LECTURE NOTE

OF

COMPUTER

APPLICATION

PREPARED BY
UMAKANTA NANDA

GOVT.POLYTECHNIC,JAJPUR,RAGADI

## Introduction To Computers

- Early day's people use fingers for computing purpose.
- As years go, the computing needs also grew. This leads to the invention of calculators and computers.
- The term computer is derived from the word compute, it means to calculate.

Definition: Computer is an electronic device which accepts data as input process the data and gives the desired result as output.

INPUT ⟶ COMPUTER ⟶ OUTPUT

Data Processing

The activity of processing data using a computer is called data processing.

Data

Capture Data

Manipulate Data

Output Results
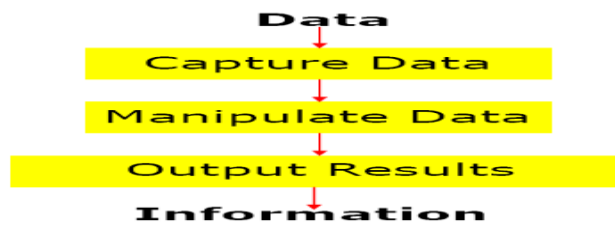
Information

Data: Data is a raw fact or basic fact through which we cannot identify the object.

Information: - The processed data is called information through which we can identify the observer.

Data	Information
I) Data is a raw fact or basic fact through which we cannot identify the object.	i)The processed data is called information through which we can identify the object.
ii) Input may be data or may be information.	ii) Output must be information.

CHARACTERISTICS OF COMPUTER

Speed: The computer process the data extremely fast. (i.e.) in millions of instructions and even more per second. A computer can perform a huge task in few sec where a human being may take hours or even days to complete the same task.

Accuracy: The degree of accuracy of computer is very high and every calculation is performed with the same accuracy. The accuracy level is determined on the basis of design of computer. The errors in computer are due to human and inaccurate data.

Automatic operation: Given a job, computer can work on it automatically without human interventions.

Diligence : A computer is free from tiredness, lack of concentration etc. It can work for hours without creating any error. If millions of calculations are to be performed, a computer will perform every calculation with the same accuracy

Versatility: It means the capacity to perform completely different type of work. For example At a moment a computer can be used to draft a letter. Next moment it can be used print a document or play a music file etc.

Storage : Computers are capable of storing huge amount of data and it can be located and retrieved very quickly. It uses storage devices such as hard disk to store huge amount of data.

No I.Q.: A computer does only what it is programmed to do. It cannot take its own *decision* in this regard.

No Feelings: Computers are devoid of emotions. Their judgement is based on the instructions given to them in the form of programs that are written by human beings.

Applications of computer:
- Business
- Industry
- Home
- Education
- Printing & publishing
- Medical Diagnosis
- Railway
- Banking
- Defence setup
- Entertainment etc.

Terms associated with Computer:

Instructions: instructions describe the programming commands.

Computer program: A computer program is a sequence of instructions written to perform a specified task with a computer.

Hardware: The physical part of the computer is called hardware.

Software: A set of program instruct the hardware to perform a job. Without software hardware cannot work.

Measurement of computer storage:

Bit →Binary digit (i.e. either 0 or 1)

1   nibble = 4 bits
1   byte = 8 bits
1 kilobyte (KB) = 1024 bytes
1megabyte (MB) = 1024 KB
1 gigabyte (GB) = 1024 MB
1 terabyte (TB) = 1024 GB
1 petabyte (PB) = 1024 TB

EVOLUTION OF COMPUTERS

Abacus : The first calculating device called ABACUS was developed by the Egyptian and Chinese people. It consisted of sticks in horizontal positions on which were inserted sets of beads. It has a number of horizontal bars each having ten beads. Horizontal bars represent units, tens, hundreds, etc.

Napier's bones : English mathematician John Napier built a mechanical device for the purpose of multiplication in 1617 A D. The device was known as Napier's bones.

Slide Rule: This machine could perform operations like addition, subtraction, multiplication, and division. It was widely used in Europe in 16th century.

Pascal's Machine: Blaise Pascal developed a mechanical calculating machine called Pascal's calculating machine. It could add and subtract. The machine consisted of wheels, gears and cylinders.

Leibniz's Machine: The Pascal's calculating machine can perform addition and subtraction but it could not perform multiply and division. Leibniz built a mechanical device that could both multiply and divide.

Babbage's Analytical Engine: In the year 1823 that a famous English man Charles Babbage built a mechanical machine to do complex mathematical calculations. It was called *difference engine*. Later he developed a general-purpose calculating machine called *analytical engine*. Charles Babbage is called the *father of computer*.

Hollerith punched card: Herman Hollerith of USA developed punched card Machine to process cenus data. The data was punched in card in coded form. The card is placed on a metallic plate wired drops move the card and makes contact with the plate in punched holes and generate electrical pulse.

ENIAC(Electronic Numeric Integrator And Calculator):
- It has nearly 18000 vacuum tubes.
- Several resisters & capacitors.
- The weight was approximately 5 ton.
- The failure have vacuum tube is very frequent.
- This was the fast calculating Machine.

EDSAC(Electronic Delayed Storage Automatic computer):
It was developed in 1949 and was first digital computer which has stored program concept. In this Machine instruction are stored in memory.

UNIVAC(Universal Automatic Computer):
- It was the first commercial computer. It was developed in 1951.
- It had both numeric and alphabetic data processing.

GENERATION OF COMPUTERS

The term generation indicates the type of technology used in the computer construction. As new technology was emerging, it was being used in the making of computer. The new technology improved the speed, accuracy and storage capacity of the computers. Different technologies have been used for computers in different times

Therefore, computers can be divided into five generations depending upon the technologies used. These are:
First Generation (1942 - 1955)
Second Generation (1956 - 1963)
Third Generation (1964 - 1975)
Fourth Generation (1975-1989)
Fifth Generation (1989-present)

First Generation (1942 - 1955)
- The time period of this generation is 1940 to 1956.
- The principal component used in the circuits of this generation computers are **Vacuum tubes.**
- Electromagnetic relay memory and Punched cards are used for storage.
- The language used by these computers was machine language.
- Examples of this generation computers are : ENIAC, EDVAC, EDSAC, UNIVAC I, ▢IBM 701.
- These computers were the fastest calculating devices of their time. They could perform computations in milliseconds.
- The demerits of this generation computers were bulky size, slow speed, high heat generation, high rate of failures, high power consumption etc.

Second Generation (1956 - 1963)

- The time period of this generation is 1956 to 1963.
- The principal component used in the circuits of this generation computers are **transistors.**
- Magnetic cores, memory, Magnetic tapes, Disks are used for storage.
- The high level languages like FORTRAN, COBOL, BASIC etc. were used as the languages by the computer
- Examples of this generation computers are : IBM 1401, IBM 7030 and CDC 1604etc.
- The **merits** of this generation computers were Faster, smaller, more reliable and easier to program than previous generation systems etc.
- The **demerits** of this generation computers was difficult and costly Commercial production.

Third Generation (1964 - 1975)
- The time period of this generation is 1964 to 1975.
- The principal component used in the circuits of this generation computers are **ICs(integrated circuits) with SSI and MSI technologies.**
- Larger magnetic cores memory, Larger capacity disks and magnetic tapes are used for storage.
- High level language like FORTRAN, COBOL, PASCAL, BASIC, ALGOL etc. were used during this generate.
- Examples of this generation computers are : IBM 360/370, PDP-8, PDP-11
 CDC 6600 etc.
- The **merits** of this generation computers were Faster, smaller, more reliable, easier and cheaper to produce Commercially, easier to use, and easier to upgrade than previous generation systems, Scientific, commercial and interactive online applications etc.
- The **demerits** of this generation computers were Air-conditioning required, Highly sophisticated technology required for the manufacturer chips .

Fourth Generation (1975-1989)

- The time period of this generation is 1975 to 1989.
- **Microprocessors** were introduced as CPU– Complete processors and large section of main memory could be implemented in a single chip
- The principal components used in the circuits of this generation computers are **ICs with VLSI technologies.**
- Larger capacity hard disks as in-built secondary storage, Magnetic tapes and floppy disks as portable storage media.
- Operating systems for PCs with GUI and multiple windows on a single terminal screen, Multiprocessing OS with concurrent programming
Languages, UNIX operating system with C programming language,   Object-oriented design and programming, Network-based, and supercomputing applications
- Examples of this generation computers are : IBM PC and its clones, Apple II, TRS-80, VAX 9000, CRAY-1, CRAY-2, CRAY-X/MP etc.
- The **merits** of this generation computers were Small, affordable, reliable, and easy to use PCs, More powerful and reliable mainframe systems and supercomputers, Totally general purpose machines, Easier to produce commercially, Easier to upgrade, Rapid software development possible etc.
- The **demerits** of this generation computers was Highly sophisticated technology required for the manufacture of microprocessor   chips.

Fifth Generation (1989-onwards)

- The time period of this generation is 1989-onwards.
- The principal components used in the circuits of this generation computers are **ICs with ULSI technologies.**
- The fifth generation computers are based on Artificial Intelligence (AI).
- Larger capacity main memory, hard disks with RAID support, Optical disks as portable read-only storage media, Notebooks, powerful desktop PCs and workstations, Powerful servers,
- Examples of this generation computers are : IBM notebooks, Pentium PCs, SUN Workstations, IBM SP/2, SGI Origin 2000, PARAM 10000.
- The **merits** of this generation computers were Portable computers, Powerful, cheaper, reliable, and easier to use desktop machines, Powerful Supercomputers, Totally general purpose machines, Easier to produce commercially, easier to upgrade, Rapid software development possible etc.
- The **demerits** of this generation computers was Highly sophisticated technology required for the manufacture of microprocessor chips.

CLASSIFICATION OF COMPUTER

There are three different types of computers according to the principles of operation. Those three types of computers are
- Analog Computers
- Digital Computers
- Hybrid Computers

1. Analog Computers

- An analog computer is a form of computer that uses the continuously changeable aspects of physical phenomena such as electrical, mechanical or hydraulic quantities.
- These are mostly used in industrial and process control system.
- No. of equipments that are used in less, so cost is less.
- Accuracy factor is less in this computer.

2.Digital Computers

- A computer that performs calculations and logical operations with quantities represented as digits, usually in the binary number system.
- Digital computer counts and answer the questions by the method of "HOW Many". The input data is represented by a number. These are used for the logical and arithmetic operations.
- All commonly used computers are digital.
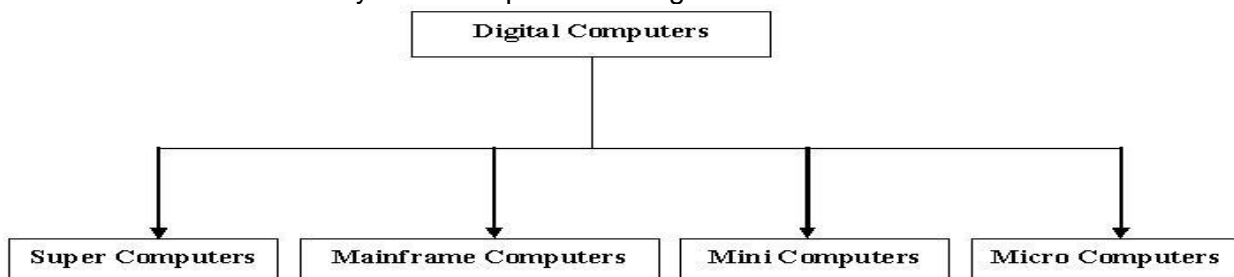
a)Super Computer

- Super computers are mostly used for complex scientific and statically application of programs.
- Super computer are mainly used for weather forecasting, air flow control traffic, defence system, railway or airlines tickets.
- Super computer have high capacity memory and high processing power.
- The cost of one super computer is millions.

- Super computers are non-portable and large in size.
- These are very sensitive to temperature humidity and dust.
- Very high qualified staff is required to operate super computers.
- Examples: CRAY, PARAM etc.

b)Main frame Computer
- These computers are smaller in size compare to super computers.
- These computers have less memory, less   processing speed as compare to super computers.
- The cost of main frame computers is very high.
- These are also non-portable.
- To execute main frame computers high speed processor are connected with high sophisticated technology is required for manufacturing and assembling of computers.

c)Mini computers
- Mini computers are smaller than main frame computers.
- These computers are slower than mainframe computers and super computers.
- The cost of mini computers is in thousands. These are portable computers.
- We cannot connect all the hardware devices with mini computers.
- We cannot execute all the languages and software on the mini computers.

d)Micro Computers
- Micro computers are smaller than mini computers.
- These have slow speed than mini computers.
- The cost of micro computers is also very low.
- Micro computers can execute limited computer languages and software.
- These have low memory as compare to mini computers.
- E.g.: Desktop, Laptop, Tablet PC, Smart Phones.

3.Hybrid computers
- Hybrid computers are computers with combined features of both digital and analog type.
- This type of computer operates by counting as well as by measuring. In other words, the output can be either in the form of numbers or required units of measurement, e.g., an analog device measures patient's heart beat (ECG).
- These measures will be converted into digital form and a digital device checks for any abnormality.
- Further, we can also input digital data like your marks and get digital results like the result of your class.
- Another example is a Modem. (Which converts the digital signals into analog, carry it along the line and at the receiving end again changes it back into digital signal).

| Personal Computers | Workstations | Minicomputers | Mainframes | Supercomputers |

Least powerful ———————————————————————→ Most powerful

Functional Block Diagram of Digital Computer/ Von-Neumann Architecture:

The computer system hardware comprises of three main components:—
1. Input Unit
2. Output Unit
3. Central Processing Unit (CPU), and
4. Memory Unit.

