer users all connected to the internet. Web conferencing allows users to carry on business meetings and seminars , make presentations, conduct demonstration, provide online education and offer direct customer support.

Electronic Newspaper :- It is online version of newspaper. Going online created more opportunities for newspaper such as competing with broadcast journalism in presenting breaking news in a more timely manner.

Online Shopping :- It involves purchasing products or services over the internet. All the products in online stores are described through text, with photos and with multimedia files. We can purchase the goods with cashless transaction.

Application of Internet :

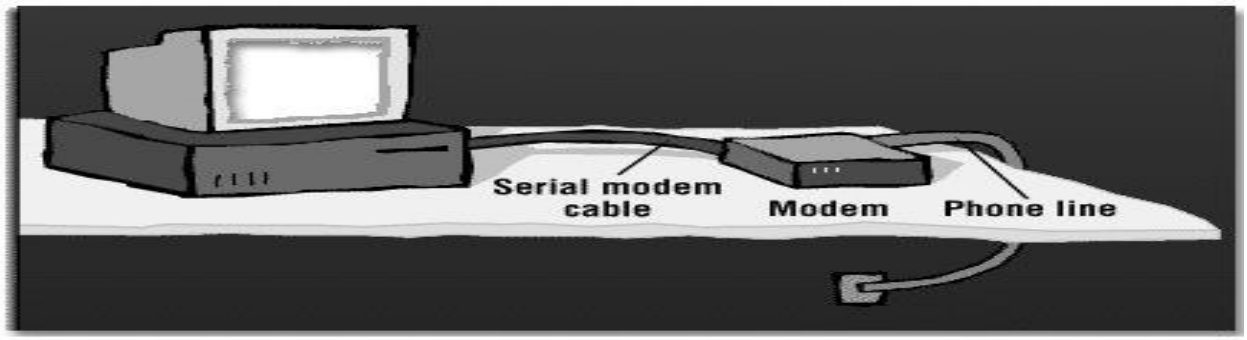
1) Exchange messages using e-mail (Electronic mail).
2) Transfer files as well as software.
3) Browse through information on any topic on web.
4) Communicate in real time (chat) with others connected to the Internet.
5) Search databases of government, individuals and organizations.
6) Read news available from leading news groups.
7) Send or receive animation and picture files from distant places.
8) Set up a site with information about your company's products and services.

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Getting Internet Connection on your Computer :-

Any terminal to activate internet services on it require a internet connection from a Internet Service Provider. To have a connection one needs to contact the Internet Service Providers. There are several Internet Service Providers (ISP) in each locality. There are some nominal charges that you

need to pay to the ISP for the installation of the connection and for the rent (either monthly or annually). Depending on the requirement you can choose any one of the available options. If you need a high speed dedicated network then you can opt for a high bandwidth broadband or leased line connection.
If you need to have connection on a single PC then you choose a low bandwidth or dialup connection For small services at your hand you can even have mobile phones connected to your Internet. There are many ways to get connected to the Internet. You can get internet connection in any of the following ways:-
1. through dial-up connection : user is supposed to get connected to the internet after dialing up the number used for connection. This is useful if your network is either confined to small group of computers or for a single PC.
2. through leased lines: in this a dedicated line is laid specifically for connection.
3. though broadband: in this you can get a broadband connection which provides a high bandwidth for the internet connection. This also provides a good speed. These days even wireless connections are available. For this you need to have a Wi Fi card attached to your computer which can be useful if you do not have proper place to lay down the wires.
To connect to the Internet you need a PC (personal computer) with requisite software including a browser, a telephone connection or a leased line, and a modem, which allows the PC to communicate with other computers.



Types of Internet Connections :-

(a) Dial-Up connection :-

This is the most common basic type of connection available from ISPs (Internet Server Providers). In Dial-up connection, we use our computer, dial a phone number (provider by ISP) to get connected to Server at Providers end through which we access Internet. It means we are not directly connected to Internet; we access the Internet through an Internet Service Provider.

(b) ISDN (Integrated Services Digital Network) connection :-

The process of connecting to server to access Internet is almost same as Dial-up, but it offers connectivity through the use of Introduction to Internet :: 245 digital phone lines instead of Analog. It offers Internet connectivity at speeds of up to 128 Kbps, allows the user to receive or make calls simultaneously on the same line. ISDN comes through a regular telephone wire from the telephone pole on the street. The line combines two 64 Kbps channels to offer 128 Kbps bandwidth broken into three bands: One band for the ringing signal of your phone, one band for your telephone conversation, and one band for data transfer.

(c) Leased Line Connection (Direct Internet Access)

A “permanent connection” between a computer system (single CPU or LAN, and the Internet). It is generally used by larger institutions, corporate and government agencies. It involves establishing your own Internet gateway (connection) and payment to have a direct full time line with the network. Your computers, in effect, become part of the Net. The main advantage of this connection is that: it is on line – 24 hrs a day, seven days a week, (24×7) and provides faster access. Dedicated links are established through an internet service provider who places a computer-controlled router (message director) at your site. A router is used to connect your local network to the Internet, allow all the members of network to have complete access to Internet.

(d) DSL (Digital Subscriber Line or Dedicated Service Line) connection :-

Broadband Connection DSL, an “always-on’ data connection is becoming widely available these days. It can provide an excellent Internet connection. It connects your home or office to the Internet through the same telephone wire that comes from telephone pole on the street. Like ISDN, with DSL, user can make and receive telephone calls while connected to the Internet. The difference between DSL and dialup / ISDN is that a DSL Internet connection uses a high-speed dedicated circuit filtering out standard phone calls and Internet signals.
(e) VSAT connection. etc

Internet Services are : - E-Mail, WWW, FTP, Chatting, Internet Conferencing

Electronic Mail (E-mail) :- Email uses TCP/IP, SMTP(Simple Mail Transfer Protocol), File attachments, MIME (Multipurpose Internet Mail Extension).
E-mail is delivered by TCP/IP, Gateways translate e-mail formats, Routers internal/external.

The World Wide Web (www) :-
- Like an Internet library with millions of books and documents
- Non-Linear structure (documents read in any order)
- Navigate by clicking on Hypertext links
- Features of the WWW – Graphical, Easy to use, Cross platform, Distributed, Dynamic,Interactive

File Transfer Protocol (FTP) :-
- Used to transfer files (any type) from one computer to another
- FTP sites all use Login and Password
  Anonymous FTP sites (anonymous login and E-mail address as password)

Chat and Instant Messaging :-
- Instant Messaging (IM) – usually just between two people, but may be more .
- Facebook Chat, Skype etc. > Internet Relay Chat (IRC) – requires software on client – many chat rooms (channels) > Though text-based, still very popular particularly in the tech community.

Internet Conferencing or Web conferencing refers to a service that allows conferencing events to be shared with remote locations. These are sometimes referred to as webinars or, for interactive conferences, online workshops. In general, the service is made possible by Internet technologies, particularly on CP/IPconnections. The service allows real-timepoint-to-point communications as well as multicastcommunications from one sender to many receivers. It offers data streams of text-based messages, voice and video chat to be shared simultaneously, across geographically dispersed locations. Applications for web conferencing include meetings, training events, lectures, or short presentations from any computer.

Electronic Online newspaper :-
An online newspaper is the online version of a newspaper, either as a stand-alone publication or as the online version of a printed periodical. Going online created more opportunities for newspapers,

such as competing with broadcast journalism in presenting breaking news in a more timely manner. Online shopping or e-shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser. Alternative names are: e-web-store, e-shop, e-store, Internet shop, web-shop, web-store, online store, online storefront and virtual store.

Different types of Internet connectivity and ISP :-

There are several different ways to connect to the Internet that will give you a powerful connection and let you browse the Internet in no time. Depending on where you live, not all Internet connections are available, so you may have to shop around. Here are five types of Internet connections for you to look into.

4) Modem :-

A modem is a device or program that enables a computer to transmit data over, for example, telephone or cable lines. Computer information is stored digitally, whereas information transmitted over telephone lines is transmitted in the form of analog waves. A modem converts between these two forms.

- Modulator/demodulator
- A device that converts analog signal to digital (modulation) and vice versa (demodulation)
- Speed 1200/2400/9600 bps 14.4/28.8/33.6 Kbps 56 Kbps

Packet : A small chunk of data transmitted over the Internet

Host & IP Address : "A host is a computer connected directly to the Internet" . Each host needs an IP address.

IP address :- A 32-bit number, arranged in 4 numbers seperated by "."

Eg. 74.125.19.147

Domain Name System (DNS) Domain name to IP address conversion

Eg. www.google.com → ??.???.??.?? Domain name or IP address lookup

http://cqcounter.com/whois/

Domain Names are : .com, .edu, .net, .org, .gov, .mil 2nd level : google.com, sony.co.jp

Internet connection options vary by Internet Service Provider and by region. Customers should consider some of the following factors before selecting an Internet package: speed or bandwidth, cost, availability, reliability and convenience. In order to determine what Internet plan is right for you, we recommend you review the different types of Internet connections available on the market today.

Internet connections :-

Wireless *:*

Radio frequency bands are used in place of telephone or cable networks. One of the greatest advantages of wireless Internet connections is the "always-on" connection that can be accessed from any location that falls within network coverage. Wireless connections are made possible through the use of a modem, which picks up Internet signals and sends them to other devices.

WiFi is the best option if you have a laptop or handheld devices we want to use around the house. We can set the WiFi connection up on all of your wireless devices and use them from anywhere in the house, perhaps even outside if the signal is strong enough. You need to be careful to keep your WiFi protected from hackers. Be sure to set up a password that only you know.

Bandwidth :

- Data rate measured in bits (not bytes) per seconds
- Kbps (Kilobits per seconds) 125 chars/sec Mbps (Megabits per seconds) 1,250 chars/sec
- Gbps (Gigabits per seconds) 12,500 chars/sec

- Connecting the Internet : (Requirement)
- A Computer, an account with an ISP (Internet Service Provider)
- A Modem (Modulator/Demodulator) for dial-up services or A NIC (Network Interface Card) for DSL/Cable services

Mobile :- Many cell phone and smartphone providers offer voice plans with Internet access. Mobile Internet connections provide good speeds and allow you to access the Internet on the go.

Hotspots :- Hotspots are sites that offer Internet access over a wireless local area network (WLAN) by way of a router that then connects to an Internet service provider. Hotspots utilize Wi-Fi technology, which allows electronic devices to connect to the Internet or exchange data wirelessly through radio waves. Hotspots can be phone-based or free-standing, commercial or free to the public.

Cable :– the Most Common High-Speed Internet Connection.

Cable : Cable Internet connection is a form of broadband access. Through use of a cable modem, users can access the Internet over cable TV lines. Cable modems can provide extremely fast access to the Internet.

Satellite In certain areas where broadband connection is not yet offered, a satellite Internet option may be available. Similar to wireless access, satellite connection utilizes a modem.

Cable is the most common type of high-speed Internet used today. You can easily add Internet service to your current cable package for a nominal monthly fee. Most cable companies offer different packages depending on the speed you require for your Internet needs, and the number of computers in your house that will run off of it. They'll usually offer high-speed Internet at a reduced cost if purchased as part of a package deal.

Dial-Up :- Dial-up connections require users to link their phone line to a computer in order to access the Internet. This particular type of connection—also referred to as analog—does not permit users to make or receive phone calls through their home phone service while using the Internet.

Broadband :- This high-speed Internet connection is provided through either cable or telephone companies. One of the fastest options available, broadband Internet uses multiple data channels to send large quantities of information. The term broadband is shorthand for broad bandwidth. Broadband Internet connections such as DSL and cable are considered high-bandwidth connections. Although many DSL connections can be considered broadband, not all broadband connections are DSL.

DSL :- DSL, which stands for Digital Subscriber Line, uses existing 2-wire copper telephone line connected to one's home so service is delivered at the same time as landline telephone service. Customers can still place calls while surfing the Internet. DSL Internet connection is usually provided by your local phone company and can also be modified to provide Internet service to a certain number of computers in your home. The speed is slightly different than a cable modem provides, but the cost may make the difference worthwhile. Another advantage to DSL over cable is you're not sharing the modem line with other subscribers in your area. The line is hardwired directly to your house, so there is no "down time" when a lot of people are logged on. You might need to have at least one phone line connected through the company in order to add DSL.

ISDN :- (Integrated Services Digital Network) allows users to send data, voice and video content over digital telephone lines or standard telephone wires. The installation of an ISDN adapter is required at both ends of the transmission—on the part of the user as well as the Internet access provider.

CHAPTER - IV : File Management & Data Processing

What is a File ?

A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user. It is an Electronic document. It can be an ordinary text file or an executable file. A file is known with a name. it consists of alphabets, numeric and special symbols. The file name consists of two parts (i) Primary name (ii) Secondary name. The secondary name represents the type of the file.

What is a Folder ? A folder is a collection of multiple files it is otherwise known as folder. A folder can contain another folder called sub folder which helps for organising files.

File Structure : File structure is a structure, which is according to a required format that operating system can understand. A file has a certain defined structure according to its type.

A text file is a sequence of characters organized into lines.

A source file is a sequence of procedures and functions.

An object file is a sequence of bytes organized into blocks that are understandable by the machine.

When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

File Type :- File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc. Many operating systems support many types of files. Operating system like MS-DOS and UNIX have the following types of files:

Ordinary files :- These are the files that contain user information.

These may have text, databases or executable program.

The user can apply various operations on such files like add, modify, delete or even remove the entire file.

File Access Mechanisms :- File access mechanism refers to the manner in which the records of a file may be accessed. There are several ways to access files

i) Sequential access

ii) Direct/Random access

iii) Indexed sequential access

Sequential Access:- A simple access method, information in a file is accessed sequentially one record after another. To process the with record all the 1-1 records previous to 1 must be accessed. Sequential access is based on the tape model that is inherently a sequential access device. Sequential access is best suited where most of the records in a file are to be processed. For example, transaction files.

Sequential access :- A sequential access is that in which the records are accessed in some sequence i.e the information in the file is processed in order, one record after the other when stored in sequential devices like magnetic tapes. This access method is the most primitive one. Example: Compilers usually access files in this fashion.

Advantages : File design is simple location required only record key. Magnetic tapes are used to store data.

Disadvantages : Addition or deletion of data is not simple as updating require for all records to be shifted physically.

Direct Access :- Sometimes it is not necessary to process every record in a file. It may not be necessary to process records in the order in which they are present. Information present in a record of a file is to be accessed only if some key value in that record is known. In all such cases, direct access is used. Direct access is based on the disk that is a direct access device and allows random access of any file block. Since a file is a collection of physical blocks, any block and hence the records in that block are accessed. For example, master files. Databases are often of this type since they allow query processing that involves immediate access to large amounts of information. All reservation systems fall into this category. Not all operating systems support direct access files. Usually files are to be defined as sequential or direct at the time of creation and accessed accordingly later. Sequential access of a direct access file is possible but direct access of a sequential file is not.

Direct/Random access :-

Random access file organization provides, accessing the records directly with its address on the file.

Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing. They are created only on magnetic disks.

The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

Advantages :- immediate access to record is possible. Update information will be available on file. Addition or deletion is not complex.

Disadvantages :-Less efficient with respect to storage space. Less data security.

Indexed Sequential Access:- This access method is a slight modification of the direct access method. It is in fact a combination of both the sequential access as well as direct access. The main concept is to access a file direct first and then sequentially from that point onwards. This access method involves maintaining an index. The index is a pointer to a block. To access a record in a file, a direct access of the index is made. The information obtained from this access is used to access the file. For example, the direct access to a file will give the block address and within the block the record is accessed sequentially. Sometimes indexes may be big. So hierarchies of indexes are built in which one direct access of an index leads to info to access another index directly and so on till the actual file is accessed sequentially for the particular record. The main advantage in this type of access is that both direct and sequential access of files is possible.

Indexed sequential access :-

This mechanism is built up on base of sequential access and direct access file organisation.. It basically organised on the key fields where an additional index is maintained with splits of access of isolation records. An index is created for each file which contains pointers to various blocks.

Index is searched sequentially and its pointer is used to access the file directly.

Advantages :- Suitable for both sequential and online or direct access processing.

Disadvantages :-Less efficient in use of storage space. Addition and deletion of records are more complex.

CHAPTER-V (PROBLEM SOLVING METHODOLOGY)

What is a Computer : A programmable device that can store, retrieve & process data.

What is Programming :Much of human behavior & thought is characterized by logical sequences. Since infancy, we have learned how to do things, and expect certain behavior from others. A lot of what we do everyday, we do automatically Fortunately it is not necessary for us to consciously think of every step involved in a process. Say turning the pages by hand :- instructions are in sequence of 1)Lift hand 2) move hand to right side of book 3) grapes top right corner of page 4)move hand from right to left page until it is positioned so that we can read it. 5)let go o page. Think how many neurons must have fired & how many muscles must respond all in a certain order or sequence to move the arm and hand. We do it un-consciously.

Definition : It is a process of planning, scheduling & performing a task or an event.

Problem solving is a logical process of breaking down the problem into small parts which can be solved step by step to arrive at a final solution. It is a creative process and there are number of ways to get solution of the problem.