The I/O unit consists of the input unit and the output unit. CPU performs calculations and processing on the input data, to generate the output. The memory unit is used to store the data, the instructions and the output information.
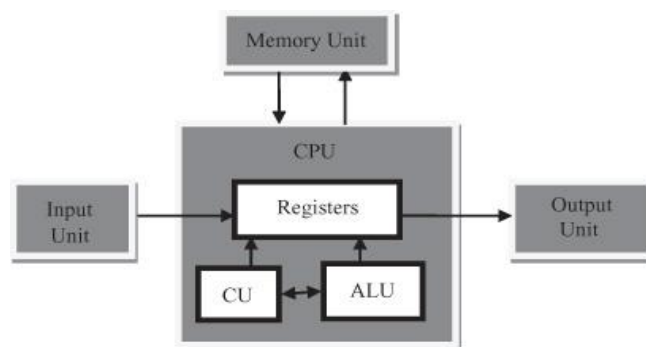
Fig: Block diagram of Digital Computer.

- *Input Unit*:
  — The user interacts with the computer via the input and output unit.
  — The Input unit accepts data from the user.
  — The Input unit converts the data that it accepts from the user, into a form that is understandable by the computer.
  — The input is provided to the computer using input devices like keyboard, trackball and mouse.

- *Output Unit*:
  — The Output unit provides the processed data i.e. the information to the user.
  — The Output unit provides the output in a form that is understandable by the user.
  — Some of the commonly used output devices are monitor and printer.

- *Central Processing Unit:*
  CPU controls, coordinates and supervises the operations of the computer. It is responsible for processing of the input data. CPU consists of:
  i.   Arithmetic Logic Unit (ALU)
  ii. Control Unit (CU)
  iii. Register Unit.

  Arithmetic Logic Unit:
  — Performs all the arithmetic and logic operations on the input data.
  — Arithmetic operations are addition, subtraction, multiplication, division etc. Also, doing some Logical operations like AND, OR, NOT etc.

  Control Unit:
  It controls the overall operations of the computer i.e.
  — It checks the sequence of execution of instructions, and, controls and coordinates the overall functioning of the units of computer.
  — It also responsible for decoding the instruction given by the user.

  Register Unit:
  — CPU also has a set of *registers* for temporary storage of data, instructions, addresses and intermediate results of calculation.
  — This is some time called as chip memory. It consists of several high speed registers.
  — CPU uses this memory for its work.

- *Memory Unit:*
  Generally, during the processing of data Memory unit stores the data, instructions, intermediate results and output *temporarily*.

— This unit is classified into two types of memory. They are:
  o Primary memory
  o Secondary memory

Primary memory:
— This memory is also called the *main memory or primary memory* of the computer.
— The input data that is to be processed is brought into the main memory before processing.
— The instructions required for processing of data and any intermediate results are also stored in the main memory.
— The output is stored in memory before being transferred to the output device.
— CPU can work with the information stored in the main memory.
— The capacity of this memory is limited and they are volatile in nature.
— ROM and RAM are examples of primary memory.

Secondary Memory:
— Another kind of storage unit is also referred to as the *secondary memory* of the computer.
— The data, the programs and the output are stored *permanently* in the storage unit of the computer.
— The capacity of this memory is huge.
— Magnetic disks, optical disks and magnetic tapes are examples of secondary memory.

Memory Classification/ Memory Hierarchy

Memory in computer is used for following purpose:
- Stores programs and data during execution.
- Stores programs results of execution.
- Stores programs and data for future reference.

The memory is classified in hierarchical structure as follows:
  I. Register Memory.
  II. Cache memory.
  III. Main memory/ Primary memory.
  IV. Auxiliary memory/ Secondary memory.

In hierarchical structure of memory the speed of access increases up the pyramid and the size increases down the pyramid.
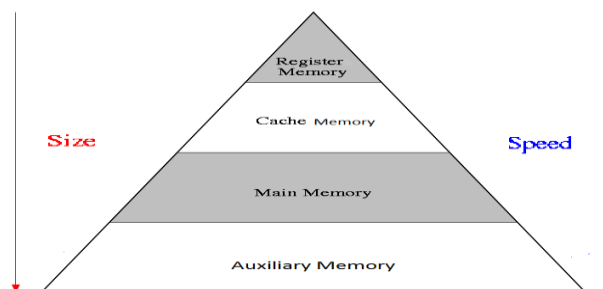
I. **Register memory:**
   - Register memories are high speed registers in the CPU, work as a memory for temporary storage of instruction and data.
   - CPU uses registers for the processing of data, the number of registers in a CPU and the size of each register affect the power and speed of a CPU.
   - The more the number of registers and bigger the size of each register, the better it is.

II. **Cache Memory:**
   - Cache memory is a fast memory which is placed in between the CPU and the main memory.
   - Cache memory consists of binary cells and the size of this memory is less than main memory.
   - When the CPU needs an instruction or data during processing, it first looks in the cache memory.
   - If the information is present in the cache, it is called a *cache hit*, and the data or instruction is retrieved from the cache.
   - If the information is not present in cache memory, then it is called a *cache miss* and the information is then retrieved from main memory.

III. **Main Memory/Primary Memory:**
   - Data and programs are stored in this memory during execution.
   - After execution the result is stored in this memory.
   - It is volatile in nature. This means whenever power goes out the content of this memory will be lost.
   - The access speed of this memory is higher than the auxiliary memory.
   - Main memory consists of RAM & ROM.

RAM (RANDOM ACCESS MEMORY):-
   - The access time of this memory to read data from the first location or the last location is same.
   - Both read and write operation is possible in this memory.
   - This is volatile in nature.
   - RAM is of two types:-
     - **a)** SRAM(Static RAM)
     - **b)** DRAM (Dynamic RAM)

a) SRAM (Static RAM)
   - SRAM (static RAM) is random access memory (RAM) that retains data bits in its memory as long as power is being supplied.
   - SRAM does not have to be periodically refreshed.
   - Static RAM provides faster access to data and is more expensive than DRAM.
   - SRAM is used for a computer's cache memory.

b) DRAM (Dynamic RAM)
   - Dynamic random access memory (DRAM) is the most common kind of random access memory for personal computers and workstations.
   - DRAM stores each bit in a storage cell consisting of a capacitor and a transistor.
   - Capacitors tend to lose their charge rather quickly; thus, it need for refreshed.

ROM (READ ONLY MEMORY):
   - In this memory only read operation is possible.
   - The content of this memory is written by the manufacturer.
   - It is non-volatile in nature.

a) PROM (PROGRAMMABLE READ ONLY MEMORY):-
   - In PROM, user can programme the ROM only once.

b) EPROM (ERASEABLE PROGRAMMABLE READ ONLY MEMORY:-
   - In this content can be erased and programmed.
   - Erasing is possible by use of ultra-violet rays.

c) **EEPROM (ELECTRICALLY ERASEABLE PROGRAMMABLE READ ONLY MEMORY):-**
- In this memory erasing the content is possible by applying the high voltage electricity.

IV. **Auxiliary Memory/Secondary Memory:**
- This memory consists of magnetic and optical materials.
- This memory capacity of this memory is high and access speed of this memory is very less.
- It is non-volatile in nature.

EX:- Magnetic disc, Magnetic tapes etc.

Classification of Commonly Used Secondary Storage Devices

Magnetic Tape
- This is commonly used sequential-access secondary storage device.
- Physically, the tape medium is a plastic ribbon, which is usually ½ inch or ¼ inch wide and 50 to 2400 feet long.
- Plastic ribbon is coated with a magnetisable recording material such as iron-oxide or chromium dioxide.
- Data are recorded on the tape in the form of tiny invisible magnetized and non-magnetized spots (representing 1s and 0s) on its coated surface.

Magnetic Disk
- Commonly used direct-access secondary storage device.
- Physically, a magnetic disk is a thin, circular plate/platter made of metal or plastic that is usually coated on both sides with a magnetisable recording material such as iron-oxide.
- Data are recorded on the disk in the form of tiny invisible magnetized and non-magnetized spots (representing 1s and 0s) on the coated surfaces of the disk.
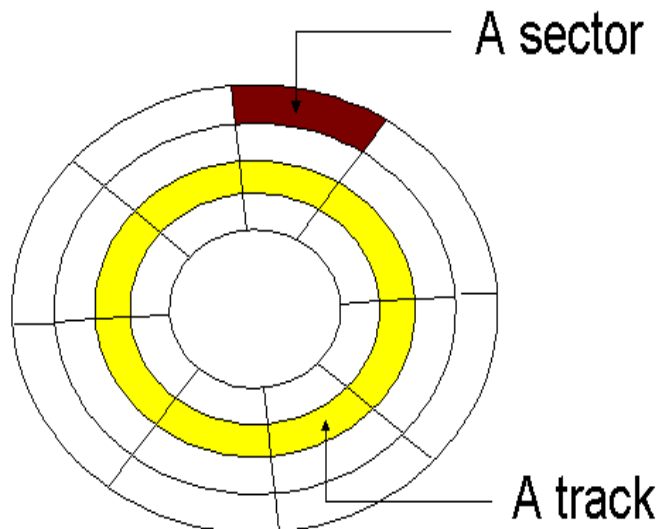
- A disk's surface is divided into a number of invisible concentric circles called tracks.
- The tracks are numbered consecutively from outermost to innermost starting from zero.
- Each track of a disk is subdivided into sectors.
- There are 8 or more sectors per track.
- A sector typically contains 512 bytes.
- Ex-hard disk,floppy disk ,etc.

Optical Disk
- It consists of a circular disk, which is coated with a thin metal or some other material that is highly reflective.
- Laser beam technology is used for recording/reading of data on the disk. because it can store extremely large amounts of data in a limited space.
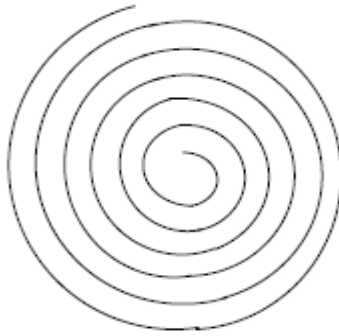
- It has one long spiral track, which starts at the outer edge and spirals inward to the centre
- Track is divided into equal size sectors.
- Ex-CD-ROM,DVD etc.

COMPUTER - INPUT DEVICES:

Following are few of the important input devices which are used in Computer Systems

- Keyboard
- Mouse
- Joy Stick
- Light pen
- Track Ball
- Scanner
- Magnetic Ink Card Reader(MICR)
- Optical Character Reader(OCR)
- Bar Code Reader
- Optical Mark Reader

Keyboard:
- Most common and very popular input device is keyboard. It helps in inputting the data to the computer.
- The layout of the keyboard is like that of traditional typewriter, although there are some additional keys provided for performing some additional functions.
- The keys are following:
  - **Typing Keys:**
    - These keys include the letter keys (A-Z) and digits keys (0-9).
  - **Numeric Keypad:**
    - It is used to enter numeric data or cursor movement. Generally, it consists of a set of 17 keys.
  - **Function Keys:**
    - The twelve functions keys are present on the keyboard. These are arranged in a row along the top of the keyboard.
    - Each function key has unique meaning and is used for some specific purpose.
  - **Control keys:**
    - These keys provides cursor and screen control.
    - It includes four directional arrow key. Control keys also include Home, End, Insert, Delete, Page Up, Page Down, Control(Ctrl), Alternate(Alt), Escape(Esc).
  - **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:
- It is a device that controls the movement of the cursor on a monitor.
- A mouse will have 2 buttons on its top. The left button is the most frequently used button.
- There will be a wheel between the left and right buttons. This wheel enables us to smoothly scroll through screens of information.
- As we move the mouse, the pointer on the monitor moves in the same direction.
- Optical mouse is another advanced pointing device that uses a light emitting component instead of the mouse ball.
- Mouse cannot be used for entering the data. It is only useful to select the options on the screen.
- **Advantages:**
  — Easy to use
  — Not very expensive
  — Moves the cursor faster than the arrow keys of keyboard.

Joystick:
- Joystick is also a pointing device 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.

Light Pen:
- Light pen is a pointing device which is similar to a pen.
- 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 light pen's tip 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.

Scanner:
- Scanner is an input device which works more like a photocopy machine.
- It is used when some information is available on a paper and it is to be transferred to the hard disc of the computer for further manipulation.
- Scanner captures images from the source which are then converted into the digital form that can be stored on the disc.
- These images can be edited before they are printed.

Magnetic Ink Card Reader (MICR):
- MICR input device is generally used in banks because of a large number of cheques to be processed every day.
- 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.
- This reading process is called Magnetic Ink Character Recognition (MICR).
- The main advantages of MICR are that it is fast and less error prone.

Optical Character Reader (OCR):
- OCR is an input device used to read a printed text.
- It scans text optically character by character, converts them into a machine readable code and stores the text on the system memory.

Bar Code Readers:
- Bar Code Reader is a device used for reading bar coded data (data in form of light and dark lines).
- 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 to which bar code reader is connected.

Optical Mark Reader (OMR):
- OMR is a special type of optical scanner.
- It 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.

COMPUTER - OUTPUT DEVICES

Following are few of the important output devices which are used in Computer Systems
- Monitors
- Graphic Plotter
- Printer

Monitors:
- Monitor commonly called as Visual Display Unit (VDU) is the main output device of a computer.
- It forms images from tiny dots, called pixels, which are arranged in a rectangular form.
- The sharpness of the image depends upon the no. of the pixels.
- There are two kinds of viewing screen used for monitors.
    — Cathode-Ray Tube (CRT)
    — Flat- Panel Display

➢ **Cathode-Ray Tube (CRT) Monitor:**
    — In the CRT display is made up of small picture elements called pixels for short.
    — The smaller the pixels, the better the image clarity, or resolution.
    — A finite number of characters can be displayed on a screen at once.
    — **Disadvantage of CRT:**
        ▪ Large in Size
        ▪ High Power consumption

➢ **Flat-Panel Display Monitor:**
    — The flat-panel display refers to a class of video devices that have reduced volume, weight and power requirement compare to the CRT.
    — You can hang them on walls or wear them on your wrists.
    — Current uses for flat-panel displays include calculators, videogames, monitors, laptop computer, graphics display.
    — The flat-panel display are divided into two categories:
        ▪ Emissive Displays
        ▪ Non-Emissive Displays

    ➢ **Emissive Displays:**
        — The emissive displays are devices that convert electrical energy into light.
        — **Examples:** 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**: LCD (Liquid-Crystal Device).

Printer:
- Printer is the most important output device, which is used to print information on paper.
- There are two types of printers
    — Impact Printers
    — Non-Impact Printers

- **Impact Printers:**
  - The printers that print the characters by striking against the ribbon and onto the paper are called impact printers.
  - Characteristics of Impact Printers are following
    - Very low consumable costs
    - Impact printers are 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
    - Line printers
- **Character Printers:**
  - Character Printers are printers which print one character at a time.
  - These are of further two types
    - Dot Matrix Printer (DMP)
    - Daisy Wheel

  - ❖ **Dot Matrix Printer:**
    - In the market one of the most popular printer is Dot Matrix Printer because of their ease of printing features and economical price.
    - Each character printed is in form of pattern of Dot's and head consists of a Matrix of Pins of size (5*7, 7*9, 9*7 or 9*9) which comes out to form a character that is why it is called Dot Matrix Printer.
    - **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 send here and there with very nice quality representation.
    - **Advantages:**
      - More reliable than DMP's
      - Better quality
      - The fonts of character can be easily changed.
    - **Disadvantages:**
      - Slower than DMP's
      - Noisy
      - More expensive than DMP's
- **Line Printers:**
  - Line printers are printers which print one line at a time.
  - These are of further two types:
    - Drum Printer
    - Chain Printer
  - ❖ **Drum Printer:**
    - This printer is like a drum in shape so it 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 characters sets are available in market 48 character set, 64 and 96 characters set.

— One rotation of drum prints one line. Drum Printers are fast in speed and speed in between 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 called Chain Printers.

— A standard character set may have 48, 64, 96 characters.

— **Advantages:**
  - Character fonts can easily be changed.
  - Different languages can be used with the same printer.

— **Disadvantages:**
  - Noisy
  - Do not have the ability to print any shape of characters.

➢ **Non-Impact Printers:**

— The printers that print the characters without striking against the ribbon and onto the paper are called Non-impact Printers.

— These printers print a complete page at a time, also called as Page Printers.

— These printers are of two types:
  - Laser Printers
  - 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 produces 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.

— **Disadvantage:**
  - 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 compare to laser printer

UNIT-II
Computer Software

Software:
- Software is a collection of instructions that enables a user to interact with the computer or have the computer perform specific tasks for them.
- The set of instructions are commonly known as a program, without which computers cannot do any operations.
- Software is classified broadly in to two groups :
    1. System software.
    2. Application software.
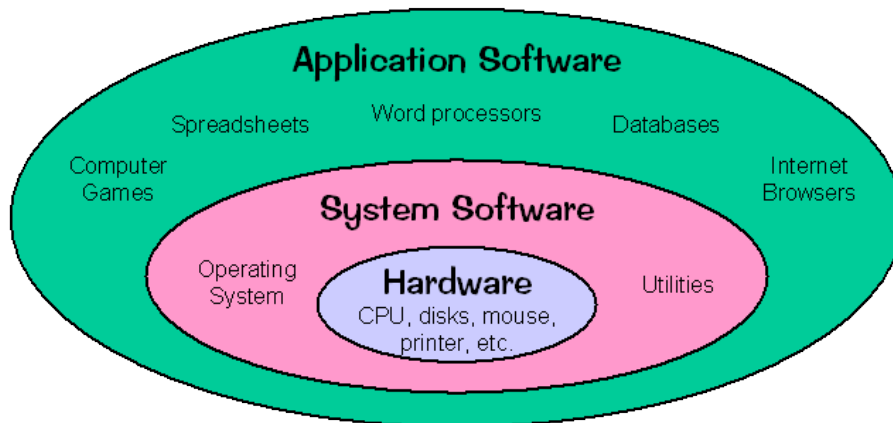
1. System Software
- System software is responsible for controlling, integrating, and managing the individual hardware components of a computer system.
- Systems software programs support application software by directing the basic functions of the computer.
- Examples: Operating System, device driver, System utilities, Compiler, interpreter etc.

2. Application software.
- Application software consists of instructions that direct a computer system to perform specific information processing activities and that provide functionality for users.
- Examples: Spreadsheet, Word processor, data base etc.

INTRODUCTION OF OPERATING SYSTEM:

- **Definition:** Operating System is a system s/w that acts as an intermediary between the user of a computer and computer h/w.
- **Purpose:** provide an environment in which a user can execute program. An OS is an important part of almost every Computer System.
- It controls the internal activities of the comp. h/w and provides the user interface.

COMPONENTS OF OS:
In general we can divide a computer system into the following four components.
- Hardware
- Operating system
- Application programs
- Users

ABSRACT VIEW OF THE COMPONENTS OF A COMPUTER SYSTEM
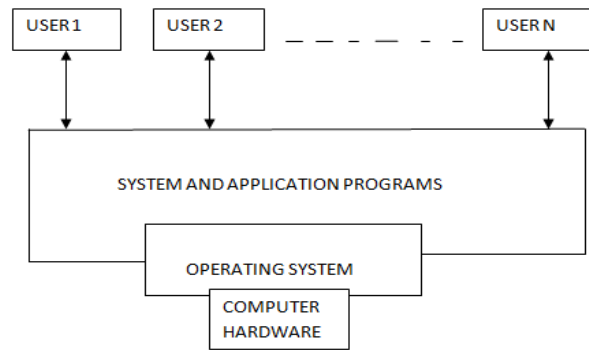
As we can see in the above figure the user interacts with the application programs.

- H/W resources basically include I/O devices, primary memory, secondary memory (hard disk, floppy disk etc.) and the microprocessor.
- The application programs do not access these h/w resources directly.
- So the OS is required to access and use these resources.
- The application programs are programmed in such a way that they can easily communicate with the resources.
- An OS is the first program that is loaded into the computers main memory, when a computer is switched on.
- **Examples:** OS are windows 9x (95, 98,), Linux, Unix, Windows XP, Vista, 7, 8 etc.

NEEDS AND SERVICES OF OS/ FUNCTIONS OF OS:

OS performs a number of functions for the computer system that are follows:

1) **It acts as an Command Interpreter:**
- Generally the CPU cannot understand the commands given by user. It is the function of OS to translate this command (human understandable) into m/c understandable instructions.
- After the execution of instructions by CPU, it retranslates the o/p back into a human understandable language.

2) **It acts as the Resource Manager:**
- Every process requires resource for accomplish its task. It is the job of OS to allocate these resources to corresponding process .
- E.g. Suppose Process A, B requires CPU, Printer respectively. The job of OS allocates a resource such as CPU to program A and printer to program B.

3) **Memory Management:**
- It keeps track of the resources (memory), what part of memory is in use and by whom, which part of the memory is not in use.
- Decides which processes are to be loaded when memory space is available.
- Allocation and de allocation of memory.

4) **Process Management:**
- A process (task) is an instance of a program in execution. A program is just a passive entity, but a process is an active entity.
- To accomplish its task, a process needs certain resources like CPU time, memory, files and I/O devices.
- The OS is responsible for allocate these resources to process either at the time of creation or when it is executing.

5) **I/O device Management:**
- Manages the peripheral devices.
- Consist of policies and procedures for handling I/O devices.
- Allocation and de-allocation of resources.

6) File Management:

- A file is a collection of related information or record defined by the user.
- The OS is responsible for various activities of file management are
  - Creation and deletion of files and directories.
  - Manipulation of files and directories.

7) Protection/Security Management:

- If a computer system has a multiple processor then the various processes must be protected of one another's activities.
- Protect resources and information against destruction & unauthorized access.

8) Error detection and Recovery:

- An error is one part of the system may cause malfunctioning of the complete system. To avoid such a situation the OS constantly monitors the system for detecting the errors.
- The OS should detect such errors and handles them.

CLASSIFICATION / TYPES OF OS:

OS are classified into different categories according to their different features. The following section will discuss the classification of OS.

Single user OS:

- In a single user OS a single user can access the computer at a particular time.
- The single user os is divided into the following types.
  - Single user, single tasking os
  - Single user , multitasking os

Single user, single tasking os:

- In a single user, single tasking OS, There is a single user to execute a program at a particular system.
- Example – MS-DOS

Single user, multitasking os:

- In single user, multitasking OS a single user can execute multiple programs.
- Example – A user can program different programs such as making calculations in excel sheet, printing a word document & downloading into the file from internet at the same time.

Advantage:

- The CPU has to handle only one application program at a time so that process management is easy in this environment.

  o **Disadvantage:**

- As the OS is handling one application at a time most of the CPU time is wasted.

Multi user OS:

- In a multi-user os, multiple numbers of users can access different resources of a computer at a time.
- Ex: UNIX, Window 2000

Batch Operating System

- In the Batch Processing similar types of jobs are grouped together and they are stored on the card and that card will be submit to the system for the Processing. The system then performs all the operations on the instructions one by one.
- In a batch processing OS interaction between the user and processor is limited or there is no interaction at all during the execution of work.
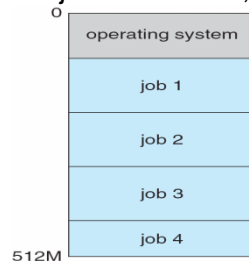
Advantage:

- It is simple to implement.

Disadvantage:

- Lack of interaction between user and the program.

Multiprogramming OS:
- In this type of OS, more than one process will reside into main memory.
- The OS picks and begins to execute one of the jobs in the memory. Eventually, if the jobs have to wait for some task, the OS simply switches to another job and executes it. When the first job finishes waiting and gets the CPU back.
- As long as there is always some job to execute, the CPU will never be idle.

Advantage:
- CPU utilization is more i.e. the most of the time the CPU is busy.

Time sharing OS:
- It allows many users to share the computer simultaneously.
- It gives illusion that each user has his own machine.
- Based on time slicing ,it divides CPU time among the users.
- Users interact (through mouse, keyboard etc..) with computers (editors, debuggers etc..).

Advantage:
- CPU utilization is more i.e. the most of the time the CPU is busy.

Disadvantage:
- The OS is more complex due to memory management, Disk management etc.

Multitasking OS:
- An operating system that is capable of allowing multiple processes to run at the same time.
- We know that processor handles only one application at a particular time, but here, OS is capable of switching the processor between the processes effectively.
- Example: Windows OS.

Multiprocessing OS:
- An operating system capable of supporting and utilizing more than one computer processor.
- More than one processor sharing the common computer bus, and memory and peripheral devices.
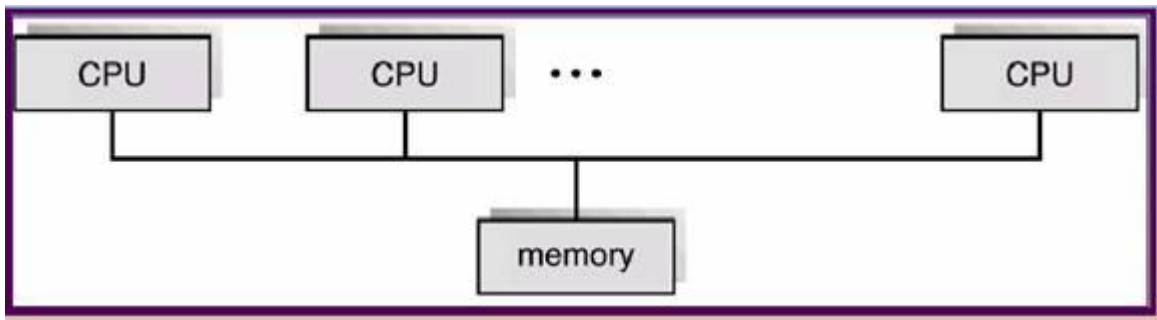
Real time OS:
- RTOS is a special purpose Operating System, used when there are rigid time requirements on the operation of a processor or the flow of data.
- It is of two type.
    1. **Hard real-time system**.
    2. **Soft real-time system**

MS-DOS(Micro-soft disk operating system) :
It is used as operating system in IBM PC compatible machine.

Features of DOS :
- It is a single user operating system
- It works on micro computer only.
- It supports batch processing concept.
- It supports 640KB or higher memory.
- It maintains hierarchical file system organization.
- It is easy to use.
- It provides command line interface to the user which means that the user will type the command to perform the job.
- It supports varieties of software.

UNIX :
It is a multi-user, multi-tasking operating system. It is developed Bell lab. By Thomson and Rithie.
Features of UNIX:
- Unix is a time –sharing operating system.
- Simple Unix command can be combined to form more complex command through pipe and redirection.
- Unix is designed by using C language.
- Unix is portable. It can run on any different machine.
- It supports virtual memory concept.

windows
Windows is an operating system produced by Microsoft for use on personal computers, including home and business desktops, laptops, and media centers.

Features of windows:
- It is a multiuser multi programming operating system.
- It supports GUI (graphical user interface).
- The role of keyboard is reduced because everything is displayed on the monitor screen and user has to select the option required .
- It supports multitasking environment.
- It helps us to connect wide area network via internet.
- It also supports audio, video, multimedia.
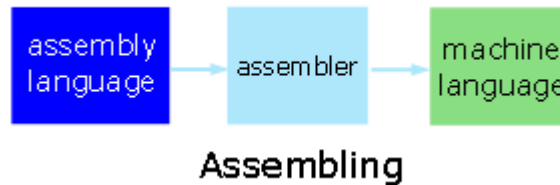
Programming Languages:
- The language used in the communication of instruction to a computer is known as programming language.
- An artificial language used to write instructions that can be translated into machine language and then executed by a computer.
- Computer language is divided in to following types:
    1. Machine-level languages.
    2. Assembly languages.
    3. High-level languages.
- 1.Machine-level languages:
- This is also khown as low level language.
- The programs are written in the form of string of 0s or 1s.
- *It is Machine dependent.*
- *It does not require any translator.*

2.Assembly languages:

- In an assembly language, mnemonics are used to represent operations to be performed by the computer.
- Strings of characters to represent addresses of locations in the computer's memory where the operands will be stored.
- *It is Machine dependent.*

Assembler

- A *assembler* is a translator which translates a program written in assembly language to equivalent Machine-level languages programme , which can be executed by a computer.