What is Computer programming : The process of planning a sequence of steps/instructions for a computer to follow.

What is Computer program : The process of planning a sequence of steps/instructions for a computer to follow. It is a set of explicit and unambiguous instructions expressed in a programming language is called a Program.

A program may also be thought of an algorithm expressed in programming language.

An algorithm is a solution of problem that is independent of any programming language.

Algorithm : A step by step procedure for solving a problem in a finite amount of time called Algorithm. A computer program and algorithm looks very similar because a program is simply implementation of an algorithm.

Properties of Algorithm :

1) Finiteness (Terminate in a finite amount of time)

2) Definiteness (It must be un-ambiguous)

3) Effectiveness (Better way of solution)

4) Input (Carried out for a given set of initial condition)

5) Output (Produce output)

How do we write program : we write sequence of instructions for a computer to follow i.e. problem solving and implementation.

1) Understand (define) the problem & what solution must do.
2) Develop general solution or algorithm or procedures or methods to follow. Algorithm is logical sequence of steps to be used to solve a problem.
3) Verify i.e. Follow the steps exactly to see if solution really does solve the problem.

i) Translate the algorithm to programming language i.e concrete solution or Program

ii) Test i.e. Have the computer follow the instruction and check the result. If error, analyze the program/algorithm to determine error & make corrections. Now use the Program.

What is programming language **:** A set of rules, symbols & special words used to construct a program.

Algorithm is defined as the step by step solution of problem in user's language.

It is considered as an effective procedure for solving a problem in finite number of steps.
The characteristics of Algorithm are

- Precise
- Unambiguous

- Finite termination
- Unique solution

Once algorithm is written, it can be coded into a program using any programming language. Algorithm uses 3 different constructs

- Sequence
- Branching or Decision making
- Repetition

Sequence says that instructions are to be executed in what order or sequence. Branching involves testing of condition and based on the outcome of the condition testing different instructions are executed. Repetition means one or more instructions shall be repeated for a number of times. This is otherwise called as loop. There are different types of loops such as

While- do , do-while , for
Example:

1. Algorithm to find out sum of two numbers to be taken as input.

Step-1 Read the 1st number x
Step-2 Read the 2nd number y
Step-3 Sum=x+y
Step-4 Print Sum

This is an example where only sequence is exhibited

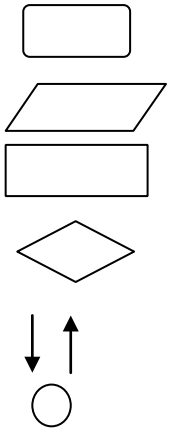
2. Algorithm to find out larger between numbers to be taken as input.

Step-1 Read the 1st number x
Step-2 Read the 2nd number y
Step-3 If x > y
Then Print x
Else if x< y
Then Print y
Else Print " Both are Equal "

This is an example where Branching is exhibited

3. Algorithm to find out sum of first 10 natural numbers.

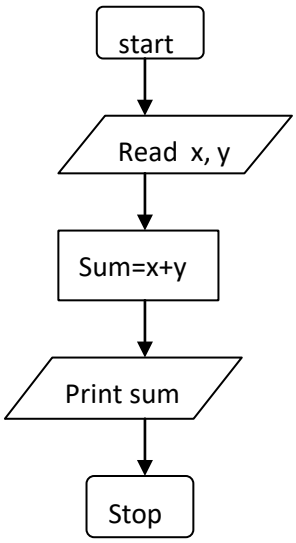
Step-1 i=1, Sum=0
Step-2 Repeat step 3 and 4 while i<>10
Step-3 Sum= Sum+i
Step-4 i=i+1
Step-5 Print Sum
This is an example where Repetition is exhibited
Flowchart is a graphical or symbolic representation of the process of solution to a problem or algorithm. It helps to visualize the complex logic of the solution of the problem in a simplified manner through diagrammatic representation. Each step of the algorithm is presented using a symbol and a short description. The different symbols used for the flowchart are



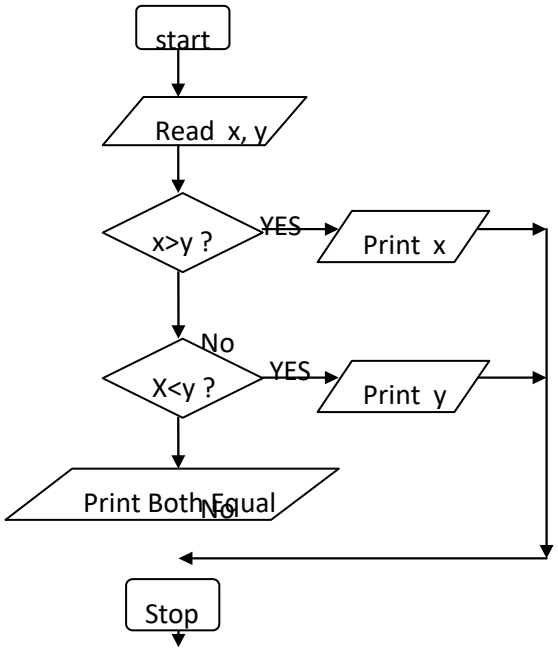
Start or Stop
Input or Output
Process
Decision Symbol
Arrow for sequence
Connector

Example

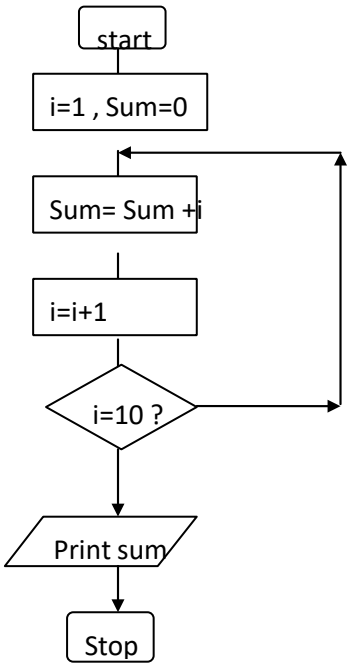
1. Flowchart to find out sum of two numbers to be taken as input



2. Flowchart to find out larger between two numbers to be taken as input



3. Flowchart to find out sum of first 10 natural numbers



Pseudocode :-

It is a concise description algorithm in English language that uses programming language constructs. It contains outlines of the program that can be easily converted to program. It focuses on the logic of the algorithm without giving stress on the syntax of programming language. This is meant for understanding the logic of the program easily. Flowchart can be considered as an alternative to pseudocode. Several constructs/key words of programming language can be used in the algorithm to write the pseudocode. Some of them are

If ... Endif
Do while ... enddo
While ... endwhile
Repeat ... until
For ... endfor
Case .... endcase
Call
Return

Programming Language

Programming language is a tool to express the logic or instructions for understanding of the computer. Any programming language has two components:

- Syntax
- Semantics

Syntax refers to the rules to be followed for writing valid program statements. Compiler can detect errors in syntax while compiling the program.

Semantics is associated with logic of the program. Compiler can not detect the semantic error. The user of programmer can diagnose semantic error.

There are good number of High level languages, each meant for specific area of data processing. Commonly known languages are BASIC,FORTRAN, COBOL, Pascal, C, C++ etc. While FORTRAN is good for Numerical and scientific calculation, COBOL is good for Business applications involving large amount of data handling.

Generations of Programming Language :-

The Programming languages can be classified into 4 generations

1st Generation: Machine Language
2nd Generation: Assembly Language
3rd Generation: High Level Language
4th Generation: Very High Level Language

Machine Level language contains instructions in binary form i.e. in 0s and 1s. Thus writing instruction was very difficult and needs heavy expertise. This was used in early days computers.

Assembly level language instructions were written using symbolic codes known as mnemonics. In comparison to Machine language it is relatively easier to write program, but still it requires lot of expertise. A translator called assembler is used to translate assembly language program to machine level language.

High level language contains instructions in English like words so that user will feel easier to formulate and write the logical statements of the program. Here the logic may spread over multiple statements as against a single statement in assembly language. It uses a translator called compiler for translation of High level language program to machine level language program. There are many High level languages used for programming such as BASIC, FORTRAN, COBOL, PASCAL, C, C++ etc.

Very High Level language other wise called as 4GL uses nonprocedural logical statements. A typical example of 4GL is the query language such as SQL.

Structured Programming Language :-

Structured Programming is also known as Modular Programming. In this type of programming technique, the program shall be broken into several modules. This helps in managing memory efficiently as the required module of the program will be loaded into the memory only and not the entire program. This will also enhance code reuse. Writing, understanding, debugging and modifying the individual module of the program is also easier.

Basic steps for problem solving :

Step1 : Read the given problem carefully & Understand the problem.

Example : Accept value m and find sum of the first m even +ve integers.

Solution : Input value of m say 5

Now to get sum of 5 even integers i.e. 2, 4, 6, 8, 10

Find the sum + 2+4+6+8+10=30. Sum of 5 even integers is 30

Step2 : Construct list of variables.

One should think in advance the number of variables along with names of variables. Variables are names in memory location. Example in the above problem , variables are

m=> which is input by user.

I=>which is to generate even integers like 2, 4, 6, 8, 10

Count=>A counter to keep track of numbers of even integers that have been summed like 1, 2, 3,4,5

Sum=> An accumulator that contains the current total value of even integers. Sum-2+4+6+8+10

A problem solver or Programmer has to decide the various kinds of variables that take different values for finding solution.

Step3 : Output design

Most of the time output design is specified in the problem. Output report must be easily understandable by the user.

Step4 : Program development

One can draw the flow chart or write the algorithm for the procedures.

CHAPTER (6 & 7) :- PROGRAMMING IN "C" LANGUAGE.

UNIT: 6. OVERVIEW OF C PROGRAMMING AND

UNIT: 7. ADVANCE FEATURES OF C

1. Introduction to C .
2. History of C .
3. Sample C Program .
4. Basic Structure of C program.
5. C - Character set .
6. Identifiers .
7. **DATA TYPES .**
8. **Keywords.**
9. **CONSTANTS** & Literals (Integer,float/real,char,string,logical) **.**
10. **VARIABLES.**
11. **Arithmatic EXPRESSION .**
12. **OPERATORS :-** Arithmatic operator,TYPE CONVERSION & TYPE CASTING, Unary operator, Relational & Logical operator, Precedence of operators, Conditional operator, Assignment Operator. Llibrary functions.
13. **Managing INPUT & OUTPUT OPERATIONS :-**

    Formatted Input & output – **scanf(),** character input & output :– **printf(),** single input & output - **getch(), putch().**
14. **DECISSION CONROL & LOOPING STATEMENT :-**

    Branching the **IF , IF-ELSE, IF-ELSE-IF**,

    Looping the **WHILE** Statement, **DO-WHILE** Statement, **FOR** loop Statement,

    the Switch Satatement, **BREAK, CONTINUE** and **GOTO** Satement.
15. **ARRAY :-**

    Defination of Array, Operation in Array, **ONE DIMENSIONAL ARRAY**, Passing array to function, **MULTIDIMENSIONAL ARRAY**, Initialisation of multidimentional Array, Arrays & Strings.
16. **FUNCTIONS :-**

    Introudction, **PASSING PARAMETERS** OR ARGUMENTS **TO FUNCTIONS (CALL BY VALUE & CALL BY REFERENCE), RECURSION,** Storage Class, automatic variable, External variable, Static variable. **STRING OPERATIONS** .
17. **POINTERS :-**

    Introduction, Passing Pointers to Function, Pointers & Arrays, Pointer expression & Pointer arithematic.

18. **STRUCTURE AND UNION :-**

Define Structure, Processing structure, Structure & Pointers, Processinga Structure, Passing structure to function, User defined data types (Type def)

1. INTRODUCTION :

We know that the program a "set of instructions" to do something and these instructions are understood by the computer. Like, we understand English, say an instruction is given as "Stand up." consists of two words to form a sentanse and we know the meaning of these two words or setense and rules/syntax for writing sentence i.e. it starts with a Capital letter and end with a period and we execute/perform the instruction. If we do not know English language, these words have no meaning to us. Similarly in order to learn Programming language, we have to learn the grammar or **syntax** of C language like its characters, numbers, words, operators and various statements and the logic or **symantic** between them & now we will study how the C programs are written.

Programming means :- preparing a set of instructions for a computer to follow. These instructions may be written in one of several **High-Level Languages**, such as FORTRAN, COBOL, BASIC, Pascal or **C** and these instructions must be translated to Machine Language because machine understand only instructions in binary form i.e. 0's and 1's. Different Languages are more suited to particular applications than others for example, **FORTRAN** has been the language of choice for the creation of programs for scientific or engineering/mathematical applications, **COBOL** is suitable for commercial or business applications, **Pascal** is a **structured programming** language user for beginners because it shows the logic of the program in structured way and **C** is also a structured programming language or procedure oriented language like Pascal which is easy to learn.

C's predecessor was called 'B' and B's predecessor was 'BCPL' (Basic Combined Programming Language).BCPL was invented in 1967 by Martin Richard at Cambridge University. Inspired by BCPL, Ken Thomson in 1970 invented Systems Programming Language called 'B' at Bell Laboratories at AT & T Lab. 'B' and 'Assembly' language were used to develop first version of Unix. **In 1972, Dennies Ritchie designed 'C'** which incorporate many ideas of 'BCPL' and 'B'. Until 1989, the definition of C was given by Brian W. Kernighan & Dennis M. Richie it was traditional C. In 1983 it becames American National Standard Institute (ANSI) C and in Dec'1989 it was standardised & now a days it comes with various versions of compilers & interpreters with little or no altrernation. There are various C, Compilers like Turbo C/C+, Code::Block GCC Compiler ,Microsoft Visual C++, Microsoft C, ANSI C etc.

C contains some additional features that **allows to be used at the lower level**, thus bridging gap between Machine Language and more conventional High level language. This is relatively good application programming /problem oriented and relatively good machine efficiency language . C is a general purpose structured programming language, the flexibility allows C to be used for system programming i.e. **writing for operating systems**.

C is charaterised by the ability to write very concise source programs, due to the large number of operators included within the language.

It has relatively small instruction set. C encourages to write additional library functions of their own. Thus features and capabilities of the language can easily be extended by the user.

C programs are highly portable, the reason is C relegates most computer dependent features to its library function. Therefore most C programs can be processed on many different computers with little or no alternation.

Some Examples where C Language is used :-

- Operating Systems, Language Compilers, Assemblers, Text Editors, Print Spoolers
- Network Drivers, Modern Programs, Data Bases, Language Interpreters, Utilities
- Knowledge of C is required for writing Object Oriented Programs like C++/Java/VC and for Drivers programs, Graphics applications for developing Video Games.
- Most OS like DOS,Windows,UNIX,LINUX are written in C.
- Database Softwares like Oracle, Foxpro are developed in C.
- Most important all programming languages has been taken from C language.
- C ffers better interaction with hardware.
- C is used for better Machine Effiency(for Assembly Language) and well as used for better Program effiency (for problem oriented HLL like BASIC,FORTRAN,PASCAL etc)

- Used for Scientists and Engineers and widely accepted by Software Industries.

2. History of C Programming Language.

C is a general-purpose high level language that was originally **developed by Dennis Ritchie in BELL Laboratories in 1972.** The Unix Operating System and virtually all Unix applications are written in the C language. C has now become a widely used professional language for various reasons.

- Easy to learn
- Structured language
- It produces efficient programs.
- It can handle low-level activities.
- It can be compiled on a variety of computers.

Facts about "C" :-

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around 1970
- The language was formalized in 1988 by the American National Standard Institue (ANSI).
- By 1973 UNIX OS almost totally written in C.
- Today C is the most widely used System Programming Language.
- Most of the state of the art software have been implemented using C

Importance of "C" :-

- It is well suited for Structure Programming i.e. requiring user to think of problem in terms of functions, modules or blocks. Modular structure makes programme debugging, testing & maintenance more easier.

- "C" contains soma additional features that allows to be used at the lower level i.e. permitting close interaction with the computer's inner workings which is different from another high level languages. Thus bridging gap between the machine language & more conventional high level language reduced.

- "C" is an middle level language which combines the features of High Level Language (HLL) and Assemble Language. Thus it is suitable for Application Software and System Software development. This flexibility allows for developing to Operating Systems.

- "C" is cauterized by ability to write very concise and efficient source programs due to the powerful operator included within the language.

- "C" has relatively small character set. i.e. only have 32 keywords.
- "C" encourages to write additional library functions of their own and it has ability to extend (Libraries can be extended by the users).

- "C" is highly portable. Programs written in one computer can be run in another computer without modification.

- "C" programs are efficient due to variety of data-types such as array, pointer, function, structure, union & registers etc. are available.

- It is many times faster than programming languages like BASIC. For example to increment a variable from 0 to 15000 takes 1 second in C, where as in BASIC takes 50 seconds.

- C is function oriented. All additional tasks including input & output, graphics, math computations & access to peripherial devices are placed as library of functions.

- C has a rich set of Operators.
- Dynamic storage location is possible in C
- C can easily manipulate with bits, bytes and address.
- It is applied in system programming like OS, assemblers, compilers, interpreters, database, utilites etc.

3. Sample 'C' Program (C Program basically has the following form) :-
   - Preprocessor Commands
   - Functions
   - Variables
   - Statements & Expressions
   - Comments

The following program is written in the C programming language. Open a text file ***sample.c*** using editor and put the following lines inside that file.

```
#include <stdio.h>
#include<conio.h>
int main()
    {
     /*  My first program  is a comment statement */
     printf("Hello, Welcome to Government Polytechnic, Boudh ! \n");
     return 0;
    }
```

Preprocessor Commands: These commands tells the Compiler to do preprocessing before doing actual compilation. Like ***#include <stdio.h>*** and **#include<conio.h>** are preprocessor commands which **tells a C Compiler to include stdio.h and stdio.h header file before going to actual compilation.**

Functions are main building blocks of any C Program. Every C Program will have atleast one or more functions and there is **one mandatory function** which is called ***main()*** **function**. This function is prefixed with keyword *int* which means this function returns an integer value when it exits. This integer value is retured using ***return*** **statement**. The first line of the program informs the system that the name of the program is main where the program is to begin execution.

The open and closed parenthesis **( )** immediately following main specify that no "**arguments**" **or "parameters"** are expected by this routine or function named main. Thus, we have told the system that the name of our program is main and we are ready to specify precisely what function this program/routine is to do/function. This is done by enclosing all program statements of the routine within a pair of braces. All the statements are enclosed within curly braces **{ }.**

In the above program, only one statement in which a routine named printf is to be invoked or called.

The parameter or argument to be passed or handed to the printf routine is the string of characters.

The **printf routine is a built-in or library function** in the C system that simply prints or displays its arguments at the terminal or on the video screen. The last two characters in the string, namely the blackslash(\) and the letter **n**, are known collectively as the **newline character** i.e. goes to a new line. Any characters to be printed after the new line character will then appear on the next line of the terminal or display.

All program statements in C must be terminated by a semicolon ; .

We can just modify the above Program.

The C Programming Language provides a set of **built-in functions**. In the above example ***printf()*****is a C built-in function** which is used to print some information on the screen.
We can write our own functions and use them.
Variables: are used to hold numbers, strings and complex data for manipulation. //Here no variab;les used.
Statements & Expressions:- Expressions combine variables & constants to create new values. Statements are expressions, assignments, function calls, or control flow statements which make up C programs.
Comments statements :- are used to give additional useful information inside a C Program. All the comments will be put inside **/* ..... */** as given in the example above. A comment can span through multiple lines.

Note the followings :-

(i)C is a case sensitive programming language. It means in C *printf* and *Printf* will have different meanings. (ii) C has a free-form line structure. End of each C statement must be marked with a semicolon (**;**) . (iii) Multiple statements can be one the same line. (iv)White Spaces (ie tab space and space bar) are ignored. Statements can continue over multiple lines.

The best way to learn C or any programming language is to begin writing programs in it; so here's our first C program:

In C programming language, upper case letters and lower case letters are distinct.

C does not have any restricttion to start program statements any where in the line, unlike other programming languages. **Only thing you can write programs so that it is easy to read.**

After Source program is written by the editor of the Compiler itself it is saved with a file name then compiled by the Compiler and if no syntax error, then it creats an an Object program, which is linked with the Library and then an Executable program is created. We can get the output by running the Executable file the same source file with extension .EXE. In order to run this EXE file no need of compiler is required .

After compiling the above program, we can get the output on the terminal as :-

/* Examples.1 Print “Hello, Welcome to Government Polytechnic, Boudh” */

```
#include <stdio.h>
#include<conio.h>
main()
    {
      printf(“Hello, Welcome to Government Polytechnic,Boudh\n”);
    }
```

The output will appear like

Hello, Welcome to Government Polytechnic, Boudh

*/*Example.2.Program* to add two integer values and displays the value*/

```
#include <stdio.h>
#include<conio.h>
main()
    {
        /* Declares variables*/
        int val1,val2,sum;
```

```
    /* Assigns value */
    val1=50;
     val2=60;
     sum=val1+val2;
     /* Displays result*/
     printf("The Sum of %d and %d is %d\n",val1,val2,sum);
}
```

Output :- The Sum of 50 and 60 is 110

A Comments Statement is initiated by the **two characters / and *** . These two characters must be written without any intervening spaces. It is a good idea to insert comment statements as it is easier to document the program while particular program logic is still fresh in one's mind than it is to go back and rethink the logic after the program has been completed. Secondly it is easier during debuging .

Next statement declares variable val1,val2,sum to be type integer. The declaration of variable specifies to C compiler how a particular variable will be used by the program and here all the variables are having integral values i.w. without any decimal places or floating point numbers.

50 is assigned with Assignment operator (=) to variable val1

60 is assigned with Assignment operator (=) to variable val2

sum is computed adding the variable val1 and val2

The percent sign (%) within the the prints function indicate what type of value to be display at this point. The letter "d" is recognized by the system as signifying that as integer value to be displayed. Whenever printf routine finds the %d character inside the character string it will automatically display the value of the next argument to the printf routine.

/* Example :- 3 */

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
void main()  {
                int k ;
                /* A number 99  is assigned to the variable k */
                k=99;
                printf("\n %d \n",k);
```

```
}
```

There are several include statements in the beginning. **#include <stdio.h> is a pre-processor directive(Compiler Directive)** and requests some action before the program is actually translated into machine code. The pre-processor directives always begins with **#(pound) sign**. These are to include C-library in the Program so that we can make calls to the routine C-functions. It can prevent lot of time consuming errors. It is advisable to write the include statements at the beginning. Include clause include header file **stdio.h** provides proper interface to the library output function printf. Similarly other header files have special purpose.

Down # the statements, there is a main statement. This is the main procedure of the program. The curly brace '**{**' above and the curly brace '**}**' below define the statement block they makes the **main** procedure of the program. The **int** statement defines the variable k to be an integer.

The statement **/*** A number 99 is assigned to the variable k. /* and ***/** is a comment statement. It begins with **/*** and ends with ***/**. In any programming language it is a good idea to include comments to read your program and others can also read program. The contents are ignored by the Compiler.

Then we print the value k with printf statement. This is the printf procedure. It is a built-in function with C language that allows to print the values of variables on the screen. With printf function there is a string and in this case we have **/n %d \n** and next string contains formatting functions. **/n** are used for new line and **%d** we are going to afford the value of the integer variable. k is the integer variable we are going to display the value of the variable on the screen. **No space is allowed between \ and n.**

Now to run the program, we can get the value on the screen by ctrl and f5 key. If we look at the screen, we will see value 99 printed out. This is a result of printf function.

Let's take another example :-

```
/* Example :- 4 Program by Asim Barla  */
#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <math.h>

#include <string.h>

#include <time.h>

void main() {
        int j ;
        j=1;
        while (j<=20)
                {
                  printf("%i \n",j);
                  j=j+1;
                }
        }
```

In the above program again there are several include statements. It is good itea to include these statements at the beginning.

This program prints the value from 1 to 20 is something very simple but gives an idea of using while statement in C.

Down the main procedure, we have two curly brackets which include main procedure. We indent two spaces just for readability. In C we can write any where and any line but it is good idea to have spacing conventions. It will be easy to read and easy to write.

In the above program, j is declared to be an integer and then j is assigned to have value 1. Then we see while statement which performs following statements when statement block is executed when statement j<=20 is executed. Statement block within the curly braces, the printf statement & j=j+1 is repeatedly executeds until j<=20. First time in the loop j=1 and j will be printed then j is incremented by 1 i.e. j = 2 and it will be printed and this way when j=11 loop stops.

Output :-

1 2 3 4 5 6 7 8 9 10

In C, any thing is case sensitive and hence upper and lower case are different.

4. Basic Structure of C program

Documentation Section

Link Section	**Pre Processor Directive** :
Definition Section Global Declaration Section	(AT the time of Compilation itself some processes can be done in C. There are 3 types of preprocessor **#include <stdio.h>** (inclusion of header file) #define Macro substitution #if directives

main() **[Function Section]**

{

declaration part Declares all variables to be used in executable part ,

executable part Must have atleast one statement.

} logical end of program.

subprogram Section

func1()

func2()

func()

- Every C Program consists of one or more functions, one of which must be called as main().
- The Program execution starts from this function main.
- The statement within the function are always enclosed within a pair of braces { }
- The group of statements within main() are executed sequentially.
- The main() function can be located anywhere in the Program, but in general practice is to place as the first function for better readability.

Each function must contain :

- The Program is written in lower case only. (Upper case letter for symbolic constants)
- Lower case and upper case letter are not same in C.
- C has no specific rules about the position at which different parts of the statement are to be written and each expression statement must end with semicolon **;**

- Comment statements may appear any where in the program as long as placed within delimiter /* & */
- The Pre-processor directive request some action before the program is actually translated into machine code. The Preprocessor directive always begins with pound sign #. The #include preprocessor causes the contents of the named file stdio.h. Such file is called a header file. All the header files are declared first before main() function.
- A function heading which consists of function name followed by an optional list of arguments enclosed in paranthesis.
- A list of arguments declarations if arguments are included in the heading.
- Statements are written in separate lines (free from language)
- Proper indentation (readability)

5. C - CHARACTER SET.

As we know, the English is a Laguage for communication and to learn English we must klnow what is alphabets, words, sentences and paragraphs. English has alphabets and digits which makes words or numbers and forms a setenses and paragraphs to express some meaningful information.

Similarly, C Language has alphabets, digits, special symbols which makes them as constants, variables, keywords and forms statements or instructions and finally convert it to a program.

The characters are building blocks to form basic program statement.

1. Letters :(Alphabets) A,B,C,D…...Z, a,b,c,d…..z.
2. Digits : 0,1,2,3....9
3. Special symbols : ,.;:?'"!/_#$%^&*()@ {} []` etc
4. White spaces : The complier ignores white spaces unless they are part of the sring constant.

   Blank spaces,horizontal tab,carriage return,new line,form feed.

C uses certain combination of these characters, such as \b, \n and \t to represent special condition such as back space, newline & horizontal tab respectively. These character combinations are known as escape sequences. (represent single character).

TOKENS :- Smallest element of the program are called as Tokon. C-Tokens are of six types :-

1) Keyword 2) Constant 3) Variables 4) Identifier 5) Specifier 6) Operator

6. CONSTANTS.

A constant is a quantity that does not change. The quality can be stored at a location in the memory of the Computer. C has 5 types of constants. They are integer constant, floating-point (real) constant, character constant, string constant & logical constant.

CONSTANT			
Numeric constant		Character constant	
Integer constant	Real constant/	Single character Const	String constant

	Floating point constant		

Integer & Real/Floating Point Constants are called numeric type constants. Following rule can be applied to all numeric constant.

- Commas & blank spaces can not be included within a constant.
- It can be preceded by – sign if desired.
- Values of constant can not be exceeded maximum & minimum limit allowed by the compiler.

(i) **Integer Constant** :- Integer constant consists of only absolute number either in the positive or negative. Example :- 25, 55, -23, -67.

Integer Constants are of 3 types :- They are Decimal, Octal, Hexadecimal.

An integer constant must have atleast one digit. This can be written in three different number system i.e. (a) Decimal [base 10] (b) Octal [base 8] (c) Hexadecimal [base16]. Integer (decimal) constant can consists of digits taken from the set [0-9] if two gigits then first digit must be other than zero. It must not have decimal point and could be either +ve or –ve, if no sign then +ve and no commas or blank spaces are allowed within it. The number could be -32768 to +32767 within allowable range of compiler.

Example of some valid Integer Constants are :-

0, 426, +782, -8000, 32767, -7605.

Short integer (16 bit) , occupy two bytes memory space.

Long integer (32 bit) , occupy 4 bytes memory space.

Long Constants are written with a trailing L for Example :- 890L.

Octal integer constant consists of any combination of digits taken from the set 0 to 7 with leading 0. However, first digit must be 0 (zero). **Octal constants are written with a leading zero - 015.**

Example : 0, 01, 073, 0777

Hexadecimal integer constant must begin with combination of digits taken from the set 0 to 9 and a through f either upper or lower case and must be preceded by 0x.

Hexadecimal constants are written with a leading 0x - 0x1ae.

Example : qx, qx1, qxabcd.

(ii) **Real Constant** :- Real constant also called the floating point constant and consist absolute number including the real number. Example :- -27, 97, 19.245.

Real constant in a base 10 number can be written in two forms i.e. either decimal points (fractional form) and/or an exponent form.

Example : 1, 0.2, 827.021, 5000

In exponent form, the representation is in two parts. The part appearing before ‘e’ is called exponent. Mentisa part may have a +ve or –ve sign and default is +ve & similar like exponent. Exponent can not be real number or fractional form.

Example : +3.3e-5, 4.1e8, -0.2e+3, -3.2e-5.

Floating point with single precission takes four bytes & Floating point with double precission takes 8 bytes of memory space.

N.B. : It is understood that integer constants are exact quanties where as floating point constants are approximation for example floating point constant 1.0 might be represented in the computer memory as 0.999999… even though it appears as 1.0, when it is displayed because of automatic rounding. Therefore. Floating point values can not be used for certain purposes such as :- counting, indexing etc where exact values are required.

(iii) **Character Constant** :- Character constant is enclosed within single quote. Example :- 'b', 'X'

Two types :- (a) Single Character Constant (b) String Constant.

A character constant is a single character enclosed in apostrophes (') – single quotation mark. Character constants have integer values that are determined by the computer's particular character set. Single character constant can be maxium one character. Most computer uses ASCII character set :-

Example of Character constant : 'A' 'X' '3' '$' ' '

Values by ASCII character set : 65 120 51 31 32

Example : printf("%d",'a'); output will be 97 the ASCII value of letter a.

Escape sequence :-

Certain non-printing characters as well as **"**(Double quote), **'**(Apostope), ?(Question mark) and the **** (Backslash) can be expressed in escape sequence.

As escape sequence always begins with a backward slash followed by one or more special characters.

Escape sequence always represents a single character.

Character	Escape sequence	ASCII value
bell (alert)	\a	007
backspace	\b	008
horinzental tab	\t	009
vertical tab	\v	011
new line (line feed)	\n	010
form feed	\f	012
carriage return	\r	013
quotation mark (")	\"	034
apostrophe (') i.e. single quote	\?	039
quotation mark (?)	\?	063
Back slash \	\\	092

ASCII Null character(end of string) used automatically to terminate character string	\0	000

Example:- '\n', '\t', '\b'

Character constant takes one byte memory space.

(iv) **String Constant** :- String constant is a group of character and enclosed wuithin double quotes. Example :- "Good", "Welcome".

String Constant consists of any number of consecutive characters (including none) enclosed within a pair of double quotation marks.

Example:- "green" "$19.95" " " Empty string or Null string.

"line1\n ine2\n ine3\n"

Output will be line1

line2

line3

Sometimes \ & " mark must be included as a part of the string constant. These characters must be represented in their escape sequence. Similarly certain non-printing characters e.g. tab, newline can be included in string constant if they are in terms of their escape sequence.

A singlr character string constant does not have an equivalent inter value, while a character constant has an integer value.

N. B. : 'A' & "A" are not same as 'A' is character constant and "A" is string constant.

Question 1: Give the output of the following program:

```
#include <stdio.h
main ( )
        {
            printf ("This is the first line of outputs");
            printf ("But is this the second\nline of output?"):
         }
```

Question 2: Execute Program and obtain the answer to the question :

```
main ( )
        {
            printf("In how many lines will the output\
            of this program be printed?");
         }
```

Question 3: Give the output of the following programs:

```
#include <stdio.h
    main ( )
    {
      printf ("\"A \\", the teacher said, \"is used to");
      printf (" insert a new line in a C string.\"");
```