Assembling

3.High-level languages:

- A **high-level language** is computer programming language which is English like language follows some syntax.
- It is not limited by the computer, designed for a specific job, and is easier to understand.
- Example**s**: FORTARN, C, C, C++ etc
- Translator like compiler or interpreter is required to translates it to equivalent machine level language programme.

Compiler:

- A compiler is a translator which translates a program written in High-level language to equivalent Machine-level languages programme at a time.
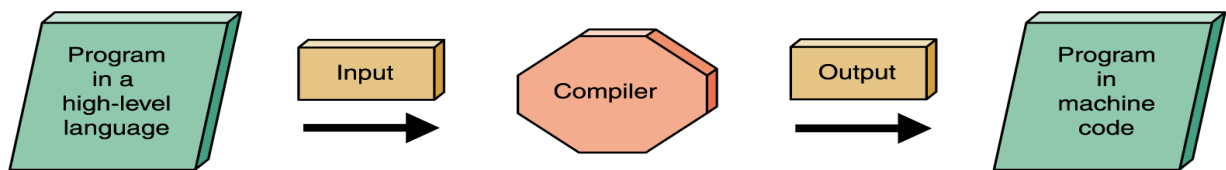
Interpreter

- A interpreter is a translator which translates a program written in High-level language to equivalent Machine-level languages programme line by line.

Difference between compiler & interpreter:

Compiler	interpreter
1. A compiler is a translator which translates a program written in High-level language to equivalent Machine-level languages programme at a time.	**1**.A interpreter is a translator which translates a program written in High-level language to equivalent Machine-level languages programme line by line.
2.Requires more main memory.	**2**. Requires less main memory.
3. if error is detected nothing is executed.	**3**. If error is encountered execution stops but previous statements are already Executed
4. Execution time is less**.**	**4**. Execution time is more.

Computer Virus

- Virus stands for Vital Information Resources Under Seize.
- A Computer Virus is a program that is designed to spread from one computer to another and may corrupt data on your computer, use your e-mail program to spread itself to other computers, or delete data on your hard disk.
- Viruses are most easily spread by attachments in e-mails and can be disguised as attachments of word documents, images, greeting cards, or audio and video files.
- A virus also spreads through downloads on the Internet. They can be hidden in software or other files or programs you download.

Types of virus

1. Boot Sector Virus
   - Infects the boot or MBR of diskettes and hard drives through the sharing of infected disks and pirated software applications.
   - Once your hard drive is infected all diskettes that you use in your computer will be infected.
2. Program Virus
   - Becomes active when the program file (usually with extensions .BIN, .COM, .EXE, .OVL, .DRV) carrying the virus is opened.
   - It then makes copies of itself and will infect other programs on the compute.r
3. Multipartite Virus
   - Hybrid of a Boot Sector and Program viruses.
   - It infects program files and when the infected program is active it will affect the boot record.
4. Stealth Virus
   - Disguises itself to prevent from being detected by antivirus software.
   - It alters its file size or conceals itself in memory.
5. Polymorphic Virus
   - Act like a chameleon, changing its virus signature (binary pattern) every time it multiples and infects a new file.
6. Macro Virus
   - Programmed as a macro embedded in a document, usually found in Microsoft Word and Excel.
   - Once it gets in to your computer, every document you produce will become infected.

Detection or Symptoms of a computer virus :

Even though your computer is infected, you may not realize that you have introduced a virus until you notice something is not quite right. Here are a few things to keep in mind that may indicate that your computer is infected: -

- Your computer appears to be running slower than normal.
- Your printer does not work.
- Unusual Error messages appear regularly.
- Your computer crashes regularly.

- Your computer restarts by itself regularly.
- Drives on your computer appear inaccessible.
- Applications on the computer do not work correctly.
- You see distorted menus and dialog boxes.
- There is a double extension on an attachment that you recently opened, such as a .jpg, .vbs, .gif, or .exe. extension.
- An antivirus program is disabled for no reason. Additionally, the antivirus program cannot be restarted.
- An antivirus program cannot be installed on the computer, or the antivirus program will not run.
- New icons appear on the desktop that you did not put there, or the icons are not associated with any recently installed programs.
- Strange sounds or music plays from the speakers unexpectedly.
- A program disappears from the computer even though you did not intentionally remove the program.

Prevention of computer viruses:

To prevent your computer against viruses, follow these steps:

- **Keep up-to-date antivirus software** running at all times on your computer. Anti Virus software is designed to protect your computer from known viruses. Unfortunately new viruses are created daily so they need to be kept up-to-date. These two products are popular choices Norton AntiVirus 2005 Special Edition, AVG Antivirus Free Edition

- **Use a Firewall** - it's not difficult to install and will prevent hackers from breaking into your computer. Some good firewalls available are Windows XP, Windows Vista , Zone Alarm and Outpost.

- Never open an email attachment from an unknown supply particularly if the attachment has an .exe extension.

- Be careful of opening email attachments from someone you know as they may not know that it contains a virus.

- Keep informed of recent threats by doing research on the Internet.
- Download from reputable sites and avoid downloading files you can't be sure are safe. These may include freeware, screensavers, games and any other executable files.

- **Backup:** No anti virus product can guarantee that a virus does not exist on your computer, as there are continually new ones out there. To guard against disasters, make sure you have an up-to-date backup of your computer and store backups in a separate location.

UNIT-3
COMPUTER NETWORK AND INTERNET

Data communication: Data communications are the exchange of data between two devices via some form of transmission medium such as a wire cable.

Components of data communication
1. *Sender*
2. *Receiver.*
3. *Medium*
4. *Message*
5. Protocol

Protocol: Protocol is the set of rule which governs data communication between computers in a network.

COMPUTER NETWORKS
A computer network is a set of devices (often referred to as *nodes)* connected by communication links. A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

Network Criteria
A network must be able to meet a certain number of criteria. The most important of these are performance, reliability, and security.
1. **Performance:** Performance can be measured in many ways, including Transit and response time.
   - Transit time is the amount of time required for a message to travel from one device to another.
   - Response time is the elapsed time between an inquiry and a response.
   - The performance of a network depends upon number of users, type of transmission medium, capabilities of hardware, efficiency of software.
2. **Reliability:** Network reliability is measured by the frequency of failure, the time it takes a link to recover from a failure, and the network's robustness in catastrophe.
3. **Security:** Network security issues include protecting data from unauthorized access, protecting data from damage.

Connecting Media/Transmission Media:

Different types of of Transmission Media are:
1.**Conducted or guided media:** It use a conductor such as a wire or a fiber optic cable to move the signal from sender to receiver
2.**Wireless or unguided media:** It use radio waves of different frequencies and do not need a wire or cable conductor to transmit signals
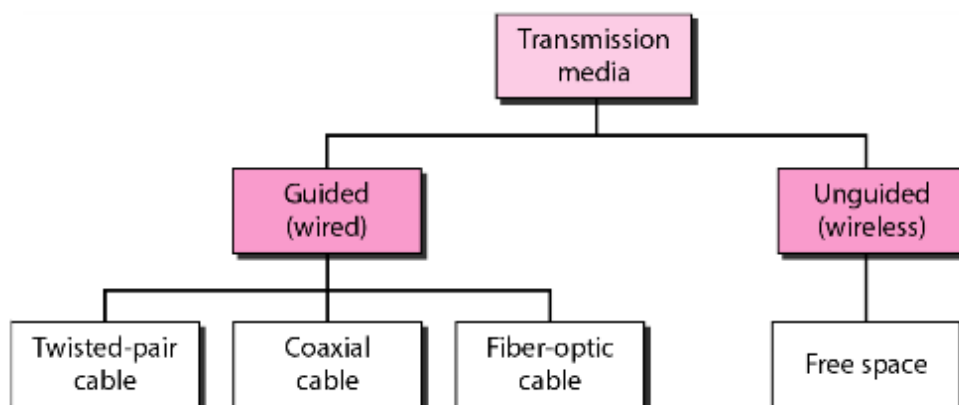
Twisted-Pair Cable:

- A twisted pair consists of two conductors (normally copper), each with its own plastic insulation, twisted together.
- One of the wires is used to carry signals to the receiver, and the other is used only as a ground reference. The receiver uses the difference between the two.
- It is two types:
1. Unshielded twisted Pair Cable (UTP)
2. Shielded Twisted-Pair Cable   (STP)

Coaxial Cable

- Coaxial cable (or *coax)* carries signals of higher frequency ranges than those in twisted pair cable.
- coax has a central core conductor of solid or stranded wire (usually copper) enclosed in an insulating sheath, which is, in turn, encased in an outer   conductor of metal foil, braid, or a combination of the two.
- The outer metallic wrapping serves both as a shield against noise and as the second conductor, which completes the circuit.
- This outer conductor is also enclosed in an insulating sheath, and the whole cable is protected   by a   plastic cover.

Fiber optics cable:

- A fiber-optic cable is made of glass or plastic and transmits signals in the form of light.
- Optical fibers use reflection to guide light through a channel.
- A glass or plastic core is surrounded by a cladding of less dense glass or plastic. The difference in density of the   two materials must be such that a beam of light moving through the core is reflected off the cladding instead of being refracted into it.

Data Transmission mode

- It is the direction of the flow of data between two communicating devices.
- Transmission mode is divided in to three types:
    1. Simplex transmission mode
    2. Half duplex transmission mode
    3. Full Duplex transmission mode/ Duplex transmission mode

1. Simplex transmission mode:
  - In simplex data can only travel in one direction.
  - It used if only one direction is necessary.
  - e.g. Television picture which is passed to a television receiver aerial, but there is no way to send data in the other direction.

2. Half duplex transmission mode:
- In half duplex mode, data can pass in both directions, but only in one direction at a time.
- Used if both directions are necessary but not at the same time.
- E.g. CB radio system in which each handset can either be set to receive mode or send mode.
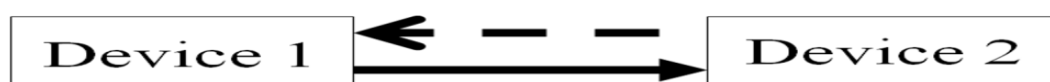
3. Full Duplex transmission mode:
- In this mode, data can pass in both directions at the same time.
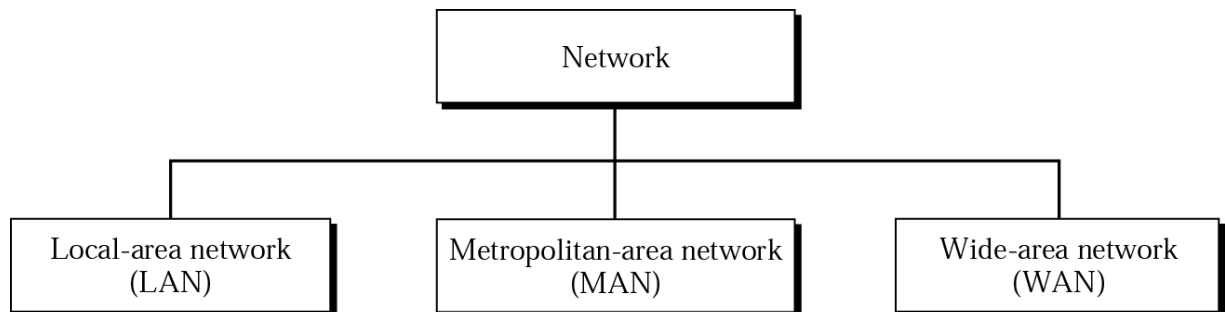- Used if both directions are necessary at the same time.
- E.g. A telephone conversation as both users can speak and be heard at the same time.

Categories of networks

Network divided in to three primary categories: LAN, MAN, WAN. In to which category a network falls is determined by its Size, Ownership, Distance it covers, and Physical architecture.

1. LOCAL-AREA NETWORK(LAN) :
- LAN is usually Privately owned and Links devices in single office, building or campus.
- LAN size is Limited to few kilometres.
- LANs are designed to allow resources (i.e. hardware or software) to be shared between PCs and workstations.
- LAN will use a single transmission media.
- The most common LAN Topologies are Ring, bus, star.

2.METROPOLITAN-AREA NETWORK (MAN):
- A MAN is designed to extend over an entire city.
- It may be single network such as cable television network, or it may be a means of connecting number of LANs in to a larger networks.
- A MAN be wholly Owned and operated by a private company, or it may be a Service provider by Public company such as a local telephone company.

3. WIDE-AREA NETWORK(WAN):
WAN provides long-transmission of data, voice, image and video information over large geographic areas that may comprise a country, a continent or even the whole world.
WAN that is wholly owned and used by a single company is often referred to as an enterprise network.
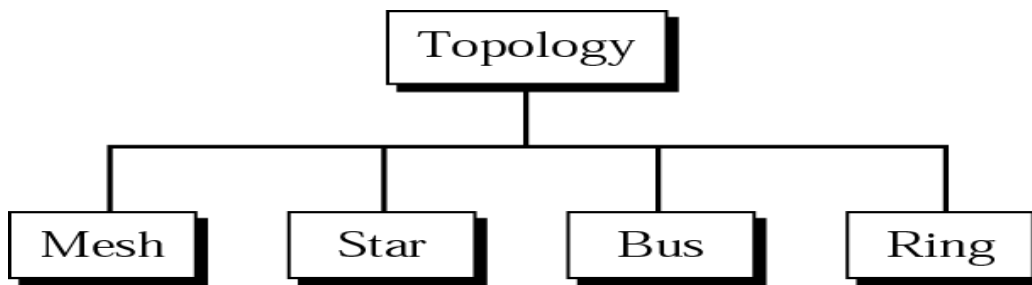
NETWORK STUCTURES

TOPOLOGY:
1.Topology refers to the way in which a network is laid out physically.
2. The topology of a network is the geometric representation of the relationship of all the links and linking devices (usually called nodes) to one another.

### Categories of topology:
There are four basic topologies possible.

There are two derived topologies: Tree, Hybrid.