```
  printf ("\n\"I C, I C\", said the blind student.\n");
}
```

7. IDENTIFIERS :

In C, word is classified as either keyword or identifier. **Identifiers are names given to various program elements, such as variables, functions and arrays etc. Identifiers are all user defined.**

VARIABLE is a name in the memory location for storing program computation & results i.e. variable in C is to request storage for particular data type & to give storage name. A variable name can be chosen by the Programmer in meaningful way so as to reflect the type of value that is to be stored in that variable. Declaration of variable does the followings:-

It tell the Compiler what the variable name is.

It specifies what type of data the variable will hold.

The rules for forming Variable name is :-

It must consists of letters (A through Z or a through z) digits (0 through 9) & underscore charachers (_). It begins with a character upper & lower case are significant. It can begin with a letter or underscore (_) and may be followed by any combination of letters (upper or lowercase), underscore or the digits 0-9.

Before using the variables in the program they must be properly Declared. Variables can be integer, floating point numbers, characters even pointers which points to memery location of computer.

There is no restriction on the total number of characters that can be used for variable name, on some system only first eight characters are significant and some distinguishes name upto 31 characters long (ANSI C).

The following is a list of valid variable names :-

Example : A96, _sum, total_cars

The following variables are not valid :-

Sum$value > dollar sign is not valid

Price list > spaces are not permited with variable name

5set > does not begin with a letter or underscrore

int > it has a special meaning to the compiler

It must not be a keyword. A keyword is a word such as int or while that has a special meaning to the compiler. And must be written in lower case letters. There are **32 keywords**. It can be arbitrarily long. Some compilers recognizes **first 31 characters, white space not allowed.**

8. DATATYPES :

C has concept of 'data types' which are used to define before its use. The definition or declaration of a variable will assign storage for the variable and define the type of data that will be held in the location. Variable are names in the memory location which values can be changed any time.

C supports different types of data each of which may be represented differently within the Compiler's memory. Basically categorized in to four types :-

1. Primary or Primitive or Fundamental Data Types :- Char,Int, Float, Double, Void
2. Derived Data Types :- Arrays, Functions,Pointer
3. User Defined Data Types :- (Structure, Union, Enum)
4. Empty Data Type :- Void

The primary or basic data types are :- int, float, double and char. Some basic data types can be augmented by using data type **qualifiers or modifiers like short, long, signed, unsighned etc.**

The qualifier or modifiers define the amount of storage allocated to the variable.
The amount of storage allocated is not cast in stone. ANSI has the following rules:
short int , long int,float , double, long double

Example : char, int or short int (same meaning) long int, unsigned int, unsigned short int, float, double.

The above data types are called primary data types & several data types can be derived called secondary data types.

C - DATA TYPES		
PRIMARY DATA TYPES		SECONDARY DATA TYPES
Char		Array
Int		Pointer
Float		Structure
Double		Union
Void		

The basic data types and typical memory requirement are :-

Data types	Description/meaning	Memory requirement or size (bytes)	Range of values	Format String
Int	Integer	2 bytes (1 word)	-32768 to +32767	"%d"
Long int		4 bytes	0 to +4,294,967,295	"%ld"
Char	Single character	1 byte	-128 to 127	"%c"
Float	Single precision, Floating point number contains decimal point or exponent	4 bytes	3.4e-38 to 3.43+38	"%f"
Double	Double precission	8 bytes	1.7e-308 to 1.73+308	"%d"
Void	Value less	0 bytes		"

Declaration of variables are done before they are used in the Program.

Declarative statements are firat statements in all C program.

Syntax :- Datatype variables1, variable2, …, variablen**;**

Example :- int count; int number, total; double ratio; float amount;

Assigning values to variables.

Values can be assigned to variables using the assignment operator = sign.

Syntax :- Variablename = Expression or variables or constant

Example :- c=a+b; p=q; x=200;

Initialization :- It is also possible to assign values to a variable while declaring variables.

The process of giving initial values to variables is called initialization.

Syntax :- Datatype variablename=constant;

Example:- int final_value=100; double balance=75.87; char yes='x';

C permits the initialization of more than one variable in one statement using multiple assignment operator.

Example :- p=q=r=0; x=y=z=max;

Integer constant consist of a sequence of one or more digits. It can negetaive value with a minus sign.

int is used to define integer numbers

```
{
        int Count;
        Count = 5;
}
```

Floating point constant is distinguished by the presence of decimal point. To display a floating point value at the terminal, the printf conversion characters %f is used. Floating point conversion can also be expressed in scientific-notation. The value $1.9e^5$ is a floating point value expressed in this notation represents the value $1.9X10^5$.

float is used to define floating point numbers.

```
{
        float Miles;
        Miles = 5.6;
}
```

Double type is same as type float but only when the accuracy provided by float is not sufficient double can store roughly twice as many as significient digits as float. To display a double value at the terminal , the format %f or %e is used.

double is used to define BIG floating point numbers. It reserves twice the storage for the number. On PCs this is likely to be 8 bytes.

```
{
        double Atoms;
        Atoms = 2500000;
}
```

Char data type can store a single character such as 'a', '5' etc.. A character constant is formed by enclosing the character within a pair of single quote marks. The format characters %c can be used in a printf call to display the value of char variable at the terminal.

char defines characters

```
{
        char Letter;
        Letter = 'x';
}
```

We can find out how much storage is allocated to a data type by using the **sizeof** operator. Here is an example to check size of memory taken by various datatypes.

```
int main()
{
  printf("sizeof(char) == %d\n", sizeof(char));
  printf("sizeof(short) == %d\n", sizeof(short));
  printf("sizeof(int) == %d\n", sizeof(int));
  printf("sizeof(long) == %d\n", sizeof(long));
  printf("sizeof(float) == %d\n", sizeof(float));
  printf("sizeof(double) == %d\n", sizeof(double));
  printf("sizeof(long double) == %d\n", sizeof(long double));
  printf("sizeof(long long) == %d\n", sizeof(long long));
  return 0;
}
```

```
main()
        {
         int integer_variable = 100;
         float floating_var = 331.79;
         double double_var = 8.44e+11;
         char char_var = 'W';
          printf("integer_var=%d\n", integer_var);
          printf("floating_var=%f\n",floating_var);
          printf("double_var=%e\n",double_var);
          printf(char_var=%c\n",char_var);
          }
```

Output

```
integer_var=100
floating_var=331.789978
double_var=8.440000E+11
char_var= W
```

Global Variable & Local Variable :-

Global variable is defined at the top of the program file and it can be visible and modified by any function that may reference it.Global variables are initalised automatically by the system when you define them!

Data Type	Initialiser
int	0
char	'\o'
float	0
pointer	NULL

If same variable name is being used for global and local variable then local variable takes preference in its scope. But it is not a good practice to use global variables and local variables with the same name.

```
   int i=4;        /* Global definition   */
  main()
    {
       i++;        /* Global variable     */
       func();
```

```
    printf( "Value of i = %d -- main function\n", i );
  }
func()
{
   int i=10;     /* Local definition */
   i++;          /* Local variable    */
   printf( "Value of i = %d -- func() function\n", i );
}
```

This will produce following result

```
        Value of i = 11 -- func() function
        Value of  i = 5 -- main function
```

i in **main** function is global and will be incremented to 5. **i** in **func** is internal and will be incremented to 11. When control returns to **main** the internal variable will die and and any reference to **i** will be to the global.

9. KEYWORDS :-

Keywords are the words whose meaning has already been defined C Compiler and they can not be re-defined to mean anything. All the keywords have fixed meaning and these meaning can not be changed. The keywords are also called reserve words. The keywords can not be used as identifiers or variable names etc. The keywords can be used only for their intended purpose. Keywords must be written in lower case.

Example : **There are only 32 keywords in C.**

a	auto	Define local variable as having local lifetime
b	break	Passes control out of compountr statement
c	case	Branch control
	char	Basic data type
	const	Makes variable value unmodifiable
	continue	Passes control to the beginning of the loop
d	do	Do-while loop
	double	Floating point data type
	default	Branch control
e	enum	Defines a set of constants of type int
	extern	Indicate that an indentifier is defiened else where
	else	Conditional statement
f	for	For loop
	float	Floaining point data type
g	goto	Unconditional transfer control
i	int	Basic data type integer
	If	Conditional statement
l	long	Type modifier
r	register	Tell the compiler to store the variable being declared on CPU register
	return	Exits the function
s	short	Type modifier
	signed	Type modifier
	sizeof	Return the size of tge variable or type
	struct	Groups variable into a single record
	switch	Branches control
	static	Preserves variable value to servive after its scope ends
t	typedef	Creats a new type
u	union	Groups variable which share the same storage space
	unsigned	Type modifier
v	void	Empty data type

	volatile	Indicates that a variable can be changed by a background routine
w	while	Repeats exwecution while the condition is true

Keywords serves as basic building block for program statements.

10. C - OPERATORS :-

What is Operator? An operator is a symbol that specifies the arithmetic, logical or relational operation to be performed. Operators are used in program to manipulate data and variables. Variables and constants are separated by operator, It is used to perform mathematical operation or when we want to separate variables and constants, we use operator.
Example:- x=50; Here x is a variable and 10 is constant and are separated by = sign. The sign (=) is an operator.

Operand : Variables are called as operand.
Example :- x+y Here x and y are operands and + is operator
C language supports 45 number of operators. All operators are classified into 3 categories:- OPERATOR:- 1) UNARY OPERATOR ii) BINARY OPERATOR III) TERNARY OPERATOR.

1) Uniary Operator :- These operators act upon a singlw operand to produce a new value.
Example :- x++;

2) Binary Operator :- These operators act upon two operands to produce a new value.
Example :- x=x+1;

3) Ternary Operator :- These operators act upobn three operators to produce a new value.
Example :- x = 10 > 5 ? 1 : 0;

Rules :-All operators follow the following two rules, such as
Precedence :- It decides which operations would be performed first in an expression.
Associativity :- It decides the order of evaluation if more than one operator enjoys the same precedence. There are three different types of associativity , such as

a) Left to right b) Right to lefty c) Right left.Left right

List of operators, Precedence, associativity and functionality :-

Category	Operator	Associativity
Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - - (type) * & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<< >>	Left to right
Relational	< <= > >=	Left to right
Equality	== !=	Left to right

Bitwise AND	&	Left to right
Bitwise XOR	^	Left to right
Bitwise OR	|	Left to right
Logical AND	&&	Left to right
Logical OR	||	Left to right
Conditional	?:	Right to left
Assignment	= += -= *= /= %= >>= <<= &= ^= |=	Right to left
Comma	,	Left to right

Other ways for different classification of operators, such as -

1) Arithmetic Operators 2) Logical (or Relational) Operators 3)Bitwise Operators 4) Assignment Operators 5) Misc Operators

1) Arithmetic Operators:-

These operators are used to perform the mathematical expression :-
Suppose variable A holds 10 and variable B holds 20 then:
Example: -

Operator(Meaning)	Description	Example
+ (Addition)	Adds two operands	A + B will give 30
- (Subtraction)	Subtracts second operand from the first	A - B will give -10
* (Multiploication)	Multiply both operands	A * B will give 200
/ (Division)	Divide numerator by denumerator	B / A will give 2
% (Modulus operator to get remainder in integer division)	Modulus Operator and remainder of after an integer division	B % A will give 0

2) Assignment Operators:-

Assignment operstors are used to assign the result of an expression to variable. The most commonly used assigbnment operatyor is = sign. Syntax : identifier = expression;

Where identifier generally represents a variable and expression reprersents a constant, a variable or a more complex expression.

Example :- a=3; x=y; delta=0.001; sum=a+b; area=length * breadth;

Operator	Description	Example
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B will assigne value of A + B into C
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	C += A is equivalent to C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	C -= A is equivalent to C = C - A
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	C *= A is equivalent to C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	C /= A is equivalent to C = C / A
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	C %= A is equivalent to C = C % A
<<=	Left shift AND assignment operator	C <<= 2 is same as C = C << 2
>>=	Right shift AND assignment operator	C >>= 2 is same as C = C >> 2
&=	Bitwise AND assignment operator	C &= 2 is same as C = C & 2
^=	bitwise exclusive OR and assignment operator	C ^= 2 is same as C = C ^ 2
|=	bitwise inclusive OR and assignment operator	C |= 2 is same as C = C | 2

3) Relational and Logical Operators:-

The Relational operators are used to compare the value of operands. Logical operators are joining more then two logical conditions. Suppose we assume variable A as 10 and variable B as 20 then:- Examples:-

Operator	Description	Example/Relational or Logical	Meaning
==	Checks if the value of two operands is equal or not, if yes then condition becomes true.	(A == B) is not true.	Equal to
!=	Checks if the value of two operands is equal or not, if values are not equal then condition becomes true.	(A != B) is true.	Not eqal to
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(A > B) is not true.	Greater than
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(A < B) is true.	Less than
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(A >= B) is not true.	Greater than equal to
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(A <= B) is true.	Less than equal to
&&	Called Logical AND operator. If both the operands are non zero then then condition becomes true.	(A && B) is true. (Logical operator)	Logical And
||	Called Logical OR Operator. If any of the two operands is non zero then then condition becomes true.	(A || B) is true. (Logical operator)	Logical OR
!	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is false. (Logical operator)	Logical NOT

5) Bitwise Operators:-

The operators provided for bit manipulation.
Assume if A = 60; and B = 13; Now in binary format they will be as follows:
A = 0011 1100
B = 0000 1101

A&B = 0000 1100
A|B = 0011 1101
A^B = 0011 0001
~A = 1100 0011
Example : Following Bitwise operators supported by C language

Operator	Description	Example
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) will give 12 which is 0000 1100
|	Binary OR Operator copies a bit if it exists in eather operand.	(A | B) will give 61 which is 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) will give 49 which is 0011 0001
~	Binary Ones Complement Operator is unary and has the efect of 'flipping' bits.	(~A) will give -60 which is 1100 0011
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 will give 240 which is 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 will give 15 which is 0000 1111

Example :-

```
main() { char x=10;
        X=x>>1;
    printf("%d"x);
    }  Output is 5

main() { char x=10;
        X=x<<1;
    printf("%d"x);
    }  Output is 20
```

Misc Operators :-

There are few other operators supported by C Language.

Operator	Description	Example
sizeof()	Returns the size of an variable.	sizeof(a), where a is interger, will return 4.
&	Returns the address of an variable.	&a; will give actaul address of the variable.
*	Pointer to a variable.	*a; will pointer to a variable.
? :	Conditional Expression	If Condition is true ? Then value X : Otherwise value Y

Operators Categories :-

All the operators we have discussed above can be categorised into following categories:

- Postfix operators, which follow a single operand.
- Unary prefix operators, which precede a single operand.
- Binary operators, which take two operands and perform a variety of arithmetic and logical operations.
- The conditional operator (a ternary operator), which takes three operands and evaluates either the second or third expression, depending on the evaluation of the first expression.
- Assignment operators, which assign a value to a variable.
- The comma operator, which guarantees left-to-right evaluation of comma-separated expressions.

Precedence of C Operators :-

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator:
For example x = 7 + 3 * 2; Here x is assigned 13, not 20 because operator * has higher precedenace than + so it first get multiplied with 3*2 and then adds into 7.
Here operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedenace operators will be evaluated first.

Expression :-

An expression in C consists of a syntactically valid combination of operators and operands, that computes to a value. An expression by itself is not a statement. A statement is terminated by a semicolon; an expression is not. Expressions may be thought of as the constituent elements of a statement, the "building blocks" from which statements may be constructed. The important thing to note, is that every C expression has a value.

Example of arithmatix expression :-

```
main()
        {
          int a =100;
         int b =2;
        int c=25;
        int d=4;
        int result;
        result =a-b;
        printf("a-b = %d\n",result);
        result=b*c;
        printf("b*c =%d\n",result);
        result-=a/c;
        printf("a/c=%d\n",result);
        result=a+b*c;
        printf("a+b*c=%d\n",result);
        }
```

Output :-

```
a-b=98
b*c=50
```

a/c=4 a+b*c=150 a*b+c*d=300

Example of modulus operator (%) :-

```
main()
        {
        int a=25,b=5,c=10,d=7;
        printf("a %% b = %d\n",a%b);
        printf("a %% c = %d\n,a%c);
        printf("a %% d=%d\n",a%d);
        printf("a / d * d +a %% d = %d\n",a/d*d+a%d);
        }
```

Output :-

```
        a % b = 0
        a % c = 5
        a % d = 4
        a / d * d + a % d = 25
```

Example of integer and floating point conversion :-

```
main()
            {
            float f1=123.125,f2;
            int i1,i2=-150;
            char c='a';
            i1=f1;        /* floating to integer conversion */
            printf("%f assigned to an int produces %d\n",f1,i1);
            f1=i2;        /* integer to floating conversion*/
            printf("%d assigned to a float produces %f/n",i2,f1);
            f2=i2/100.0;          /* integer divided by float */
            printf("%d divided by 100.0 produces %f\n",i2,f2);
            }
```

Output :-

123.125000 assigned to an int produces 123

-150 assigned to a float produces -150.00000

-150 divided by 100 produces -1.00000

-150 divided by 100.0 procuces -1.500000

13. INPUT OUTPUT FUNCTIONS :-

In C, Input operations are carried out through its built-in functions. The functions which are used of I/O operation are called standard I/O library. Each program that uses I/O functions must contain statement **#include<stdio.h>** header file at the beginning. **stdio.h** is an abbreviation of standard input output header file. Followings are standard input output library functions.
Character oriented input-output functions :- getchar() , putchar()
String oriented input-output function :- gets() , puts()
Formatted input-output funsctions :- scanf() , printf()

getchar() :-
The getchar() function reads a character from the keyboard.
The getchar() function waits until a key is pressed and then returns a value.
The key pressed is also automaticallty echoed to the screen
Syntax :- variable=getchar();
Example :- x=getchar();

putchar() :-
The putcahr() function writes a character to the screen at the current cursor position.
Syntax :- putchar(variable);
Example :- void main() { char x; x=getchar(x); putcahr(x); }

gerts() :-
The gets() function reads a string odf cahracters entered at the keyboard and places them at the address pointed to by its argument. We may type the characters at the keyboard until we press ENTER eky.
Syntax :- gets(variable);
Example :- gets(x);

puts() :-
The puts() function writes its string argument to the screen followed by a newline.
Syntax :- puts(variable);
Example :- vaoid main() { char x[20]; gets(x); puts(x); }

scanf() :-
The scatf() function is an input function, used to read multiple types of data values. This function reads integer, real, character and strings from the standard input (leyboard).
Syntax :- scanf("format string", argument list);
The "format string" contains the "format codes" and determines the number of arguments that follow it. The format string should be given within double quotes. The argument list contains list of variables. An ampersand symbol "&" should be preceed each numeric and character type variables by which addresses of the variable are denoted. In the case of string variable, the ampersand sign "&" should not be given. Each format code starts with the character % sign. The inpout from the keyboard should be given in either of the folllwing two ways:-
Value of each argument separated from the previous one by a space.
Value of each argument should be ended by pressing the ENTER Key i.e. each argument in a separate line.
Example :- scanf("%d",&x); scanf("%d%f%c",&basic,&da,&gender);

printf() :-
The printf() functiobn is used to diosplay values or result at the terminal. This function can be used to output any combination of numerical values, single characters and strings.
Syntax :- printf("format string", argument list);
The fiormat string contains :- characters that are simply printed as they are, conversion specification that begins with %sign, escape sequences that begin with a \sign.
The argument list may contain constants, variables, expressions or even functions. The following is the list of conversion characters that can eb used wiyjh the print() function.
Example :- printf("%c%d%d",ch,i,k);
printfs("%c%f%s%d",toupper(ch),152.53,name,pow(2,5)); //pow(x,y) i.e. x^y
printf("average=%d\n","percentage=%f",avg,per);

printf("Octal equivalent is%o\n",n);
conio.h is an abbreviation of console input output header file. **#include<conio.h>**
Followings are standard input output library functions for conio.h header file.
getch() :-getch reads a single character directly from the keyboard without echoing the screen.
getche() :-getche reads a single character from the keyboard and echoes it to the current text window, using direct video or BIOS.
clrscr() :- used to clear the screen.
Mathematical functions such as cos(x), sin(x), abs(x)sqrt(x) when used must include the header file **math.h. #include<math.h>** at the beginning.
Example :-

```
#include <stdio.h>
main()
{
  int dec = 5;
  char str[] = "abc";
  char ch = 's';
  float pi = 3.14;
  printf("%d %s %f %c\n", dec, str, pi,  ch);
```

} The output of the above would be: 5 abc 3.140000 c
****
STORAGE CLASSES :-

A storage class defines the scope (visibility) and life time of variables and/or functions within a C Program. There are following storage classes which can be used in a C Program

- auto
- register
- static
- extern

auto - Storage Class

auto is the default storage class for all local variables.

```
{
    int Count;
    auto int Month;
    }
```

The example above defines two variables with the same storage class. auto can only be used within functions, i.e. local variables.

register - Storage Class

register is used to define local variables that should be stored in a register instead of RAM. This means that the variable has a maximum size equal to the register size (usually one word) and cant have the unary '&' operator applied to it (as it does not have a memory location).

```
{
    register int  Miles;
    }
```

Register should only be used for variables that require quick access - such as counters. It should also be noted that defining 'register' goes not mean that the variable will be stored in a register. It means that it MIGHT be stored in a register - depending on hardware and implimentation restrictions.
static - Storage Class
static is the default storage class for global variables. The two variables below (count and road) both have a static storage class.

```
static int Count;
    int Road;
    {
      printf("%d\n", Road);
        }
```

static variables can be 'seen' within all functions in this source file. At link time, the static variables defined here will not be seen by the object modules that are brought in.

static can also be defined within a function. If this is done the variable is initalised at run time but is not reinitalized when the function is called. This inside a function static variable retains its value during vairous calls.

```
void func(void);
     static count=10; /* Global variable - static is the default */
     main()
  {
    while (count--)
    {
       func();
    }
  }
     void func( void )
  {
    static i = 5;
    i++;
    printf("i is %d and count is %d\n", i, count);
  }
```

This will produce following result

```
  i is 6 and count is 9
  i is 7 and count is 8
  i is 8 and count is 7
  i is 9 and count is 6
  i is 10 and count is 5
  i is 11 and count is 4
  i is 12 and count is 3
  i is 13 and count is 2
  i is 14 and count is 1
  i is 15 and count is 0
```

NOTE : Here keyword *void* means function does not return anything and it does not take any parameter. You can memoriese void as nothing. static variables are initialized to 0 automatically.

Definition vs Declaration : Before proceeding, let us understand the difference between*defintion* and *declaration* of a variable or function. Definition means where a variable or function is defined in realityand actual memory is allocated for variable or function. Declaration means just giving a reference of a variable and function. Through declaration we assure to the complier that this variable or function has been defined somewhere else in the program and will be provided at the time of linking. In the above examples *char *func(void)* has been put at the top which is a declaration of this function where as this function has been defined below to *main()*function.

There is one more very important use for 'static'. Consider this bit of code.

```
char *func(void);
  main()
  {
    char *Text1;
    Text1 = func();
  }
  char *func(void)
  {
    char Text2[10]="martin";
    return(Text2);
  }
```

Now, 'func' returns a pointer to the memory location where 'text2' starts BUT text2 has a storage class of 'auto' and will disappear when we exit the function and could be overwritten but something else. The answer is to specify static char Text[10]="martin";

The storage assigned to 'text2' will remain reserved for the duration if the program.

extern - Storage Class :-

extern is used to give a reference of a global variable that is visible to ALL the program files. When you use 'extern' the variable cannot be initalized as all it does is point the variable name at a storage location that has been previously defined.

When you have multiple files and you define a global variable or function which will be used in other files also, then *extern* will be used in another file to give reference of defined variable or function. Just for understanding *extern* is used to decalre a global variable or function in another files.

File 1: main.c

```
int count=5;
  main()
  {
    write_extern();
  }
```

File 2: write.c

```
void write_extern(void);
  extern int count;
  void write_extern(void)
  {
    printf("count is %i\n", count);
  }
```

Here *extern* keyword is being used to declare count in another file.

Now compile these two files as follows :- gcc main.c write.c -o write

This fill produce *write* program which can be executed to produce result.

Count in 'main.c' will have a value of 5. If main.c changes the value of count - write.c will see the new value.

****

14. DECISSION CONTROL STRUCTURE :-

One of the fundamental properties of a computer is its ability to **repetitively execute a set of statements or instructrions one by one in a sequential manner**. There are many situation in which this sequential execution need to be altered with few instructions being executed only when a condition is satisfied. There may also be occasiobn when a group of instuctions to be repeated for a fixed number of times until some condition is satisfied. Such situations may require control statements in the program.

The control structures can be as follows :-

1) Conditional control structures :-

Here, conditions are tested for whether result is true for false and the action is taken according to the result. There are three strcutures available under this group.

a) if **b) if-else** **c) nested if-else**

2) Multiple branching control structure :-

In this case a selection is made from sveral alternatives availabkle, and a group of instructions pertaining to the elected alternatives onkly executed. These is onky one type under thius group using keywords.

a) switch-case

3) Loop control structure :-

The is repeated execution of group of instructions till a condition evaluates to a true. There are three strcutures available under this group.

a) while **b) do-while** **c) for**

4) Jump control structure :-

These statement cause to pass to any desired location in the program, They are :-

a) break **b) continue** **c) goto** **d) exit()** **f)return**

1. Conditional Control Structure :-

a) **If** :-

Syntax :- if (condition)

```
{
    Statemets;
}
```

The condition must be placed within parentheses and the statement end with a semicolon. The construct is executed in two steps. First, the condition is evaluated. In the second step, a statement or group odf statements are executed if the evaluation is **true**.

If the condition evaluation is **false** the statement or the entire block of statement within the curly braces is skipped without execution.

Example :-

```
if (salary>=1000)
        {
          da=10;
          hra=20;
          gpf=5;
        }
```

Exercise :- WAP in C to find maximum among three given numbers.

```
#include <stdio.h>
#include <conio.h>
int main()
    {
        float a, b, c;
        printf("Enter three numbers: ");
        scanf("%f %f %f", &a, &b, &c);
        if(a>=b && a>=c) printf("Largest number = %2f", a);
        if(b>=a && b>=c) printf("Largest number = %2f", b);
        if(c>=a && c>=b) printf("Largest number = %2f", c);
        return 0;
    }
```

b) **if-else** :-

```
syntax :-   if (condition)
                    {
                      statements;
                     }
```

```
else
    {
      statements;
    }
```

First the condition is evaluated. If irt is true, then statements within if block are executed. If it is false then the stratements within elswe block is executed. The if block is called TRUE block and else block is called FALSE block.

Example :-

```
if (num%2 == 0)
    {
      printf(“the number is even”);
    }
else
    {
      printfs(“the number is odd”);
    }
```

```
#include<stdio.h>
#include<conio.h>
int main()
    {
      int a,b,c;
      int big;
      printf("Enter any there numbers: ");
      scanf("%d%d%d",&a,&b,&c);
      if(a>b && a>c)
          big = a;
      else if(b>c)
          big = b;
      else
          big = c;
      printf("Largest number is: %d",big);
      return 0;
    } Output:Enter any there numbers: 13 25 6 Largest number is: 25
```

c) **nested if – else** :-

When a series of decisions are involved, we may have to use more than one if-else statement in nested form.

Syntax :-

```
if (condition)
        {
            if (condition)
                        {
                          statements;
                         }
             else
                    {
                      statements;
                    }
        }
    else
        {
            if (condition)
                        {
                          statements;
                         }
             else
                    {
                      statements;
                    }
        }
```

```
#include <stdio.h>
#include <conio.h>
int main()
        {
          float a, b, c;
          printf("Enter three numbers: ");
          scanf("%f %f %f", &a, &b, &c);
          if (a>=b) {
                        if(a>=c) printf("Largest number = %.2f",a);
                        else printf("Largest number = %.2f",c);
                    }
```

```
                else { if(b>=c) printf("Largest number = %.2f",b);
                        else printf("Largest number = %.2f",c);
                     }
                return 0;
            }
```

2) Multiple branching control structure :-

a) **switch – case** :-

Syntax :-

```
switch  (expression)
        {
          case value1 :
                        statement1;
          case value2 :
                        statement2;
          case value3 :
                        statement3;
           default :
                        statements;
         }
```

Switch is a multi-section structure to handle decision making. The switch structure consists of a series of case labels and an optional default case. The switch-case is alternative of else-if ladder.The switch-case statement tests the value of given variable against a list of case values and when a match is found, a block of statements associated with that case is executed.

The expression is an integer expression or characters, value1, value2, ..are constants or constant expressions and aere known as **case labels**. Each of these values should be unique within a switch statement. Block1, Block2, … are statements lists and may contain zero or more statements. There is no need to put braces around these blocks. The case labels are end with a colon(**:)**.

The switch is executed , the value of the expression is successfully compared against the value1, value2, …. If a case is found whose value matches with the value of the expression, then the block of statements that follows the case are executed.

The break statement at the end of each block signals the end of a particular case and causes an exit from the switch statement, transferring the control to the statement-x following the switch.

The default is an optional case. When present, it will be executed if the value of nthe expression does not match with any of the case values. If not present, no action takes place if all matches fail and the control goes to the statements after the switch-case statement.

Example :- A C program to check an alphabet is vowel or not using **SWITCH** statement:-

```
#include <stdio.h>
#include <conio.h>
main()
	{
	char ch;
	printf("Enter a character\n");
	scanf("%c", &ch);
	switch(ch)
		{
		case 'a' :
		case 'A' :
		case 'e' :
		case 'E' :
		case 'i' :
		case 'I' :
		case 'o' :
		case 'O' :
		case 'u' :
		case 'U' :
		printf("%c is a vowel.\n", ch);
		break;
		default :
		printf("%c is not a vowel.\n", ch);
		}
		return 0;
	}
```

3) Loop control structure :-

The loop control structures are useful to execute a statement or a set of statements for a particular number of times. The following are the loop control structures.

a) **while** :-

Syntax : while (condition)

```
{
    Statements;
}
```

The condition is first evaluated and if the condition is true , then the body of the loop is execuyed. After execution of the body , the test condition is once again evaluated and if it is true , the body is executed once

again. This process of repeated execution of the body continues untill the test-condition fanally becomes false and the control is transferred out of the loop. On exit, the program continues with statement immediately after the body of the loop.

The body of the loop may have one or more statements. The braces are needed only if the body contains two of more statements. However, it is a goos practice to use braces even if the body has only one statement.

Example :- while (i<=10)

```
{
    printf("%d",i);
}
```

Example of C Program to print natural numbers from 1 to 10.

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int num=1;      //initializing the variable
    while(num<=10)      //while loop with condition
        {
            printf("%d",num);
            num++;              //incrementing operation
        }
    return 0;
}
```

Output :- 1 2 3 4 5 6 7 8 9 10

b) **do-while** :-

Syntax :

```
do
    {
        Statements;
    }
    while (condition)
```

In do-while first the control directly enters into the loop and execute the statements once, after that it checks the condition, If it is true the statements are executed once again so that the statements are executed repeatedly until the condition becomes false.

Example :- do (i<=10)

```
{
    printf("%d",i);
}
```

```
        while (i<=10);
```

Example of C Program to print Table 2.

```
#include<stdio.h>
#include<conio.h>
int main()
    {
        int num=1;      //initializing the variable
        do      //do-while loop
            {
                printf("%d",2*num);
                num++;                  //incrementing operation
            } while(num<=10);
        return 0;
    }
```

Output will be : 2 4 6 8 10 12 1`4 16 18 20

Diffrence between while and do-while :-

In while loop , if the condition becomes false at first checking, the statements are not executed.

In do-while, the condition are executed at least once, what ever may be the condition (true of false).

In while loop the condition is evaluated once after that the statements are executed.

In do-while the statements inside the loop are executed first after that the condition is tested.

In while loop the condition is evaluated more than one compared to do-while condition.

In do-while the condition is evaluated less than one compared to while condition.

While loop can not be terminated with semicolon.

Do-while loop should be terminated with semicolon

c) **for loop** :-

Syntax :-

```
        for (expression1; expression2; expression3)
            {
              Statements;
            }
```

Here, expressions1 is called initialization expression , expression2 is called conditional statements and expression3 is called incrementation or decrementation statement.

The first components of the for, initialization expression is used to set initial value before the loop begins.

The second component, the loop condition that determines the condition that must be satisfied for the looping to continue. It is evaluated before each iteration. Looping continues as long as this condition is satisfied. When the loop condition is evaluated as false, execution of the loop is terminated and execution continues with a statenment following the for loop.

The third component, increment defines how the init expression will change each time to the loop is repeated.

Example :-

```
for  ( i=1; i<=10; i++)
     {
          Printf("%d",i)
     }
```

Notes :-

1) The body of the loop in for statement need not contain any statement at all. In that case, it becomes a do nothing loop or time delay loop.

Example :- for (i=1; i<=2000; i++);

This loop executes 2000 times without doing nothing.

2) The condition check can be a compound expression made up of various relational expressions connected by logical AND, OR operators. Also the loop testing need not be limited only to the loop control variable.

Example :-

```
for (i=1; i<20&&sum<100; i++)
    {
      sum=sum+I;
      printf("%d%d\n"I,sum);
     }
```

In the for construct, the presence of two semicolon within the parenthesis is compulsory. But may or all the three parts may be blank. The initialization can be done outside the loop. The incrementing step can be incorporated within the loop.

Example :-

```
init i=0;
 for (  ; i<=4;  )
  {
    printf("%d",i);
   }
```

If the test condition is omitted , the statenments sets up an infinite loop. Such loop can broken using break statement.

Example :-

```
init i=1;
 for (  ;  ;  )
```

```
{
  If (i>10) break;
  printf(“%d”,i);
  i++
 }
```

The for statement wuill never be executed if the test condition fails at the beginning itself.

Example :- for (x=10; x<10; x=x-1)

```
{
  printf(“%d”,x);
 }
```

More than one variable can be initialized/incremented a time in the for statement using comma operator. That is , we may have more than one expression in place of the first and third expressions of the for loop.

Example :- for (i=0,j=9;i<9;++I,j--)

```
{
  printf(“%d%d”,i,j);
 }
```

Here both i and j are initialized and modified here.

Nested for loop :-

The for loop structure can be nested with one another.

Example :-

```
for (i=1;i<=20;i++)
  {
    for (j=1; j<=10; j++)
        {
           printf(“%%”,I,j);
         }
  }
```

```
// Program to calculate the sum of first n natural numbers
// Positive integers 1,2,3...n are known as natural numbers
#include <stdio.h>
#include <conio.h>
int main()
    {
    int num, count, sum = 0;
    printf("Enter a positive integer: ");
    scanf("%d", &num);
    // for loop terminates when num is less than count
```

```
for(count = 1; count <= num; ++count)
                                        {
                                          sum += count;
                                        }
printf("Sum = %d", sum);
return 0;
}
```

Output : Enter a positive integer: 10
Sum = 55

4) Jump Control Structures :-

a) **The break statement** :-

The break statement has two uses :- It is used to terminate the case in switch statement. It is also used to force immediate termination of a loop, bypassing the normal loop conditional test. When the break statement is encountered inside a loop , the loop is immediately terminated and program control resumes at the next statement following the loop.

```
#include <stdio.h>
#include <conio.h>
int main()
        {
            int num = 5;
            while (num > 0)
                            {
                              if (num == 3)
                              break;
                              printf("%d\n", num);
                              num--;
                            }
        }   Output : 5 4
```

b) **the continue statement** :-

The continua statement works similar to the break statement. Instead of forcing termination, continue forces the next iteration of the loop to take place, skipping any code in between. For the for loop, continue causes the conditional test and increment portions of the loop to execute. For the while and do-while loops, program control passws to the conditional tests.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
        {
            int nb = 7;
            while (nb > 0) {
                        nb--;
                        if (nb == 5)
                        continue;
                        printf("%d\n", nb);
                    }
    }  Output : 6 4 3 2 1
```

c) **The goto statement** :-

This is an un-conditional branching statement and its use is discouraged in the structure programming. If we want to come out of several nested loops at one stroke, the use of the goto statement may be useful.

Syntax : goto label;

Here label is an identifier which indentifies the statement to which is to be passed. Control can be passed to any part of the current function in which the goto appears.