1.Mesh topology:

1. In a mesh topology, every device has a dedicated point-to-point link to every other device.
2. The term *dedicated* means that the link carries traffic only between the two devices it connects.
3. A fully connected mesh network therefore has **$n(n-1)/2$** physical channels link n devices.
To accommodate that many links, every device on the network must have $n-1$
input/output (*I/O)* ports to be connected to the other $n-1$ stations.
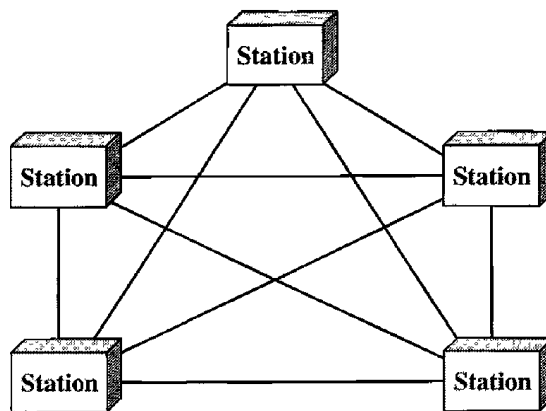
ADVANTAGES:
1. The use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems
2. A mesh topology is robust. i.e. If one link becomes unusable, it does not incapacitate the entire system.
3. There is the advantage of privacy or security.
4. point-to-point links make fault identification and fault isolation easy.

DISADVANTAGES:
1. Because every device must be connected to every other device, installation and reconnection are difficult.
2. The bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.

2.Star Topology:
1. In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub.
2. The devices are not directly linked to one another.
3. A star topology does not allow direct traffic between devices. The controller acts as an exchange: If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected devices.
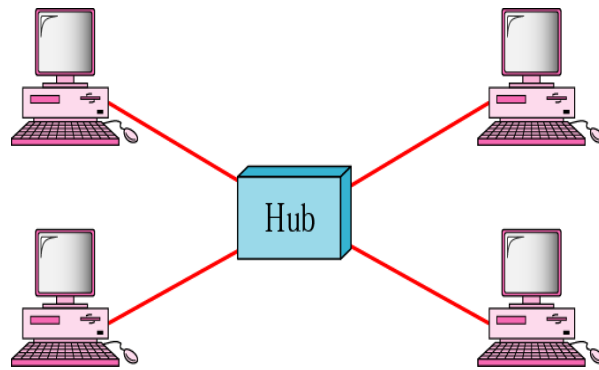
ADVANTAGES:
1. A star topology is less expensive than a mesh topology.
2. It is easy to install and reconfigure.
3. Other advantages include robustness
4. Easy fault identification and fault isolation.

DISADVANTAGE:

The dependency of the whole topology on one single point, the hub. If the hub goes down, the whole system is dead.

3.Bus Topology:

1. A bus topology, is multipoint connected . One long cable acts as a backbone to link all the devices in a network.

2. Nodes are connected to the bus cable by drop lines and taps. A drop line is a connection Running between the device and the main cable. A tap is a connector that either splices into the main cable or punctures the sheathing of a cable to create a contact with the metallic core.
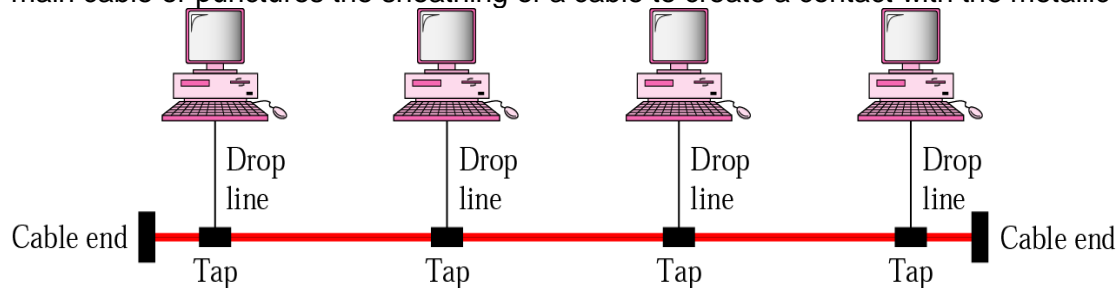
ADVANTAGES

1. Advantages of a bus topology include ease of installation
2. Bus uses less cabling.

DISADVANTAGES:
1. Difficult reconnection and fault isolation is also difficult.
2. Signal reflection at the taps can cause degradation in quality.

4.Ring topology:
- In a ring topology, each device has a dedicated point-to-point connection with only the two devices on either side of it.
- A signal is passed along the ring in one direction, from device to device, until it reaches its destination.
- Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.
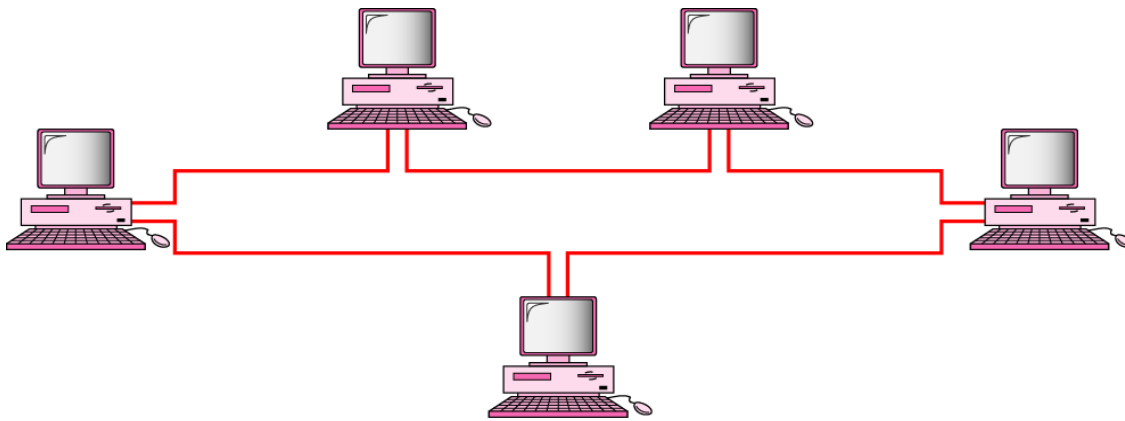
ADVANTAGE:
1. A ring is relatively easy to install and reconfigure.
2. Fault isolation is simplified.

DISADVANTAGE: Unidirectional traffic can be a disadvantage.

Tree topology:

A tree topology is a variation of star topology. As in star, nodes in a tree are linked to a central hub that controls the traffic to the network. However, not every device plugs directly into the central hub. The majority of devices connect to a secondary hub that in turn connected to the central hub. The central hub in the tree is an active hub (i.e. an active hub contains an repeater which regenerates signal.).the secondary hubs may be active or passive hub(i.e. the passive hub provides a simple physical connection between the attached devices)
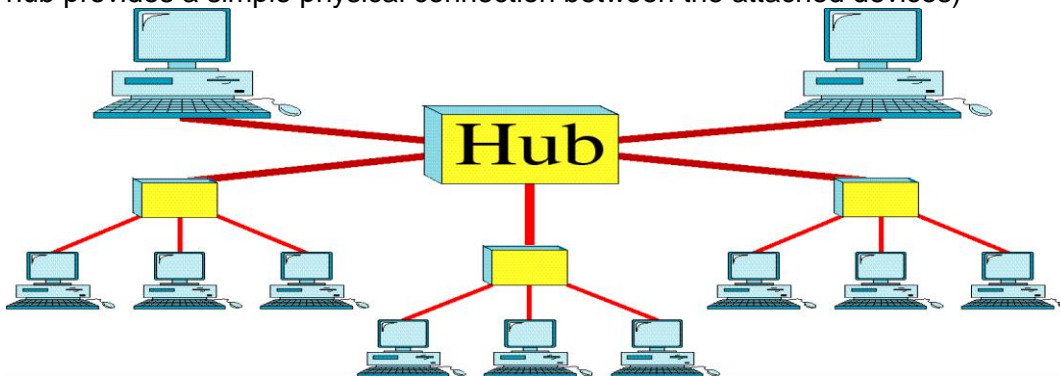
Advantage

- It allows more devices to be attached to a single central hub and can therefore increase the distance a signal can travel between devices.
- It allows the network to isolate and prioritize communication from different computers.

Disadvantage

The dependency of the whole topology on one single point, the central hub. If the central hub goes down, the whole system is dead.

5.Hybrid topology:

Often a network combines a several topologies as subnet works linked together in a larger topology. For example ,one department of a business may have decided to use a bus topology while another department has a ring . The two can be connected to each other via a central controller in a star topology.

Star

Hub

Bus

Ring

Star
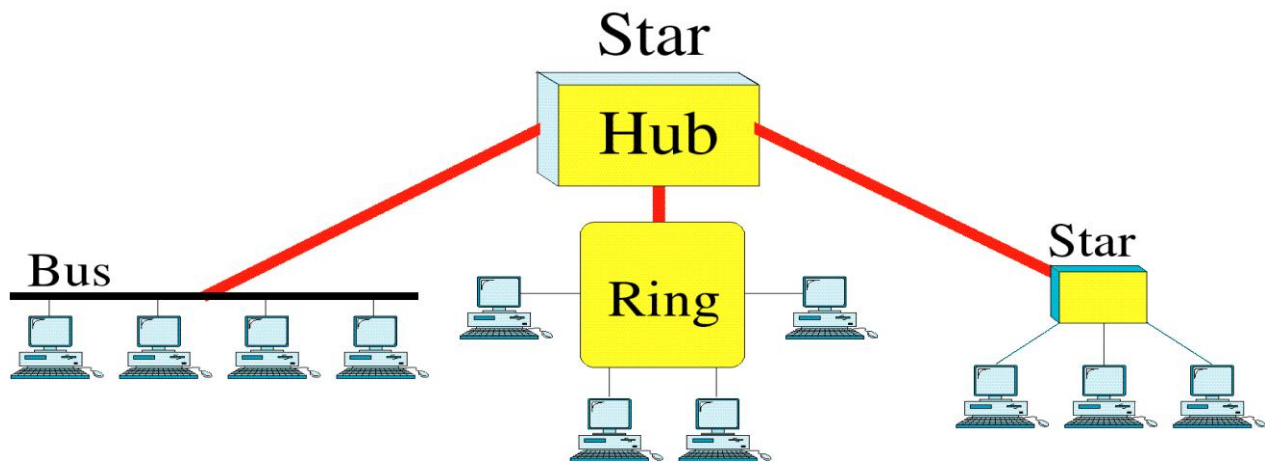
NETWORKING DEVICES

Different Networking Devices are:, Repeater, Hub, Switch, Bridge, Router, Gateway & NIC

Repeater:
- A repeater is an electronic device which is used or installed in the link that it will receive the signals before it becomes weak or corrupted.
- It regenerates the original bit pattern and puts the refreshed copy back unto the link so a repeater is also known as a regenerator.

Regenerated signal   Repeater   Corrupted signal
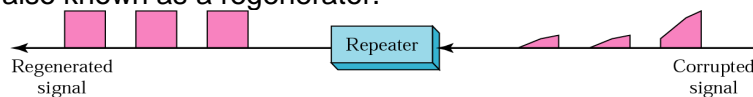
a. Right-to-left transmission.

Corrupted signal   Repeater   Regenerated signal
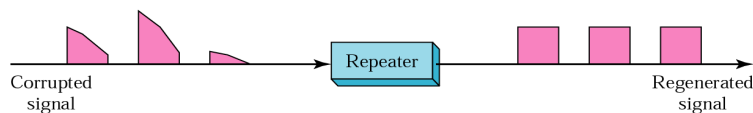
b. Left-to-right transmission.

Hub:
- A common connection point for devices in a network.
- It connects PCs together in a network, but it actually refers to a multi-port repeater.
- A hub contains multiple ports. When a packet arrives at one port, it is copied to the other ports so that all segments of the LAN can see all packets.

Switch:
- Switches are also known as intelligent hubs.
- Switches operate on hardware addresses to transfer data across devices connected to them.
- The reason switches are known as intelligent hubs is because they build address table in hardware to keep track of different hardware addresses and the port to which each hardware address is associated.

Bridge:
- It is interconnecting device which interconnects two segments of LAN.
- It also regenerate signal.

Router
- It is a interconnecting device which interconnects two different networks.
- Routers relay packets among multiple interconnected networks.

Gateway
- Gateways provide translation services between incompatible LANs or application
- Gateway is a protocol converter.
- It is software which installed in the router.

Network interface card (NIC)
- A network interface card (NIC) is a computer circuit board or card that is installed in a computer so that it can be connected to a network.
- Transmits and receives signals to the LAN. Computers cannot communicate on LAN

without this device.
* Each Network Card has a Media Access Control (MAC) address. This is also known as the physical address or Ethernet address.

SERVICES OF INTERNET

Different services of internet are FTP, E-Mail, WWW, IRC, Web Conferencing, Online Shopping etc.

File Transfer Protocol (FTP) :
* File Transfer Protocol (FTP) is the standard mechanism for copying a file from one host to another.

World Wide Web:
* The World Wide Web, abbreviated as WWW or W3 and commonly known as the Web.
* It is the collection of web resources like web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

Internet Relay Chat(IRC):
* Internet Relay Chat (IRC) is a form of real-time Internet text messaging (chat) or synchronous conferencing.
* It is mainly designed for group communication in discussion forums, called *channels*, but also allows one-to-one communication via private message as well as chat and data transfer, including file sharing.

Email:
* Electronic mail, commonly called email or e-mail.
* It is a method of exchanging digital messages from an author to one or more recipients, all having accounts on internet.

Web Conferencing:
Web Conferencing is conducting live meetings, presentations, training, collaborations, or marketing and product launches via the Internet.

Online Shopping:
* The purchase of products and services on the Internet.
  Advantages:
* Wide selection of products and services
* Offers greater convenience than shopping at multiple online shops
* Consumers can make multiple purchases in one transaction.

Different types of internet connectivity:
The connection types in this section include:
1) Dial-up Connections 2) **Leased Line Connection**3)ADSL Connections 3) Cable Connections 4) VSAT

1) Dial-up Connections:
* A dial-up connection allows you to connect to the internet via a local server using a standard 56k modem.
* Your PC literally dials (hence the name) a phone number (provided by your ISP) and connects to the server and therefore the internet.