```
#include <stdio.h>
int main()
        {
            int num,i=1;
            printf("Enter the number whose table you want to print?");
            scanf("%d",&num);
            table :
            printf("%d x %d = %d\n",num,i,num*i);
            i++;
            if(i<=10)
            goto table;
        }
```

Output :- Enter the number whose table you want to print?10

10 x 1 = 10

10 x 2 = 20

10 x 3 = 30

10 x 4 = 40

10 x 5 = 50

10 x 6 = 60

10 x 7 = 70

10 x 8 = 80

10 x 9 = 90

10 x 10 = 100

d) **the exist() function :-**

It is a library function in the headerfile <stdlib.h>

This function causes immediate termination of the entire program, forcing a return to the operating system. In effect, the exit() function acts as if it were breaking out of the entire program.

Syntax:- exit(return_code);

```
#include <stdlib.h>
#include <stdio.h>
int main()
        {
        char choice;
        choice = getchar();
        if(choice=='Q')
       { exit(0);
        } else
        {
            printf("Forget Code Retains");
        }
          return 0;
        }
```

15. ARRAYS :-

It is data-type in C which allows to assigning multiple values with a single variable name.Variables can be accessed and manipulated easily using array as they are present in contiguous memoey location. It is simple data structure widely used in different application like stack, queue, heap and many more.

Example of normal variables used so far :-

```
main()
        {
           int x; x=90; x=10; x=30; x=5; x=100;
           printf("%d%d%d%d%d",x,x,x,x,x);
         }
// Output will be : 100 100 100 100 100 100
```

Here in the program different values are assigned to x. But x is unable to hold all these different values and we are unable to prepare the list. When it is necessary to hold nore than one value in a single name we need an array as example below":-

```
main()
  {
           int x[5]=90,10,30,5,100};
           int I;
           for(i=0; i<=5; i++)
                 printf("%d",x[i]);
           }
// Output : 90, 10, 30, 4, 100
```

Types of Array :-
One dimentional array such as :- int x[15];
Two dimentional array such as :- int y[3][5];

Single Dimentional Array :-
A one-dimentional array is a list of values of same data type and stored in the format of a row. A one-dimentional array contains only one subscript.
Example :- int A[100];
(i) Declaration of Array :-
Like any other variable, arrays must be declared before they are used.
Syntax :- data-type ArrayName[array-size];
The data-type can be int, char, float, double or any other data types available in C.
Array-name must be a valid bvariable.Array-size represents total no of element in aray.
Example :- int age[10]; declares age as an array containing maximum 10 integerrs.
float height[20]; declares height as an array containing maximum 20 float numbers.
char name[20]; declares the name as character array containing maximum 20 characters.

(ii) Accessing Array elements :-
Once an array variable is defined , its elements can be accessed by using an index.
Syntax :- ArrayName[index];
To access array elements in an array, specify the array name followed by square braces enclosing an integer, which is called an array index. The array index indicates the particular element of the array which we want to access. In an array first element is indexed bu 0 (zero) and the last element of an array is indexed by N-1;
Example :- printf("5",age[4]);
Above expression accessed the 5^{th} element of the array 'age'.

(iii) Initialization of Array :-
We can inialize the elements of arrays in the same way as the ordinary variables when they are declared.

Syntax :- Data-type array-name[size] = {value list };
The value-list is a common-separated list of constants.
Exampe :- int age[10]={30,25,56,38,60};
float height[20]={5.4,6.2,5.9,6.8,4.5};
char name[20]={'r','a','m','a','n','a','\o');
If initalisation value are fewer than cells in the array then remaining cells are initialized to zero.
The declaration of maximum number of cells in the array can be omitted if the initialization are present.
Example :- int mark[]={45,56,78,89};

iv) Perocessing with Arrays :-
An easy way to read values to an array is by using a for loop.

```
for (i=0; i<n; i++)
   {
     scanf("%d",&A[i]);
   }
```

The conmtents of of an array can be accessed and printed as follows :-

```
 for (i=0; i<n; i++)
   {
     printf("%d",&A[i]);
   }
```

Note : C does not require that range of the subscript is checked . If the index was allowed to go out of the range then the result will be unpredicatable when the program is executed.

Representation of Single Dimentional Array :- int x[5]={5,6,3,1,8};
X[0]=5, x[1]=6, x[2]=3, x[3]=1,x[4]=8

** We can access any element of an array in the following different ways.
** Subscript format and Pointer format.

Two-dimentional arrays :-
Multi dimentional arrays are repitive structures. For example a two-dimentional array is a repetition on one dimentional arrays. A two dimentional array is composed from one-dimentional array. Two dimentional arrays are used to store the data in the form of rows and columns such as the data resided in the form of matrices of in the form of tables.
Example :- int x[5] [5] ;

Declaration :- Syntax :- Data-type array-name[row-size][colum-size];
Note that C requires that each subscript can be written within its own separate pair of brackets.
Example :- int value[10][10]; float marks[5][[10];
Accessing two dimentional array :-
The elements of a two dimentional array can be accessed by using two indexces called row index and column index. For example, if we want to access the second row and third column element to print in a two dimentional array X, then we write :- printf(%d",X[1][2]);
The general form to access all elements is :- printf("%d",X[i] [j];

Initialization :- Contents of the two-dimentional array can be initialized in a similar way to one-dimentional array. The elements are to be initialized one by one.
Example :- INT A[3][2]={1,2,3,4,5,6};
Alternatively, elements of each row can be enclosed withibn pair odf braces with each row separated from the next by a comma, as follows :- int a[3][2]={{1,2},{3,4},{5,6});
In a row initialization values does not contain enough rvalues then the remaining elements are set to zero. Thus the following two assignments are equivalent.
Example :- int a[3][2]={{{1,2},{3},{4}};
Example :- int a[3][2]={{{1,2},{3,0},{4,0}};
Here the first element of each row is explicitely initialized to zero. While other elements are automatically initialized to zero.

Processing with two-dimentional array :-

Nested loopo are useful when dealing with two dimentional arrays. Elements of a two-dimentional array can be read , accessed and printed using a nested loop as follows.

Reading elements of two-dimentional array :-

```
for (i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
          {
                scanf("%d",&x[i][j]);
           }
    }
```

Printing elements of two-dimentional array :-

```
for (i=0; i<m; i++)
    {
        for (j=0; j<n; j++)
          {
                printff("%d",&x[i][j]);
           }
    }
```

To organize all the rollnumbers in different categories, we use double dimentioanl array.

int roll[100]; can be represented in 2-D array int roll[2][50]; or int roll[5][20]; etc.

FUNCTIONS :-

What is a Function : It is block of codes to perform specific task and the codes that are to be executed/run again again.

There are two types of function. 1) Library function Ex:- scanf(), printf(), clrscr(),exit()

2) User defined function. (Defined by the user for specific application or purpose)

A C program is nothing but collection of 1 or more functions. It contains atleast one function i.e. main function from where actual execution of statements begins and main function will call other functions. User defined functions can be defined above the main function or below the main function. If defined below main function then need to declare prototype of the function i.e function declaration so that compiler knows there exists a function. If defined above main function then write the codes of function definition. Function can be parameterized or arguments or non- parameterized (if no parameters) or no arguments.

Again argumrnts can be actual arguments or formal argumrnts.

Function Category :-

(i) No Return type with No arguments or parameters.

(ii) Return type with No arguments or parameters.

(iii) No Return type with argumernts or [arameters.

(iv) Return type with arguments or parameters.

Example1 : **To print message**

```
#include <stdio.h> #include <conio.h>  // void function does not return any value
// 1. No return type function i.e. void 2. Name of function is message 3. No parameters
void message( )     // declaration of function before main function
                        {  // Here function declared above main function
                          printf(”Hello Welcome to Polytechnic\n”);
                }
int main()
                {   // to call name of the function message
                    message( ); // Here no need to write return type
                    message( );
                    message( ); // message function is called 3 times
                    return 0;
                }
// 2. return type of the function is void 2. Name of function is message 3. No parameters
#include <stdio.h> #include <conio.h>
void message( )     // declaration function after main function & we declare prototype
int main()
                {
                    message( );
                    message( );
                    message( ); // message function is called 3 times
                    return 0;
                }
void message( )     // function definition after main
                        {
                           Printf(“Hello Welcome to Polytechnic\n”);
                        }
```

Example : To add 2 numbers or values

```
#include <stdio.h> #include <conio.h>  // void function does not return any value
// 1. return type of the function is void 2. Name of function is sum 3. Two parameters
```

```
void sum(int firstno, int secondno)
                                        {  // function name sum has 2 parameters & data type int
                                          Printf(”Sum of 2 numbers is”,firstno+secondno);
                                         }
int main()    {  // to call name of the function sum with passing value
                   sum(476, 587);
                   return 0;
                 }
// Modify the Program slightly #include<iostream> using namespace std;
void sum(int firstno, int secondno)    // function pass values & returns nothing
                                        {
                                          printf(”Sum of 2 numbers is”,firstno+secondno);
                                         }
int main()       {  int a, b;
                   printf(”please enter 2 numbers\n”);
                   scanf(“%d%d,&a,&b); // User will enter values and will pass to the function
                   sum(a,b);
                   return 0;
                 }
#include <stdio.h> #include <conio.h>
void sum(int firstno, int secondno)    // function pass values & returns nothing
                                        {
                                          printf(”Sum of 2 numbers is”,firstno+secondno);
                                         }
int main()       {  int a, b;
                   printf(”please enter 2 numbers\n”);
                   scanf(“%d%d”,&a,&b);
                   sum(a,b);
                   int x, y;
                   printf(”please enter 2 numbers\n”);
                   scanf(“%d%d”,&x,&y);
                   sum(x,y);
                   return 0;
                 }
#include <stdio.h> #include <conio.h>
int sum(int firstno, int secondno)    // function pass values & returns something
```

```
                                        {
                                          int result=0; // if not initialized garbage will come
                                          result=firstno + secondno;
                                          return result ;
                                         }
int main()      {  int multiply = 2;
                   int mainresult;
                   mainresult=multiply * sum(253, 456);
                   printf(”Result is%d ”,sum(253,456));;
                   printf(”The main result is%d”,mainresult);
                   return 0;
                }
```

// Program To find maximum among 2 given values

```
#include <stdio.h> #include <conio.h>
int max(int a, int b)  // declaration
int main()      {  int a, b;
                   printf(”please enter numbers\n”);
                   scantf(%d%d”,&a,&b);
                   int c=max(a,b);
                   printf(”The greatest value is\n”);
                   return 0;
                }
int max (int x, int y)              // a & b corresponds to x & y
        {   If (x>y)
                   return x ;
          else
                   return y ;
        }
```

Example : Default Parameter

```
#include <stdio.h> #include <conio.h>
int sum(int firstno=20, int secondno=30)    // initialized function with initial value
                                        {
                                          int result=0;
                                          result=firstno + secondno;
                                          return result ;
                                         }
```

```
int main()      {
                    printf(sum( ));
                    return 0;
                }
#include <stdio.h> #include <conio.h>
int sum(int firstno=20, int secondno=30)    // initialized function with initial value
                                            {
                                              int result=0;
                                              result=firstno + secondno;
                                              return result ;
                                             }
int main()      {
                    printf("%d",sum(134 )); // Here only one value will be assigned to first
                    return 0;
                }
int main()      {
                    printf("%d",sum(134, 234 )); // Here two value are assigned to first & second
                    return 0;
                }
```

If declared below main then initialize in prototype not in function definition

```
#include <stdio.h> #include <conio.h>
int sum(int firstno=20, int secondno=30)    // initialized function with initial value
int main()      {
                    printf("%d",sum(134,234));
                    return 0;
                }
int sum(int firstno, int secondno)
                                            {
                                              int result=0;
                                              result=firstno + secondno;
                                              return result ;
                                             }
```

Example: Call by Value Function : Take something & return something

// Formal arguments are when we define the function in function Definition

// formal arguments are of 3 types ordinary variable, pointer variable & Reference variable

// **Here formal arguments are ordinary variables**

// When formal arguments are ordinary variables then in the function we pass the values we Call as PASS BY VALUE

```
#include <stdio.h> #include <conio.h>
int sum(int , int)                    // Function Prototype
void main()
                    {
                      int a=5, b=6;
                      int s;
                      s=sum(a,b);  // a & b are actual arguments
                      printf("Sum of 2 numbers are%d ",s);
                    }

int sum(int x, int y)  {   // Function Definition
                              return(x+y);  // x & y are formal arguments & when we define function
                           }
// Example of call by Value function To calculate factorial of a number
#include <stdio.h> #include <conio.h>
void main() {
                    int num, f;
                    printf("Enter a number\n");
                    scanf("%d",&num);
                    f=fact(num);
                    printf("%d",f)'
                 }
fact (int x)
                {
                    int p=1;
                    while (x>=1)
                        {
                            p=p*x;
                            x--;
                        }
                    return p ;
                }
```

Example: Call by Address Function

// When formal arguments/parameters are pointer variables it is CALL BY ADDRESS Function

```
#include <stdio.h> #include <conio.h>
```

```
int sum(int  *, int *)              // Function Prototype (Star)* indicate pointer variables
void main()
                {
                  int a=5, b=6;
                  int s=sum(&a, &b);  // Here 5 & 6 is not passed but address of a & b passed
                  printf(”Sum of 2 numbers are%d“,s);
                }
int sum(int *p, int *q)  {          // Function Definition
                          return(*p + *q);  // p & q are formal pointer variables
                        }
```

Example: Call by Reference Function

// When formal arguments are reference variables it is CALL BY REFERENCE Function

```
#include <stdio.h> #include <conio.h>
int sum(int  &, int &)              // Function Prototype & indicate reference variables
void main()
                {
                  int a=5, b=6;
                  int s=sum(a, b);  // Here looking it, will not show call by value
                  printf(”Sum of 2 numbers are%d “,s);
                }
int sum(int &x, int &y)  {          // Function Definition
                          return(x + y);  // x & y are formal reference variables
                        }
```

Example: Pass by value Function

```
#include <stdio.h> #include <conio.h>
void passbyval ( int val )                  // create a function
                                    {
                                      val = 100;
                                    }
int main   {
                int x=20;
                passbyval(x);
                printf(“%d”,x);      // Output is still 20 eventhough we have assigned 100
                return 0;
          }
```

Example: Pass by reference Function

```
#include <stdio.h> #include <conio.h>
void passbyref ( int *val )                    // create a function
                                    {
                                      *val = 100;
                                    }
int main   {      int x=20;
                  int y=20;
                  passbyval(x);
                  printf("%d",x);
                  passbyref(&y)
                  printf("%d",y);
                  return 0;
           }                        Output 20 100
```

POINTER :

What is a pointer :- Pointer Points to Memory Location. Pointers are used to access the information from a Memory Location or Cell i.e a Pointer is a variable, which holds the memory address of another variable. If one variable contains the address of another variable, then the first variable is said points to the second.
If the memory is not there and accessing the data is not there, then what is the use of Programming Language and hence accessing the memory location is done by Pointers.

Normally, variable declaration is : Data-type Identifier **;** Example :- int a ;

(Identifier is any variable name to identify it.)

Syntax : Pointer variable is declared by adding a * symbol either in front of the variable or followed by the Data-type.

Example :- int***** a**;** or int *****a**;**

How many types of pointers :- 1) Types Pointer 2) Un-typed Pointer

Typed Pointer : Points to specific type of data i.e integer pointer (int *a) points to only integer type data, *double pointer (double *b) points to double type data and character pointer points to character type data. It is* not that integer pointer will points to double type of data.

Ex:- int *a;

Un-typed Pointer : It can points to any type of data. It is also called Generic Pointer and Void Pointer can points to any type of data. Ex:- void * a;

Pointer operations are done with the help of **two operators** only and they are :-

1) **&** called **address of operator** which returns the address of a particular variable or memeory location

Syntax : & variablename Example :- &i

2) ***** called **pointer operator** and returns the value which is inside of a specific address or called value at the address operator. Syntax : ***** memorylocation-address

Example :- *(&i)

Note : Both pointer operators are unary operators since they operate on a single operand.

Example :-