2) Leased Line Connection:
* Leased connection is also known as direct Internet access.
* It is the secure, dedicated and most expensive, level of Internet connection.
* With leased connection, our computer is dedicatedly and directly connected to the Internet using high speed transmission lines.
* It is on-line twenty-four hours a day, seven days a week.

3) **ADSL (Asymmetric Digital Subscribers Line) Connections:**

   The connections work by splitting your phone line into two separate channels, one for data (internet) and one for voice (phone calls), which means you can talk on the phone and be connected to the internet at the same time.

4) **Cable connections:**

- Through the use of a cable modem you can have a broadband Internet connection that is designed to operate over cable TV lines.
- Cable Internet works by using TV channel space for data transmission, with certain channels used for downstream transmission, and other channels for upstream transmission.

5) **VSAT(very small aperture terminal):**

- It is an earthbound station used in satellite communications of data, voice and video signals, excluding broadcast television.
- A VSAT consists of two parts, a transceiver that is placed outdoors in direct line of sight to the satellite and a device that is placed indoors to interface the transceiver with the end user's communications device, such as a PC.

ISP: An Internet service provider (ISP) is a company that provides access to the Internet.

UNIT-4
FILE MANAGEMENT AND DATA PROCESSING

Concept of File:

- A file is an external collection of related data treated as a unit.
- Files are stored in auxiliary/secondary storage devices.
  - Disk
  - Tapes
- A file is a collection of data records with each record consisting of one or more fields.
- A file is a collection of data records grouped together for purpose of access control and modification.
- It is the primary resource in which we can store information and can retrieve the information when it is required.
- The absolute file name consists of:
  - drive name
  - directory name(s)
  - file name
  - Extension

  For example: d:/network/LAN.doc

Terms Used with Files

- **Field:** This is the basic element of data which contains a single value and characterized by its length and data type.
- **Record:** This is the collection of related fields that can be treated as a unit by some application program.

Types:

Extension	Type of Document	Application
.doc	Word-processing document	Microsoft Word 2003
.docx	Word-processing document	Microsoft Word 2007
.wks	Word-processing document	Microsoft Works word processing
.wpd	Word-processing document	Corel WordPerfect
.xls	Spreadsheet	Microsoft Excel
.slr	Spreadsheet	Microsoft Works spreadsheet
.mdb	Database	Microsoft Access
.ppt	PowerPoint Presentation	Microsoft PowerPoint
.pdf	Portable Document Format	Adobe Acrobat or Adobe Reader

Operations:

- Different types of operations can be performed on the file.
  - Create
  - Delete
  - Open
  - Close
  - Read

- Write

FOLDER:

➢ Folder is a storage location which stores multiple files.
➢ It is a way to organize computer files.
➢ A folder can also contain other folders known as sub-folder.
➢ Folders allow people to organize their files in a way that makes sense to them.
➢ For example, a college student might store all her photos in a folder named "Pictures," all her papers in a folder named "School Work," and all her financial information in a folder named "Finances." All these folders might reside within a folder called "My Documents."

FILE ACCESSING METHODS

Information is stored in files and files reside on secondary storage. When this information is to be used, it has to be accessed and brought into primary main memory. Information in files could be accessed in many ways. There are 3 different file access methods.

- **Sequential access method**
- **Direct access method**
- **Indexed sequential access method**

Sequential Access Method:

- This method is simplest among all methods. Information in the file is processed in order, one record after the other.
- Magnetic tapes are supporting this type of file accessing.

- Ex: A file consisting of 100 records, the current position of read/write head is 45th record, suppose we want to read the 75th record, then it access sequentially from 45,46,........74,75.
- So, the read/write head traverse all the records between 45 to 75.
- Sequential files are typically used in batch application and payroll application.
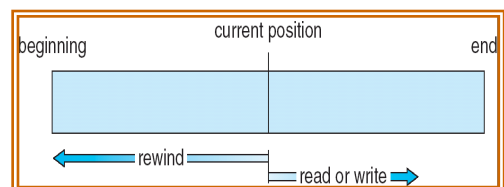
Direct access/ Random access method:

- Direct access is also called as relative access.
- In this method records can read/write randomly without any order.
- The direct access method is based on disk model of a file, because disks allow random access to any file block.
- A direct access file allows arbitrary blocks to be read or written.

- For ex. A disk consisting of 256 blocks, the current position of read/write head is at 95th block. The block to be read or write is 250th block. Then we can access the 250th block directly without any direction.
- Best example for direct access is a CD consisting of 10 songs, at present we are listening the song no.3, suppose we want listening the song no.9, then we can shift from song no.3 to 9 without any restriction.
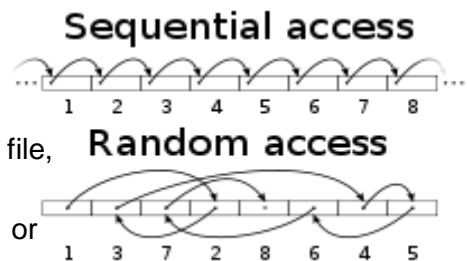
Indexed sequential access method:

- The main disadvantage in the sequential file is, it takes more time to access a record, to overcome this problem, we are using this method.
- In this method, the records are stored sequentially for efficient processing. But they can be accessed directly using index or key field. That's why this method is said to be the indexed sequential file.
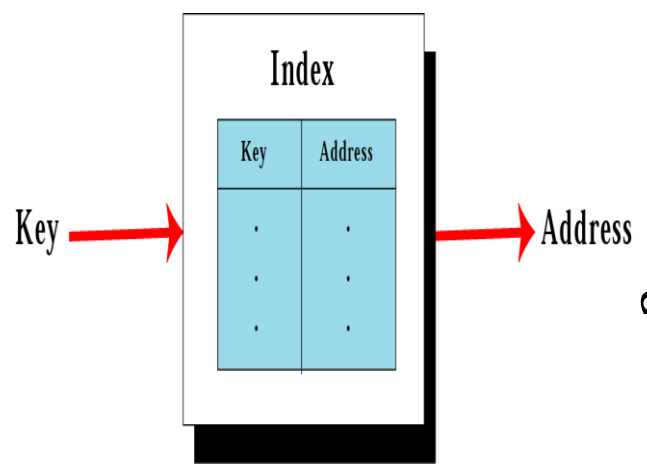
- Keys are pointer which contains address of various blocks.
- Records are organized in sequence based on a key field.
- Generally indexed files are used in airline reservation system and payroll packages.

FILE ALLOCATION/STORAGE METHOD

- Files are normally stored in the disks. So the main problem is how to allocate space to these files so that disk space is utilised effectively and files can be accepted quickly.
- Three major methods of allocating disk space, they are:

  - **Contiguous allocation**
  - **Linked allocation**
  - **Grouped allocation / Indexed allocation**

CONTIGUOUS ALLOCATION:

- In this method each file occupies a set of continuous blocks on the disk.
- Ex- a disk consisting of 1kb blocks. A 100kb file would be allocated 100 consecutive blocks.

  Advantage:-
  This method is best suited for sequential files.

  Disadvantage:-
  - It is difficult to find the contiguous free blocks in the disk.
  - Another problem is external fragmentation; it means some free blocks could happen between two files.
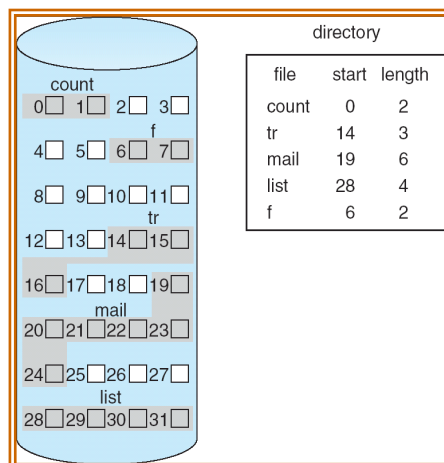
LINKED ALLOCATION:

- In this method every file is a linked list of disk blocks.
- It is easy to locate the files, because allocation is on an individual block basis.
- These disk blocks are present all over the disk.
- Every block contains a pointer for the next free block.
- These pointers are not available to users.
- Ex- there is a file "sort", which consist of 7 blocks, it starts from block 8 and continue to block 15 and from block 15 to block 22 and so on and friendly it is ended with the block is.

  Advantage:-
- Avoid external fragmentation
- Suited for sequential files

  Disadvantage:-
- The pointer itself occupies some memory within the block. So less space available for storing information.
- Takes much accessing time.
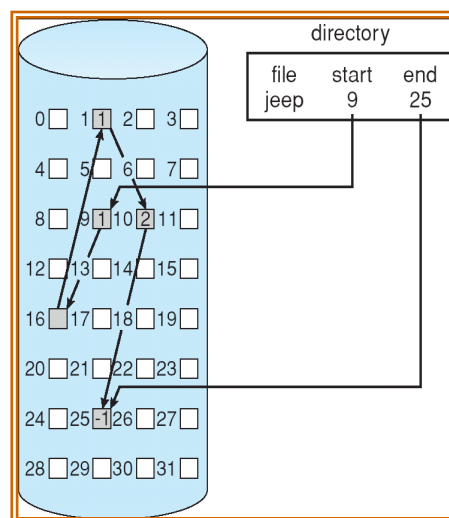
GROUPED ALLOCATION / INDEXED ALLOCATION:-

- This method solves all the problems of the linked allocation methods.
- It solves this problem by bringing all the pointers at a particular place, which is known as index block.
- Individual block having the pointers to the other blocks.
- An individual index block is provided to every file and it contains all the disk below addresses.
- When creating a file, all the pointers are set to nil.

Advantage:-
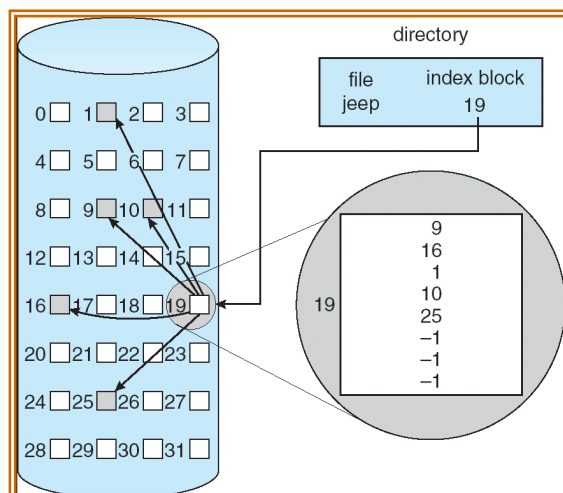
- Indexed allocation supports both sequential and direct access of files.
- The file indexes are not physically stored as part of the file allocation table.
- When the file size increases, we can easily add some more blocks to the index.
- No external fragmentation.

Data Capture:

- ➢ Data capture means obtaining data for a computer:
    - – Collecting documents to be typed in.
    - – Making measurements and keying in.
    - – Asking people to fill in questionnaires.

Methods of Data Capture
Manual methods:
- This is the most commonly used method of collecting or capturing data.
- People are given a form to fill in with their personal details, e.g. name, address, telephone number, date of birth etc.
- Once the form is completed, it is given to a member of staff who will enter the data from it, into a database or information system.

Advantages:
- there shouldn't be much of a need for training, as most people are already familiar with the concept

Disadvantages:
- It can be slow to enter data.
- Transcription (data entry) errors can occur.
- Handwriting recognition can be unreliable.

Automated Methods
- Automated data capture means obtaining data directly by an input device without using a keyboard.
    - – Using document readers (OMR, MICR, OCR)
    - – Bar code readers.
    - – Scanning pictures.
    - – Using sensors for Data Logging.

Advantages:
- Large amounts of data can be read quickly
- Data can be read without human intervention

Disadvantages:
- Specialist equipment is needed to prepare the data for entry – e.g. tags or forms
- Only good for a limited range of data – closed questions.
- Medium is often paper – easily damaged.

Data Processing:

- ➢ Any operation or set of operations performed upon data, Such as collection, recording, organization, Storage or alteration to convert it into useful information.
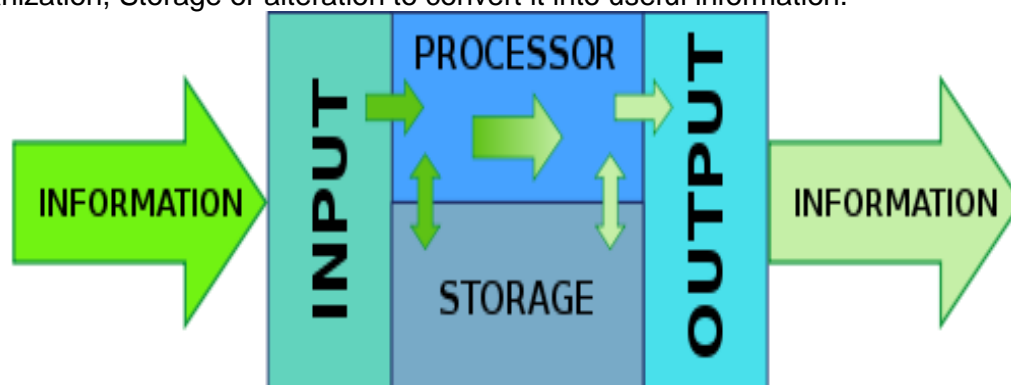
Data Processing Cycle:

➢ Once data is collected, it is processed to convert it into useful information. The data is processed again and again until the accurate result is achieved. This is called data processing cycle.

➢ The data processing is very important activity and involves very careful planning. Usually, data processing activity involves three basic activities.
  1. **Input**
  2. **Processing**
  3. **Output**

  1. **Input:**
     It is the process through which collected data is transformed into a form that computer can understand. It is very important step because correct output result totally depends on the input data. In input step, following activities can be performed.
     i) Coding:
     The verified data is coded or converted into machine readable form so that it can be processed through computer.
     ii) Verification:
     The collected data is verified to determine whether it is correct as required.
      e.g. The collected data of all B.Sc. students that appeared in final examination of the university is verified. If errors occur in collected data, data is corrected or it is collected again.
     iii) Storing:
     The data is stored on the secondary storage into a file. The stored data on the storage media will be given to the program as input for processing.
  2. **Processing:**
     The term processing denotes the actual data manipulation techniques such as classifying, sorting, calculating, summarizing, comparing, etc
  3. **Output:**
   • After completing the processing step, output is   generated.
   • The main purpose of data processing is to get the required result.
   • The output is stored on the storage media for later use.

Data storage and retrieval
   • Data storage and retrieval, the systematic process of collecting and cataloging data so that they can be located and displayed on request.
   • Computers and data processing techniques have made possible the high-speed, selective retrieval of large amounts of information for government, commercial, and academic purposes.
   • There are several basic types of data-storage-and-retrieval systems. Document-retrieval systems store entire documents, which are usually retrieved by title or by key words associated with the document.
   • In some systems, the text of documents is stored as data. This permits full text searching, enabling retrieval on the basis of any words in the document.
   • In others, a digitized image of the document is stored, usually on a write-once optical disc.
   • Database systems store the information as a series of discrete records that are, in turn, divided into discrete fields (e.g., name, address, and phone number);
   • Records can be searched and retrieved on the basis of the content of the fields (e.g., all people who have a particular telephone area code).

- The data are stored within the computer, either in main storage or auxiliary storage, for ready access.
- Reference-retrieval systems store references to documents rather than the documents themselves. Such systems, in response to a search request, provide the titles of relevant documents and frequently their physical locations.
- Such systems are efficient when large amounts of different types of printed data must be stored. They have proven extremely effective in libraries, where material is constantly   changing.

UNIT-5
PROBLEM SOLVING METHODOLOGY

ALGORITHM:
> Algorithm is a step-by-step method to solve a problem.

Characteristics of Algorithms:
> While designing an algorithm, we must take care of the following five important characteristics of an algorithm.

Finiteness:
> An algorithm must terminate after a finite number of steps and each step must be executable in finite amount of time.

Definiteness (no ambiguity):
> Each steps of an algorithm must be precisely defined and the action to be carried out must be unambiguously specified.

Inputs:
> An algorithm has zero or more but only finite, number of inputs.

Output:
> An algorithm has one or more outputs.

Effectiveness:
> An algorithm should be effective. This means each of the operation to be performed in an algorithm must be sufficiently basic that it can be written by person using pen and paper.

Example 1:
> Finding average of any three numbers

> Step1   Start
> Step2   Read values of X,Y,Z
> Step3   S=X+Y+Z
> Step4   A=S/3
> Step5   Write value of A
> Step6   Stop

Example 2:
> Finding biggest of two numbers.

> Step1   Start
> Step2   Read A,B
> Step3   If A>B, then BIG=A, otherwise BIG=B
> Step4   Write BIG
> Step5   Stop

PSEUDOCODE
- Pseudocode is a compact and informal high-level description of an algorithm.
- It is meant for human reading rather than machine reading, so it omits the details that are not essential for humans.
- There are no standards defined for writing a pseudocode because a pseudocode is not an executable program

Example: Write a pseudocode to calculate the weekly wages of an employee. The pay depends on wages per hour and the number of hours worked. Moreover, if the employee has worked for more than 30 hours then he gets twice the wages per hour, for every extra hour that he has worked.

> Read hours worked
> Read wages per hour
> Set overtime charges to 0
> Set overtime hrs to 0
> If hours worked > 30 then
> Calculate overtime hrs = hours worked – 30
> Calculate overtime charges =overtime hrs * (2 * wages per hour)
> Set hours worked = hours worked - overtime hrs
>   ENDIF
> Calculate salary = (hours worked * wages per hour) + overtime charges
> Display salary
> End

FLOWCHART

A flow chart is a graphical or symbolic representation of an algorithm.

SYMBOLS USED IN FLOWCHARTS:

The following table shows a listing of different symbols used in flowcharts along with their meaning and purpose –

Symbol	Name	Function
	Start/end	An oval represents a start or end point.
→	Arrows	A line is a connector that shows relationships between the representative shapes.
	Input/Output	A parallelogram represents input or ouptut.
	Process	A rectangle represents a process.
	Decision	A diamond indicates a decision.

ADVANTAGES OF FLOWCHARTS

The benefits of flowcharts are as follows:
1. **Communication:** Flowcharts are better way of communicating the logic of a system to all concerned.
2. **Effective analysis:** With the help of flowchart, problem can be analysed in more effective way.
3. **Proper documentation:** Program flowcharts serve as a good program documentation, which is needed for various purposes.
4. **Efficient Coding:** The flowcharts act as a guide or blueprint during the systems analysis and program development phase.
5. **Proper Debugging:** The flowchart helps in debugging process.
6. **Efficient Program Maintenance:** The maintenance of operating program becomes easy with the help of flowchart. It helps the programmer to put efforts more efficiently on that part

LIMITATIONS OF FLOWCHARTS

1. **Complex logic**: Sometimes, the program logic is quite complicated. In that case, flowchart becomes complex and clumsy.
2. **Alterations and Modifications:** If alterations are required the flowchart may require re-drawing completely.
3. **Reproduction:** As the flowchart symbols cannot be typed, reproduction of flowchart becomes a problem.

GENERATION OF PROGRAMMING LANGUAGES

FIRST GENERATION: MACHINE LANGUAGE

- Machine level (or low level) language was used to program the first generation computer systems.
- It is the only language that the computer can understand.
- All the commands and data values are expressed using 1s and 0s.
- Machine language is machine-dependent. This makes the code written in machine language less portable.
- Although there were similarities between each of the machine language but a computer could not understand programs written in another machine language
- The main advantage of machine language is that the code can run very fast and efficiently.
- This language is difficult to learn and is more difficult to edit if errors occur.

SECOND GENERATION: ASSEMBLY LANGUAGE

- The second generation includes the assembly language which is a symbolic language
- An assembly language statement consists of a label, an operation code, and one or more operands.
- like the machine language, assembly language is also machine-dependent. This makes the code written in assembly language less portable.
- Code written in assembly language will be very efficient in terms of execution time and main memory usage as the language is also close to the computer.
- Programs written in assembly language need a translator often known as the assembler to convert them into machine language.

THIRD GENERATION PROGRAMMING LANGUAGE

- The second generation includes the high level language.
- 3GLs made programming easier, efficient and less prone to errors.
- 3GL includes languages like FORTRAN and COBOL.
- A translator is needed to translate the instructions written in high level language into computer-executable machine language known as interpreters and compilers.
- The programs written in such languages are portable between machines.

FOURTH GENERATION: VERY HIGH-LEVEL LANGUAGES

- 4GLs are nonprocedural languages. When writing code using a procedural language, the programmer has to tell the computer how a task is done characteristics of 4GLs include:
- The code is written in English-like sentences.
- The code is easier to maintain.
- 4GL code enhances the productivity of the programmers as they have to type fewer lines of code to get something done.
- A typical example of a 4GL is the query language that allows a user to request information from a database with precisely worded English-like sentences.

FIFTH-GENERATION PROGRAMMING LANGUAGE

- 5GLs are centered on solving problems using constraints given to the program, rather than using an algorithm written by a programmer.
- They are widely used in artificial intelligence research. 5GL is that it contains visual tools to help develop a program.
- Example of a fifth generation language are Visual Basic, Prolog etc .
- 5GLs are designed to make the computer solve a given problem without the programmer.

STRUCTURED (MODULAR) PROGRAMMING

- It is a subset of procedural programming.
- It employs a top-down approach which allows the code to be loaded into memory more efficiently and also be reused in other programs.
- Modules are coded separately and once a module is written and tested individually, it is then integrated with other modules to form the overall program structure.
- Modularization makes it easier to write, debug, and understand the program.

Advantages
- Helps to write programs that are easy to write, understand, change, debug, and re-use.
- Modules enhance programmer's productivity
- Takes less time to be written than other programs.