```
void main() {
    int i = 100;  // i value is 100 and say it is in memory location 2046
    int *ptr;     // declare pointer variable ptr & say gets memory location 3002
    ptr = &i;     // address of  i is stored in ptr
    printf("%d",i);   // print value of I i.e  100
    printf("%d",ptr);  // print prt value i.e  2046
    printf("%d",&i);   // print address of i i.e  2046
    printf("%d",&ptr);  // print address of ptr i.e  3002
    printf("%d",*ptr);  // print ptr location i.e 100
    printf("%d",*(&i));  // print pointer to address of i i.e  100
    } // Whenever we are printing address, it  must be +ve integer i.e. %u .
```

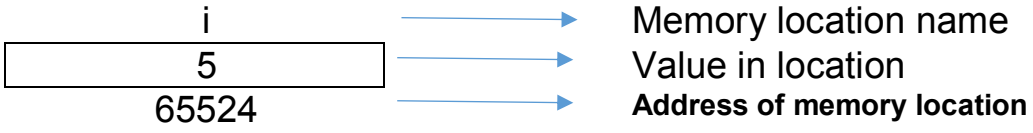
Following is the declarative statement as well as initialization statement

```
int i=5;
```

When this statement is encountered, the following tasks are done:-

- Allocates a memory location.
- The name 'i' is assigned to the allocated memory location.
- The value 5 is stored into the memory.

Diagram :-



Program: /* Demonstrating & and * operators*/

```
#include<stdio.h>
#include<conio.h>
Void main()
{
Int i =5;
clrscr();
printf("Address of i:%u \n",&i);
printf("Value of i: %d \n",*(&i) );
printf("Vlaue of i: %d \n",i);
}
```

Output:

Address of i = 65524.

Value of i = 5

Value of i = 5

Declaration of pointer variables & initialization:

A pointer variable can be declared in the same way as the other variables, but an asterisk symbol should precide the variable name. This identifies the variable is a pointer.

Syntax: Datatype * variablename;
Example:- int *p1; float * p2; char * p3;

Program: /* Accessing Address and value of variable i using its pointer*/

```
#include <stdio.h>

#include<conio.h>

Volid main()

 {

 int i=5

 it *p;

 clrscr();

 p=&i

 pintf("Address of i: %u\n", &i);

 pintf("Value of i:%d\n",*(&i));

 printf "Address of I: %u\n",p);

 print f("Value of i:%d\n",*p);

 }
```

Output :

Address of i: 65524

Value of i:5

Address of i:65524

Value of i:5

Pointer to pointer:-

We have a pointer, which contains another pointer's address. These pointer variable are called pointer – to – pointer variables.
Program:-

```
#include<stidio.h>
#include.<conio.h>
Void main()
{
int i=5
int*p,**p1;
Clrscr();
p&i;
p1=&p;
```

```
printf("Address of i:%u\n",&i);
printf("Value of i:%d\n",*(&i));
printf("Address of i:%u\n",p);
printf("Value of i:%d\n",*p);
printf("Address of i:%u\n",*p1);
printf("Value of i:%d\n",**p1);
}
Output:
Address of i: 65524
Value of i: 5
Address of i:65524
Value of i: 5
Address of iL65524
Value of i:5
```

Pointer arithmetic:-

The following operations can be performed on a pointer:

1. Addition of a number to a pointer.

```
int i=5;

int *p;

p=&I;

p=p+2;
```

2. Subtraction of a number from a pointer.

```
int i=5;
it *p;
p=&i;
p=p-2;
```

3. Subtraction of one pointer from another.

One pointer variable can be subtracted from another provided both variables point to elements of the same array.

The resulting value indicates the number of bytes separating the corresponding array elements. This is illustrated in the following program.

Program:-

```
#include<stidio.h>
void main()
{
int arr[]={10,20,30,40,50,60,70};
int*I,*j;
i= &arr[1];
j= &arr[5];
printf("%u\n",j-i);
printf("%d\n",*j-*i);
}
```

Here I and j have been declared as integer pointers holding addresses of first and fifth element of the array respectively.

Suppose the array beging at location 4002, then the elements arr[1] and arr[5] would be present at locations 4004 and 4012 respectively, since each integer in the array occupies two bytes in memory. The expression j-I would print a value 4 and not8. This is because j and I are pointing to location, which are 5 integers apart. The output of *j-*I is 36, since*j and * I return the values present at addresses contained in the pointers j and i.

4. Comparison of two pointer variables.

Pointer the array can be compared provided both variables point to objects of the same data type. Such comparisons can be useful when both pointer variables point to elements of the same array. The comparison can test for either equality or inequality. Moreover a pointer variable can be compared with zero(usually expressed as NULL). The following program illustrates how the comparison is carried out.

Program :-

```
#include<stidio.h>
Void main()
{
Int arr[ ]={10,20,30,40,50,60,};
Int*j,*k;
j= &arr[4];
k= &arr[4];
if (j==k)

{

        Printf(" The two pointers point to the same location");

}

Else

{

        Printf(" The two pointers do not  point to the same location");

}

}
```

5. Pointer Assignments.

As with any variable, you may use a pointer on the right-hand side of an assignment statement to assign its value to another pointer.

Program:-

```
#include<stidio.h>
Void main()
{

        Int x;

        Int*p1, *p2;

P1-&x;

P2=p1;

Printf("%u",p2);/*print the address of x, not x's value!*/

Return 0;

}
```

Both p1 and p2 now point to x.

Don't attempt the following operations on pointer variables.

a) Addition of two pointer
b) Multiplication of a pointer with a constant.
c) Division of a pointer with a constant.

Note: A pointer when incremented or decremented is always relative to its base type. That is, its value is increased or decreased by the length of the data type.

Pointer & Arrays:

There is a close relationship between pointers and arrays. Whenever an array is created a block of memory is created and a pointer is made to point it. We may access the elements of this array by the subscripts or with the help of this pointer. An array name can be thought of as a constant pointer.

You can generate a pointer to the first element of an array by simply specifying the array name, without any index. For example, given int sample[10];

You can generate a pointer to the first element by using the name sample. Thus, the following program fragment assigns p the address of the first element of sample:

Int *p;

Int sample[10];

P= sample;

You can also specify the address of the first element of an array using the & operator. For example, sample and &sample[0] both produce the same results. However, in professionally written C code, you will almost never see &sample[0]

One way of accessing the array elements by using pointers is illustrated below.

Programn

```
#include <stdio.h>

#include <conio.h>

void main()

{

int a[5]={1,2,3,4,5)

int i, *ptr;

clrscr();

ptr=&a[O];

for(i=0;i<5;it +)

{

printf("%u %d \n", ptr, *ptr);

ptr+t; /* Increments the pointer to point to the next element */
```

Output:

65516 1

65520 2

65524 3

65528 4

65532 5

In the above example, an integer pointer ptr is explicitly declared and assigned the starting address. The same result be obtained by using one of the property concerning pointers and arrays. It is interesting to observe that the name of the array represents the starting address (base address) of the array. It is the address of the first element in the array. Thus, the name of the array can be used effectively as any other pointer to the array in accessing the elements of the array. This is illustrated in the following example.

Program:-

```
#include <stdio.h> #include <conio.h>
void main()
{
int a[5]={1,2,3,4,5} , 1,*p;
p=a;
for(i=O;i<5;i++)
{
printf("%d ", a[i]);
printf(%d ", *(a+));
printf(%d ", i[a]l);
printf(*(i+a));
printf(%d ", *p);
p++;
printf("\n");
}
}
```

Output:

```
1 1111
2 2222
3 33 3 3
4 4444
55555
```

Program:

```
/*The following program displays the addresses of zeroth one dimensional array s[0]
first one dimensional array s[1]..... */
#include <stdio.h>
#include <conio.h>
void main ()
{
int x[3][3] = { {3,4, 7}, {6,7,2},{4,9,1}}
int i,j
clrscr();
for(i=0; i<3; i++)
{
printf("n Address of %d th 1-D array = %u \n",i,x[i]);
```

}

}

Pointers & Strings:

There is a close relationship between pointers and arrays. Consider this program

fragment:

char str[80], *p1;

p1= str;

Here, p1 has been set to the address of the first array element in str. To access the fifth

element in str, you could write

str[4] or *(p1 +4)

Both statements will return the fifth element. Remember, arrays start at 0. To access the fifth element, you must use 4 to index str. You also add 4 to the pointer p1 to access the fifth element because pi currently points to the first element of str. (Recall that an array name without an index returns the starting address of the array, which is the address of the first element.)

Pointers & Functions:

i. Parameter Passing Techniques:

Parameter passing is a mechanism for communication of data and information between the calling function and the called function. C supports the following three types of parameter passing schemes:

1. Call by Value

2. Call by Address or Call by Reference

1. Call by value:

When the arguments are passed by value, the actual arguments in the calling Tunction are copied into corresponding formal arguments of the called function. With this method the changes made to the formal arguments in the called function have no effect on the values of actual arguments in the calling function.

Program: /* call by value */

#include <stdio.h>

void swap(int x, int y);

void main()

{

int a,b;

printf("Enter any two numbers \n");

scanf(/%d%%d", 8a, &b) ;

printf("Before calling function: \n");

printf(a= %d \n", a);

printf("b= %d \n" , b);

swap(a,b);

printf(" After calling function:\n");

```
printf("a= %d \d \n", a);

printf("b=%d \n" , b);

}

void swap(int x, int y)

{

int temp;

temp=x

X=Y

y=temp;
```

Input/Output:

Enter any two numbers

10

20

Before calling function

a=10 b=20

After calling fun ction

a=10 b=20

2. Call by address :

When the arguments are passed by address, the addresses of the actual arguments in the calling function are copied into formal arguments of the called function.With this method the changes made to the format argumentsin the called function have

effect on the values of actual arguments in the calling function. This method uses the concept of pointers. In this method the formal parameters must be declared as pointers.

Using a call by address, we can make a function return more than one value at a time, which is not possible ordinarily.

Program: /* call by address */

```
#include <stdio.h>

void swap(int *x,int *y);

void main()

{

int a,b;

to gete Pefm a aje

printf("Enter any two numbers \n")

scanf(%d%d", &a,&b);

printf("Before calling function : \n ");

printf(a=%d\n",a);
```

```
printf("b=%d\n",b);
swap(8a, &b);
printf("After calling function:\n");
t mex Ccht numl nt nn2)
Helie.
printf("a=%d \n",a);
printf("b= %d\n", b);
{
hwoon()
void swap(int *x,int "y)
{
200
{
int temp
Tl= nar (a,1)
temp-*X
*x=*y;
Pndf("e=", r
*y=temp;
}
Input/Output:
Enter any two numbers
10
20
Before calling function
/feh
a=10 b=20
After calling function
t refd
nnd7vn)
a=20 b=10
```

ii. Pointers as Function Argument:

When we pass addresses to a function, the parameters receiving the addresses should be pointers. The process of calling a function using pointers to pass the addresses of variable is known as call by address.

Program:-

```
#include <stdio.h>
```

```
void modify(int *p);
void main()
     {
      int x;
      X-20;
     modify(8x);
printf(%d",x);
}
void modify(int *p)
{
*p = *p + 10;
}
```

iii**. Returning Pointers**:

Functions can return addresses as they return integers, float or character type data. This return type must explicitly be declared using pointers.

Syn:

```
returntype *function-name(arguemnts-list)
{
/* function body*/
}
```

Program:

```
#include <stdio.h>
int *func();
void main()
{
int *b;
int func(); /function that returns pointer/
clrscr();
b=func()
printf(%u %d \n", b, *b);
}
int func()
{
int j = 50;
return(&j) ;}
```

iv**. Pointer to Function:**

As we have pointers to other data types, we can also have pointers to functions. Like other variable function are also having addresses associated with them. A pointer to a function can be passed to another function as an argument. This allows one function to be transferred to another as the first function was variable.

Syn: data type (*funptr) (arguments)

Program:

```
#include <stdio.h>

void show();

void main()

{

void (*funptr)0; /*it is pointer to function"/

void showO;

funptr = show; /#address of show is assigned to funptr/

funptr();

}

void show()

{

printf("This is function show");

}
```

Dynamic Memory Allocation:

OR

Memory Management Operators:

A problem with many simple programs such as those written so far is that they tend to use fixed-size arrays, which may or may not be big enough. There are more problems of using arrays. Firstly, there is the possibility of overflow since 'C does not check array bounds. Secondly, there is wastage of space if an array of 100 elements is declared and a few are used, it leads to wastage of memory space.

How can the restrictions of fixed-size arrays be avoided? The answer is dynamic memory allocation. It is the required memory that is allocated at run-time (at the time of execution). Where fixed arrays are used. Static memory allocation, or memory allocated at compile time, is used. Dynamic memory allocation is a way to defer the decision of how much memory is necessary until the program is actually running, or give back memory that the program no longer needs.

The area from where the application gets dynamic memory is called heap. The heap starts at the end of the data segment and grows against the bottom of the stack. If both meet, the program is in trouble and will be terminated by the operating systenm. Thus, C gives programmers the standard sort of facilities to allocate and de-allocate dynamic heap memory. These will be discussed here.

Static memory allocation: The compiler allocates the required memory space for a declared variable. By using the address of operator, the reserved address is obtained that may be assigned to a pointer variable. Since most declared variable have static memory, this way of assigning pointer value to a pointer variable is known as static memory allocation.

Dynamic memory allocation: A dynamic memory allocation uses functions such as mallocO or calloc() to get memory dynamically. If these functions are used to get memory dynamically and the values returned by these functions are assigned to pointer variables, such assignments are known as dynamic memory allocation. Memory is assigned during run-time.

C provides access to the heap features through library functions that any C code can call. The prototypes for these functions are in the file <stdlib.h>. So any code which wants to call these must #include that header file. The four functions of interest are as follows:

void* malloc (size_t size): Request a contiguous block of memory of the given size in the heap. malloc() returns a pointer to the heap block or NULL if the request is not satisfied. The type size_t is essentially an unsigned long that indicates how large a block the caller would like measured in bytes. Because the block pointer returned by malloc() is a void * (i.e., it makes no claim about the type of its

pointee), a cast will probably be required when storing the void pointer into a regular typed pointer.

Calloc(): works like malloc(), but initializes the memory to zero if possible. The prototype is

void * calloc (size_t count, size_t eltsize)

This function allocates a block long enough to contain an array of count elements,

each of size **eltsize**. Its contents are cleared to zero berore calloc() returns.

void free (void* biock): free () takes a pointer to a heap block earlier allocated by maloc() and returns that block to the heap for reuse. After the free(), the clier should not access any part of the block or assume that the block IS Vand The block should not be freed a second time.

Void realloc(void* block, size t size) : Take an existing heap block and try to reallocate it to a heap block of the given size which may be larger or smaller than the original size of the block. It returns a pointer to the new block, or NULL if the reallocation was unsuccessful. Remember to catch and examine the return value of realloc(). It is common error to continue to use the old block pointer.

Realloc) takes care of moving the bytes from the old block to the new block. realloc(O exists because it can be implemented using low-level feakures that make it more efficient than the C code a programmer could write.

To use these functions, either stdlib.h or alloc.h must be included as these functions are declared in these header files.

Program: 1:

```
/* Demonstrating malloc() function */

/* Adding 'n' given numbers */

#include <stdio.h>

#include <conio.h>

#include <alloc.h>

void main()

{

int n,i, sum=0, *ptr;
```

```
clrscr();
printf("Enter n value \n");
scanf("%d", &n);
ptr=(int *) malloc(n *sizeof(int));
printf("Enter those numbers \n");
for(i=0;i<n;i++)
{
scanf("%d", ptr+i)
}
for(i=0;i<n;i++)
{
Sum=sum+(ptr+1);
}
free(ptr)
printf( The sum is %d",sum);
}
```

Program:2:/* Demonstrating realloc() function*/

```
#include <stdio.h>
#include <alloc.hs
#include <string.h>
int main(void)
{
char *str;
/*allocate memory for string */
str= (char *) malloc(10);
/ copy "Hello" into string */
strcpy(str, "Hello");
printf("String is %s\n Address is %p\n", str, str);
str (char *) realloc(str, 20);
printf("String is 9%s\n NeW address is %p\n", str, str);
/* free memory */
free(str);
return 0;
}
```

Functions and Arrays:

- We can also pass an array variable as an argument to an user defined function as an argument just by specifying an array name without index.

- Array variables are passed by address.

```
void main()
{
int a[10];
funcl(a);
.
.
.
}
```

- If a function receives a single-dimension array, you may declare its formal parameter in one of three ways:
  a) as a pointer

  b) as a sized array

  c) as an unsized array.

  Example:

```
void func1 (int *x) /* pointer */
{
.
.
.
}
```

Or

```
void funci (int x101)/* sized array */
{
.
.
.
}
```

Or

```
void func1(int x[1)/* unsized array */
{
.
.
.
}
```

Program: 1. /* W.A.P to sort 'n' given numbers in ascending order */

```
#include <stdio.h>
```

```
# include <conio.h>
void main()
{
int a[ 20],n,i,j,temp;
void sort(int x[10], int n);
clrscr();
printf("Enter how many numbers u have");
Scanf("%d", &n);
printf("Enter those numbers");
for(i=0;i<n;it+)
{
scanf("%od",&a[i]);
}
Sort(a,n);
printf"The following are the sorted elements\n");
for(i=0;i<n;i++)
{
printf("%5d", a[il);
}
}
void sort(int x[10], int n)
{
int I,j,temp;
for(i=0;i<n;it+)
{
for(j-i+ 1;j<n;j++)
{
if(x[i]>x[j)
temp x[i];
x[i]=x[j};
x[]=temp;
}
}
}
}
```

Passing strings as Function arguments:

OR

Functions & Strings:

When an array is used as a function argument, its address is passed to a function, This is an exception to the call-by-value parameter passing convention. In this case, the code inside the function is operating on, and potentially altering, the actual contents of the array used to call the function.

Program:1: W.A.P to convert a given lowercase string into an uppercase string.

```
#include <stdio.h>
#include <ctype.h>
void print_upper(char "string);
int main(void)
{
char s[80];
gets(s);
print upper(s);
printf(ns is now uppercase: %s", s);
return 0;
}
/* Printa string in uppercase.*/
void print upper(char *string)
{
register int t;
for(t-0; string[t]; ++t)
{
string[t]= toupper(string [t
putchar(string[t])
}
}
```

STRUCTURE :

What is a structure :- It is a user defiened datatype. Using structure, we can define data-type which holds more than one element of different data-types. It holds n number of elements of different data-type. When one or more variables are grouped together under one name it is known as Structure. Structures are used to group different data-types in one object. It is called compound data-type.

How to define a structure (syntax) :-

struct < structure-name > // structure-name is the identity or structure variable name.

```
// struct is keyword to define structure data-type.
{
   data-type element1 ;
   data-type element2 ;
   data-type elementn ;
} ;       // Every structure ends with a semicolon (;).
```

Example of Employee Structure :-

```
struct employee   {
                     int eno;
                     char name[20];
                     float esal;
                  };
```

struct is a keyword and it is useful to define user defiened data types. The name of the structure is employee. As such, employee identifies that particular data structure and is its type specifier. The elements may be any valid C identifier. The data-type may be any valid primary data types or any valid user defined data type. Notice that declaration is terminated by semicolon.

Anothwer way declaration of structure variables :-

```
struct student
        {
           int rno;  int age; char gender; float ht; float wt;
        }  s1, s2;
```

The above structure defines a structure type called student and declares variables s1 and s2 of that type. If we only need one structure variable, the structure type name is not needed. That means that:-

```
struct student
        {
           int rno;  int age; char gender; float ht; float wt;
        } s1;
```

Syntax :- struct struct-type-name structure-variable1, structure-variable2,.. structure-variablen;

Example :- struct student s1,s2;

The above statement declares structure variable of type **student** called **s1, s2**.

When a structure variable is declared, the compiler, automatically allocates sufficient memory to accommodate all of its members.

Accessing members of structure :- Accessing is done mainly with dot (.) operator.

Two operators are used to access members of strutures.

(i) The structure member operator (**.**) also called the dot operator.

(ii) The structure pointer operator (**>**) also called the arrow operator.

The structure member operator accesses a structure member via the structure variable name.

Syntax :- struct-variable.struct-member;

Example :- s1.rno, s1.age, s1.gender, s1.ht, h1.wt

Example :- struct emp {

```
                    int eno; char ename[20];
                    float esal;
                  };
        main() {
                struct emp e;
                printf("Size of  Employee : %d",sizeof(e));
                printf("Size of Emp%d",sizeof(struct emp));
              }   // output 26
```

Local structure and Global structure :-

Declaration of structure inside a perticular function is local structure and outside to all the functions is global structure.

Defining inside main is local structure		**Defining outside main is global structure**

Mian() {
Struct local{ int a,b; };
Struct local l;
}
		Struct global { int a, b; };
		Mian() {
		Struct global g;
		}

Check() { struct local l; } will give error		Check() { struct globa;l g; }

Assigning values to the member of the structure :-

By using an assignment operator (=), we can assign the right side value to the structure members.

Syntax : structure-variable.structure-member=value;

Example :-

s1.rno=100; s1.age=24; s1.gender='M'; s1.height=5.5; s1.weight=60.5;

Example :

```
#include <stdio.h>  #include <conio.h>
    Struct student          // declaration of structure
                {
                   int eng,
                   int math,
                   int science;
                };
    int main( )
            {
                int total;
                struct student stud;
                stud.eng=75; stud.math=40; stud.science=60;
                total=stud.eng+stud.math+stud.science;
                printf("Total=%d", total);
                return 0;
            }
```

Initialization of structure :-

Structure can be initialized lists as with arrays. To initialize a structure, follow the variable name in the structure declaration with an equals sign and a brace-enclosed, comma-separated list of initializers. For example, the declaration :-

```
struct student
            {
              int rno;  int age; char gender; float ht; float wt;
            };
    struct student s1={1001,24,'M',5.5,60.5};
```

How to initialize value directly in structure.

```
struct emp {
            int eno; char ename[20]; float esal; }
void main() {
              struct emp e = {100,"Kanak",5000 };
              // directly we cal initialize structure
              printf("Emp details");
```

```
    printf("Eno%d\n",e.eno);
    printf("Ename%s\n",e.name);
    printf("Esal%f\n",e.esal);
}
```

Reading and displaying values of structure variable :-

The C compiler will not read or write an entire structure as a single command. It will read or write the members of a structure as follows :-

```
scanf("%d",&s1.rno);
scanf("%d",&s1.age);
scanf("%d",&s1.gender);
scanf("%d",&s1.ht);
scanf("%d",&s1.wt);
```

To display the members of a structure :-

```
printf("%d",s1.rno);
printf("%d",s1.age);
printf("%d",s1.gender);
printf("%d",s1.ht);
printf("%d",s1.wt);
```

Note :- Two structure variable can not be compared. The structure variable can be assigned another structure variable, if both are belongs to same structure, but can not be assigned , if both are belongs different structure.

Structure assignments :-

The information contained in one structure may be assigned to another structure of the same type using a single assignment statement. That is, we do not need to assign the value of each member separately. The following program illustrates structure assignments.

Example :- #include <strdio.h> #include <conio.h>

```
int main(voiod) {
    struct {
        int a; int b;
      } x,y;
    x.a=10; x.b=20; y=x; /* assign one structure to another */
    printf("%d",y.a); printf("%d",y.b);
   return 0;
  } // After assignment y.a will contain 10 & y.b will be value 20.
```

How Structure allocates memory :-

Structurwe always allocates memory in the form of block and block size depednds on the longest data member of the structure. Example :-

```
struct xxx
        {   char a;
            short int b;
         };
int main()
        {
          printf("%d",sizeof(struct xxx));
        };
```

Now, xxx allocates 2 blocks of memory, where each block size of of two bytes because the longest data members of the structure is a short integer, whose size of is two bytes long.

The very first data member is character which is one byte long and senond data member is short int is two bytes long can not be seated in second byte of the first block. So for it a new block memoery is allocated where the data member b is seated.

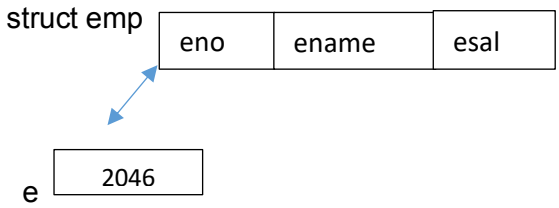
Why structure is used ?

Array	Vrs	Structure
We can store 'n' no of elements but only same data-type or homogeneous elements in an array.		Diffrent type of data can be stored in structure.
Ex:- int arr[5] = {10,20,30,40,50} char stn[5]="flag";		struct emp { int eno; char ename[20]; float esal;);

How to allocate memory in structure variable :- Just declaring structure data-type, it will not allocate memory. Here data-type is struct emp and it is collection of two words & it is user-defiened data-type and primitive data-types are single word like int, float, char etc. When we will declare variable memory is allocated. Example :-

```
main() {
        int a;
        struct emp e;
      }
```

Here **e** is a structure variable & memory allocated to **e** and **e** has members as in **emp**.



Structure variable holds the Base address only. (It is just a random address)

e is called internal pointer variable because it is holding the base address only.

eno – occypies 2 bytes only (2 byte for int) address from 2046 to 2047.

ename – occupies 20 bytes (1 byte for char) addrerss from 2048 to 2067.

esal – occupies 4 bytes (4 byte for float) address from 2068.

Total 26 bytes of memory is allocated.

UNIONS :

It is a user-defiened data-type.

Let us compare structure. We can create user-defiened data-type by which we can store n number of elements of different data-types.

```
Struct  identity {
        Data-type element1;
        Data-type  element2;
        Date-type element3;
};
```

When we are allocating memory in structure all the elements gets independent memory location, say n elements are there then n memory locatioons are created.

```
Union  identity {
        Data-type element1;
        Data-type  element2;
        Date-type element3;
};
```

In case of union, all the elements share same memory location. At a time all the elements can not be processed. Example :-

Struct st { char c; short s; float f; }; Struct st var1; var.c; var.s; var.f;		Struct un { char c; short s; float f; }; Union un var1; / allocating variable var.c; var.s; var.f;
It occupies more memory.		It occupies more memory.

Elements :-

c – 1 byte, s – 2 byte, f- 4 byte length and total 7 bytes will be allocated to variable var1.		Out of variables c, s and f – 4 bytes will be allocated to memory. We can process all the elements of union one after another not all at a ime. All the elements we are declaring in union are sharing the same lomory location and process one element at a time.

Example :-

```
union un {
        Short a; short b;   // variable a oer b are pointing to same memory location
    };
main() {
    union un var;
    var.a=10;
    printf("b.var%d\n",var.b);
    var.b=20;
    printf("a.var%d\n",var.a);
    }
```

Only work with one element of Union.

STRINGS :-

It is one-dimentional character array.

String -> char[ ]; // char is One-dimentional

If we want to process data like name, we use string data-type.

Syntax of array is :- data-type identity [size];

Example of string is :- char name [20]

If size of the array is 20 , maximum we can store 19 characters, because last character is null characher by default.

We can initialize string :-

Example :- char s1[5]="Hari"; // H a r i '\o' Null character or Blank character ASCII value is zero

Array variable always store base address, Hence array variable is called internal pointer variable, because string holds address. OR

Char s1[b]={'H', 'a', 'r', 'i' } ; and %s format specifier is used to read & write string elements.

```
#include <stdio.h>
#include <conio.h>
main() {
    char name[20];
    printf("Enter your name");
    scanf("%s", name);
    printf("%s,Welcome",name);
```

```
        }
//Function to find out length of the string
int stringlength(char[ ]);          // prototype
int main() {
                char str[30];
                int len;
                printf("Enter String");
                gets(str);
                len=stringlength(str);
                printf("Length %d\n", len);
            }
    Int stringlength (char x[ ] )
                                        {
                                          int i=0;
                                          count = 0;
                                          while (x[ i ] != '\0')
                                                                {
                                                                   count ++;    i++;
                                                                 }
                                             return count ;
                                           }
```

STRINGS :-

In C language the group of characters, digits, and symbols enclosed within quotation marks are called string. The string is always declared as character arrays. In other words character arrays are called strings. To manipulate text such as words and sentences normally strings are used.

Every string is terminated with '\0' (NULL) character. The NULL character is a byte with all bits at logic zero. Hence, its decimal value is) (zero). Each character of string occupies 1 byte of memory. The last character is always '\0'. It is not compulsory to write'\0' in string. It is automatically appended at the end of the character array or string. The character of string are store in contiguous memory locations.

One – Dimensional character Arrays:-
1D character arrays are useful to store a single string. For example to store a person name or a city name or any object name.
Example : char s [10];

(a) Declaration :

Strings can be declared like one- dimensional arrays.

Syntax: char string –name [size];

A string variable is any valid C variable name and is always declared as an array.

The size determines the maximum size of string variable.

Example:-

1. Char city [10];
2. Char name [20];
3. Char str [9];

(b) Initialization :

Character array may be initialized when they are declared.

Ex : char str[9] = " I like C"; Which is the same as

char str [9] = {'I' , ' ' , 'I' 'K' 'e', 'c', '\'};

Whenever a string, enclosed in double quotes, is written, c automatically creates an array of characters containing that string, terminated by the '\0' character.

C language allows the alternative notation:

Char msg [ ] = "Hello";
that is always used in practice.

It should be noted that the size of the aggregate 'msg' is six bytes, five for the letters and one for the terminating NULL.

(c) **Reading strings** :

I. **use scanf() with %s format:**
strings my be read by using the &s conversion with the functions scanf() but these are some irksome restrictions. The scanf() terminates reading string, after encounter of white space character.

Example:

```
void main ()
{
        Char x[20];
printf ('Enter a string\n'');
scanf ('%s" , x);
printf ( '' The string is : %S ", X);
}
```

II. Using scanset:
The scanset conversion facility provided by scanf() is useful string input method. This conversion facility allows the programmer to specify the set of characters that are acceptable as part of the string. A scanset conversion consists of a list of acceptable characters enclosed within square brackets.

Example:

```
Void main ()
{
Char str{50};
Printf (Enter a string in lower case : \n");
Scanf ("% [a-z] ", str);
Printf ("The string was : %s |n", str);
}
```

III. Single line input using scanset with^:
The circumflex(^) plays an important role while taking inpur.

For a single-line text inpur, the user presses the <Return> or <Enter> key to terminate the string.
Example: scanf("%[^\n]", str);

IV. Using scanf() with conversion specifier % c:
An alternative method for the input of strings is to sue scanf () with the %c conversion which may have a count associated with it.

Example : scanf ("%10c" , str);

Str[9] = '\0';

V. **Using gets():**
The best approach to string input is to use a library function called tets(). This takes the start address of an area of memory suitable to hold the input as a single parameter.

Emaple : gets (str);

VI. **Using getchar():**

```
While(1)
{
CH=getchar():
If (ch=='\n')
Break;
S[i]= ch;
}
```

//Output: india with some garbage character

⇒ **Strcat ()** : marge the contents of one string to another string

```
Main()
{
        Char x [] =''lakshya";
        Strcat (x, ";");
        Strcat (x, "welcome");
        Printf ("%s",x);
        }
        // output: lakshya;welcome
```

⇒ **Strcmp ()** : compare between two string, returns 0 if both strings are equal otherwise return nonzero.

```
Main()
{
        Char x [] ="lakshya";
        Int k=strcmp (x,"lakshya");
Printf("%d", k);
}
//output: 0
```

Two dimensional character arrays: (or) Array of Strings:

We often use lists of character strings, such as a list of names f students in a class, list of name of employees in an organization, list of places etc. a list of names can be treated as a table of strings and a two-dimensional character array can be used to stre the entire list.
Example : char names[10][30];

i. **Declarations**.

Syntex: char string – name [size][size];
Example: char students[30][15];
Is used to store a list f 30 names, each of length not more than 15 characters.

ii. **Initialization**:
Example: char city [][]={"hyd","delhi","Bombay","madras"};
to assess the name f the ith city in the list we write
city[i-1]

iii. **Reading a two – dimensional character arry**:

Two dimensional character array can be read as follows by using for loop

```
Example :   for (i=0; i<n;i++)
                    {
                       gets (name[i]);
                    }
```

lv. **Writing a two – dimensional character array;**

Two dimensional character array can be displayed as follows by using for loop

```
Example:    for (i=0 ; i < n; I ++)
                    {
                      Puts (name [i])
                     }
```

Array of pointers To Strings

Char *arr [ ] ={“white” , “red”,”green”};

->arr is an array of pointer to strings.

->arrays of pointers contain a number of addresses of strings

->it contains the base addresses of all the strings such as the base address

Of white is stored in arr [0], red in arr[1] , green in arr [2]

(D) **Displaying Strings**:

The corresponding output function for scanf () is printf () . By using this

Function a string can be display with %s format .

Example: 1. Print f (“%s ‘ ,X);

2.print f (“Ravi”)

We can also display a string using the output function puts ( ).

Example: 1. Puts (X)

2. puts (“ Ravi “);

(e) Visiting a string

Char X [ ] =” man is mortal “ ;

‘m’	‘a’	‘n’		’i’	‘s’		‘m’	‘o’	‘r’	‘t’	‘a’	‘l’	‘\0’

```
main ( )
{
char X [ ] = “ man is mortal “ ;
char  * p = X ;
while  (* p ! = ‘ \ 0\ )
{
        print f (“ % c “,*p ) ;
        p ++;
      } }
```

Standard Library Function to Handle Strings

Strlen ( ) : mesearue the length of a string without null character

```
main ( )
{
```

```
        Char X  [ ] = " India " ;
        Print f  ( " %d " 'sizeof ( X ) );
}       // output :  5
```

Sizeof ( ) : measure the length of a string including null character

```
main ( )
{
        Char X  [ ] = " India ";
        Print f ("%d", sizeof ( X ) ) ;
}       // output :   6
```

Strcpy ( ) : copy contents of one string to another string including null

Character

```
main ( )
 (
        Char X [ ] = " India is the best"
        Char  y [ 20 ];
       Strcpy ( y,x );
      Print f (" %s " ,y ) ;
 }
 // output : India is the best
```

Strncpy ( ) : copy ' n ' character from one string to anotrher string with out null Character

```
main ( )
{
        Char x [ ] = " India is the best ",
        Char y [20];
        Strncpy ( y,x,5);
        Print f ("%s",y);
}
 //output : India
```