EXAMPLES OF PROBLEM SOLVING THROUGH FLOWCHART

1. Draw a flowchart to print the sum of two numbers.

```
          ( Start )
             |
         / Read n1 and n2 /
             |
        [ Sum ← n1 + n2 ]
             |
         / Print Sum /
             |
          ( Stop )
```
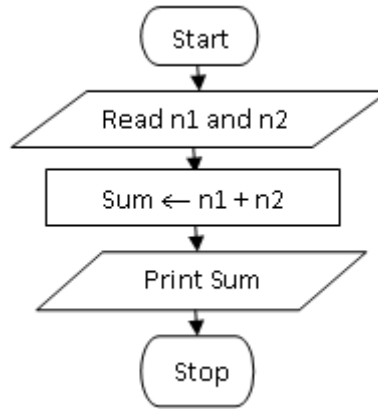
2. Draw a flowchart to print the average of any three numbers.

```
          ( Start )
             |
        / Read n1, n2, n3 /
             |
      [ Sum ← n1 + n2 + n3 ]
      [ Avg ← Sum / 3      ]
             |
         / Print Avg /
             |
          ( Stop )
```
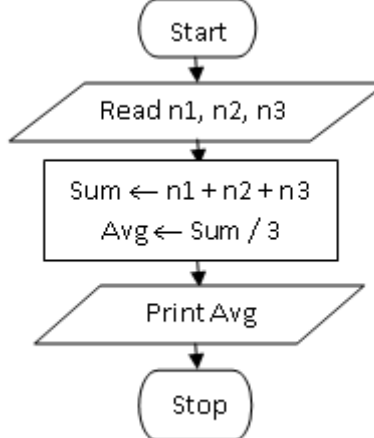
3. Draw a flowchart to print Area and Perimeter of a Square.

```
          ( Start )
             |
        / Read side (a) /
             |
        [ A ← a * a ]
        [ P ← 4 * a ]
             |
        / Print A and P /
             |
          ( Stop )
```
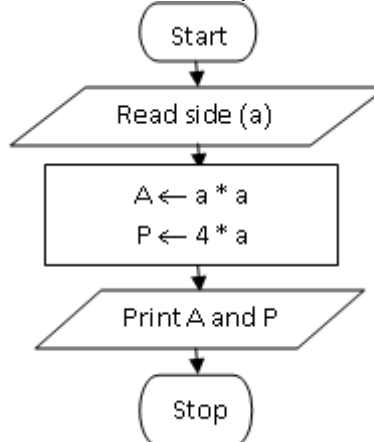
4. Draw a flowchart to print biggest of two numbers.

```
          ( Start )
             |
      / Read n1 and n2 /
             |
        < Is n1 > n2 > ---T---> / Print "n1 is biggest" /
             |                            |
             F                            |
             |                            |
    / Print "n2 is biggest" /            |
             |<---------------------------
          ( Stop )
```
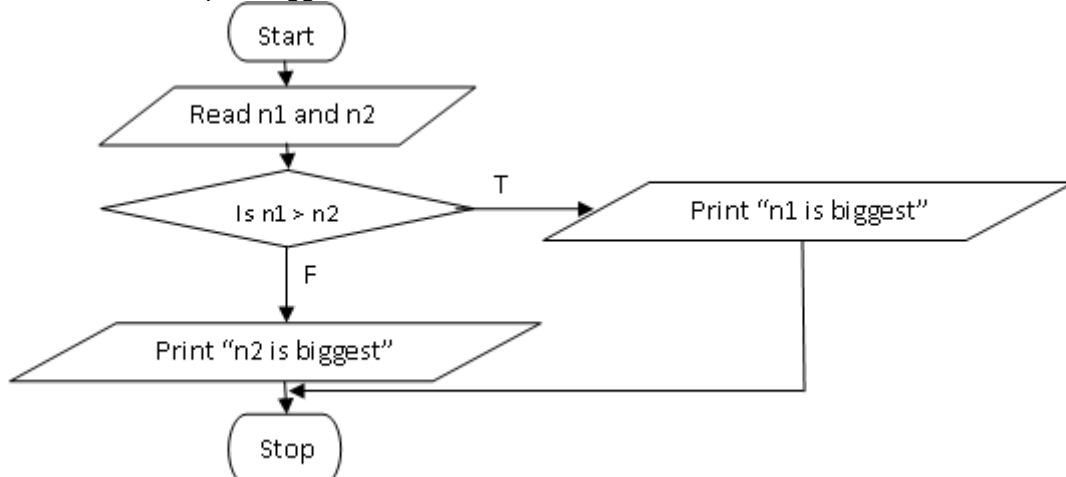
5. Draw a flowchart to check whether a number is positive, negative or zero.

```
                    ( Start )
                        |
                   / Read n /
                        |
              /‾‾‾‾‾‾‾‾‾‾\         T
             <  Is n > 0   >--------->  / Print "n is positive" / ----+
              _____/                                            |
                   | F                                                |
                   v                                                  |
              /‾‾‾‾‾‾‾‾‾‾\         T                                   |
             <  Is n < 0   >--------->  / Print "n is negative" / ----+
              _____/                                            |
                   | F                                                |
                   v                                                  |
        / Print "n is a zero" /                                       |
                   |                                                  |
                   v <------------------------------------------------+
                ( Stop )
```
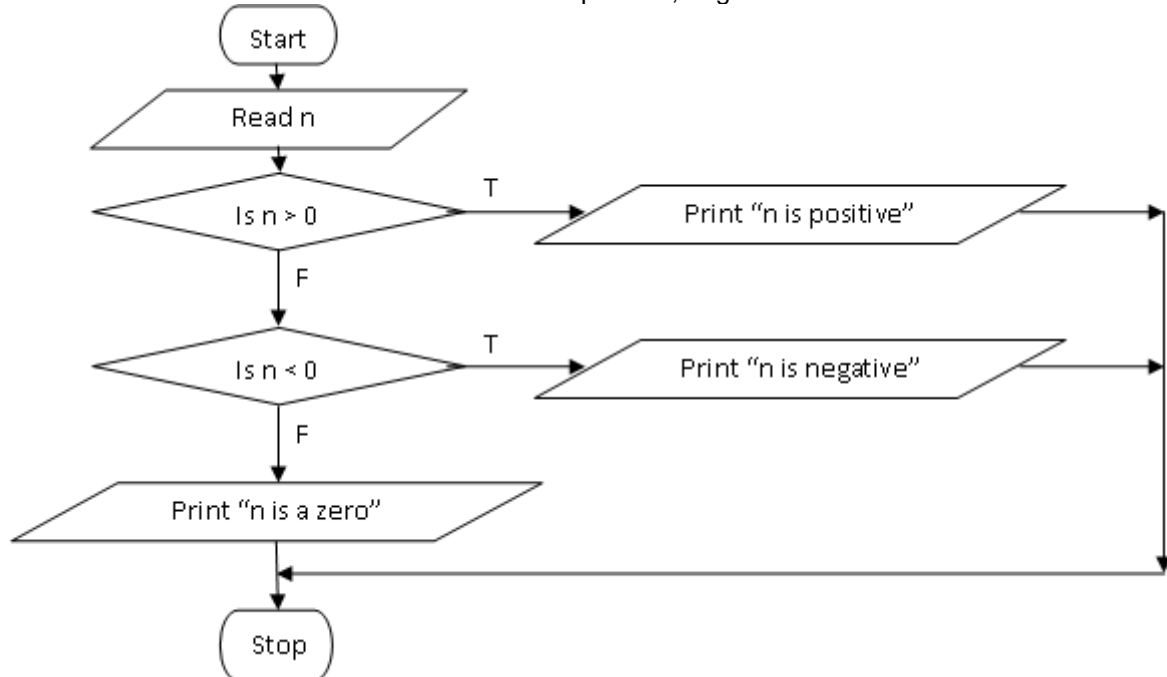
6. Draw a flowchart to print biggest of three numbers.

```
                         [ START ]
                             |
                             v
                      / READ A, B, C /
                             |
                             v
      YES    /‾‾‾‾\   NO  /‾‾‾‾\   YES   /‾‾‾‾\   YES
      +-----<  IS   <----<  IS   >----->  IS   >-----+
      |      \B>C?/       \A>B?/         \A>C?/       |
      |       \__/         \__/           \__/        |
      v                     | NO           | NO       v
 / PRINT B /          / PRINT C /     / PRINT C /  / PRINT A /
      |                     |              |          |
      +---------------------+--------------+----------+
                            |
                            v
                        [ END ]
```
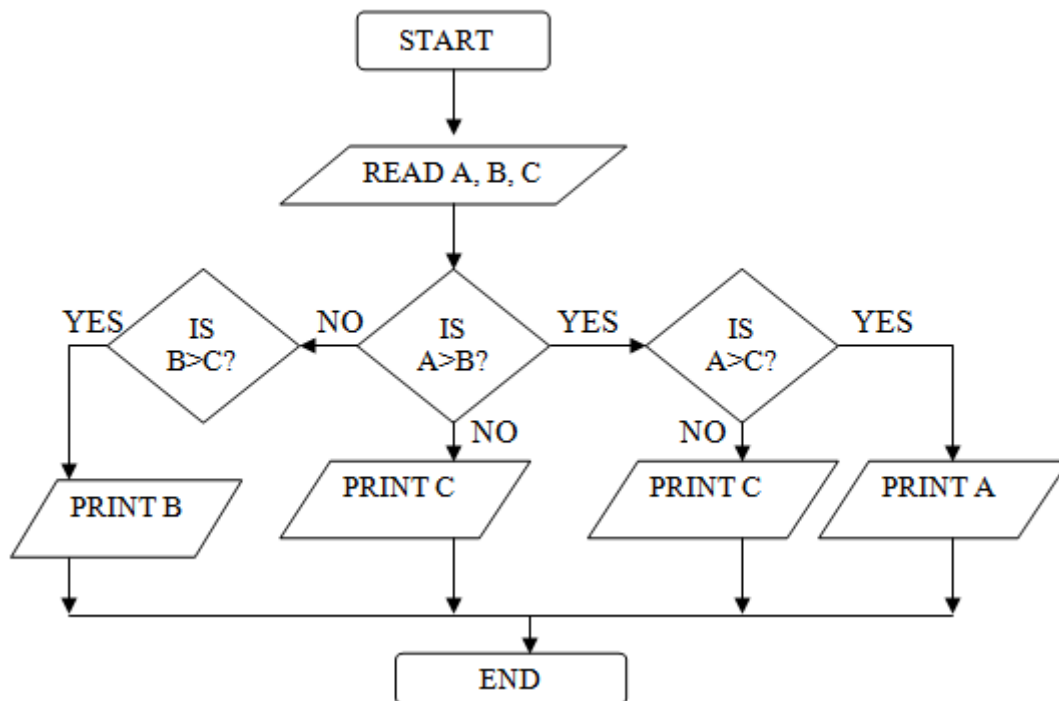
7. Draw a flowchart to print a number 'n' 10 times.

```
              ( Start )
                  |
             / Read n /
                  |
         [ Counter <- 1 ]
                  |
                  v <------------------------------+
            / Print "n" /                          |
                  |                                 |
   [ Counter <- Counter +1 ]                        |
                  |                                 |
                  v                        T        |
         /‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾\ ---------------------+
        < Is counter <=10    >
         _____/
                  | F
                  v
              ( Stop )
```
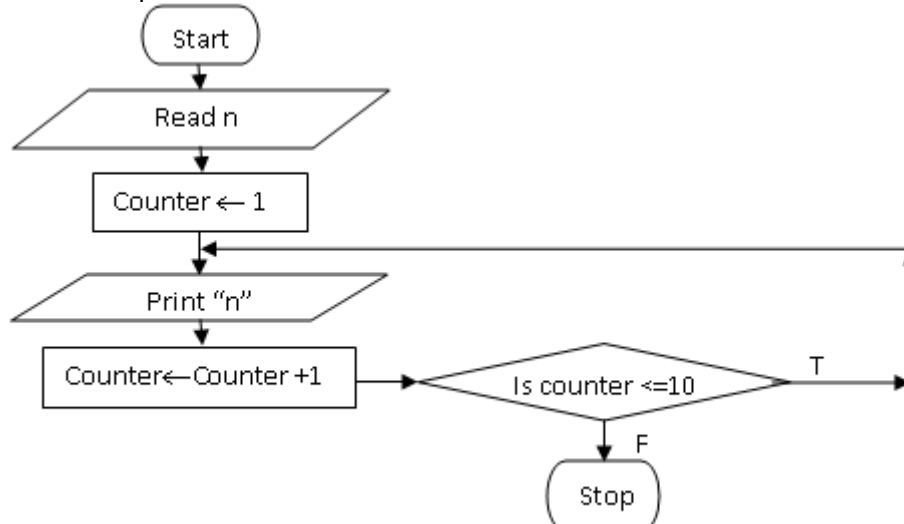
8. Draw a flowchart to print all the odd numbers between 1 and n.

```
        ( Start )
            |
       / Read n /
            |
     [ Counter ← 1 ]
            |
     / Print "Counter" /
            |
  [ Counter ← Counter +2 ] → < Is counter <=n > --T--→ (loop back)
                                    |
                                    F
                                    |
                                ( Stop )
```
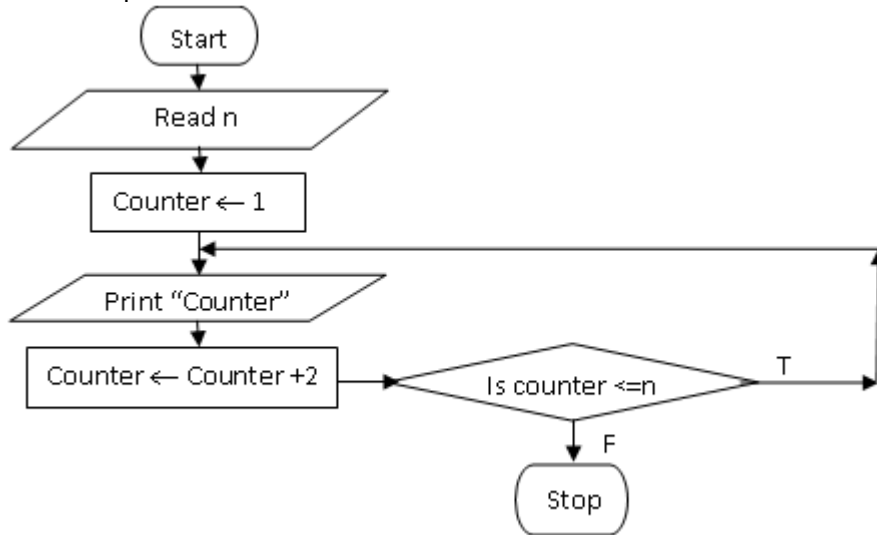
9. Draw a flowchart to print all the even numbers between 1 and n.

```
        ( Start )
            |
       / Read n /
            |
     [ Counter ← 2 ]
            |
     / Print "Counter" /
            |
  [ Counter ← Counter +2 ] → < Is counter <=n > --T--→ (loop back)
                                    |
                                    F
                                    |
                                ( Stop )
```
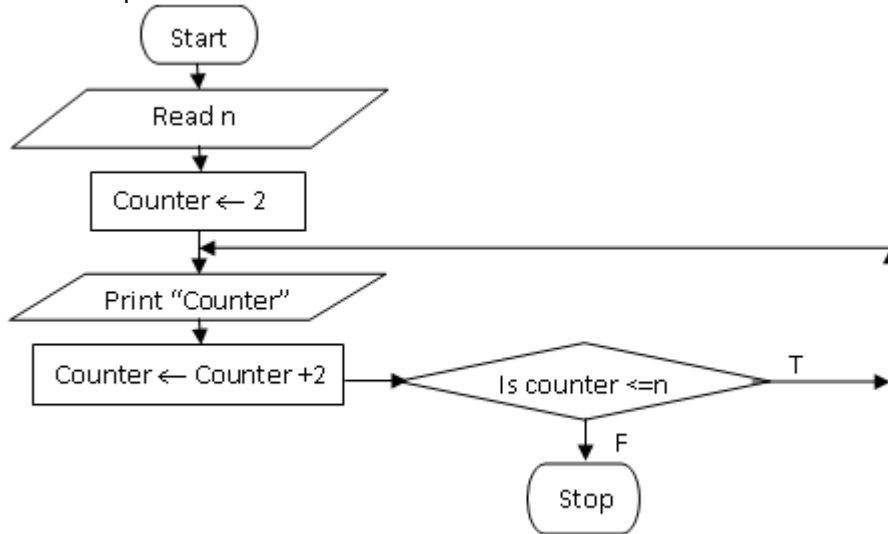
10. Draw a flowchart for computing factorial N (N!)
    Where N! = 1 × 2 × 3 × …… N .

```
        ( START )
            |
       / READ N /
            |
       [ M = 1
         F = 1 ]
            |
       [ F = F * M ]
            |
   NO       |
[ M=M+1 ] ← < IS M=N? >
                |
               YES
                |
          / PRINT F /
                |
            ( END )
```
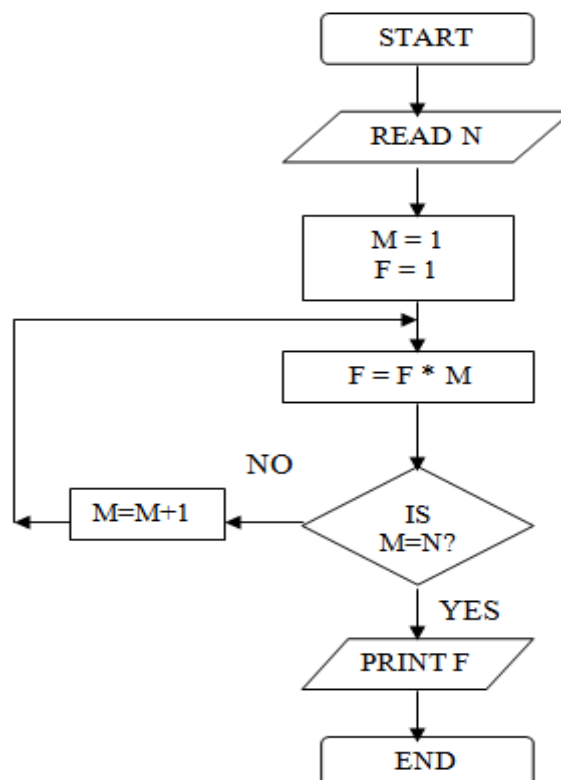
UNIT-6
OVERVIEW OF C PROGRAMMING LANGUAGE

INTRODUCTION TO C:
- C was developed by Dennis Ritchie at Bell laboratory in 1972 .
- It is an upgrade version of languages B and BCPL.

Features of C:
- It is a structured programming language.
- It is highly portable.
- It is a middle level language.
- It is a case sensitive language.
- It uses Top-Down approach**.**

C Character Set:
- Character denotes any alphabet, digit or special symbol used to represent information.
- These characters can be combined to form variables. C uses constants, variables, operators, keywords and expressions as building blocks to form a basic C program.

The characters in C are grouped into the following two categories:

1. **Source character set**

Using this source text are created.

a.	Alphabets	: A to Z and a to z.
b.	Digits	: 0-9 .
c.	Special Characters	: + - * / ^ ~ % = ! &	( ) [ ] { } ? " ' , : : . \ # blank space.

2. **Execution character set:**
- These characters are interpreted at the time of execution .
- These characters are not displayed on the screen or printer.
- Those characters perform other functions aside from displaying text.

Character	Escape Sequence	Result
Null	\0	Null
Alarm (bell)	\a	Beep Sound
Back space	\b	Moves previous position
Horizontal tab	\t	Moves next horizontal tab
New line	\n	Moves next Line
Vertical tab	\v	Moves next vertical tab
Form feed	\f	Moves initial position of next page

Structure of C Program:-

1. Preprocessor directives
2. Variable and function declarations
3. Main function
4. Other functions

1. Preprocessor Directives :
At the time of compilation itself some processes can be done in C. The commands which invoke such processes are called pre-processor directives. These can optionally be present in the program. There are three types of pre-processor directives available with C. They are inclusion (#include), macro substitution (#define) and conditional (#if) directives.

2. Variable and function declarations:
Outside the main function, variables and functions can optionally be declared. Such variables and functions declared before the function main() are global and hence are available to all functions.
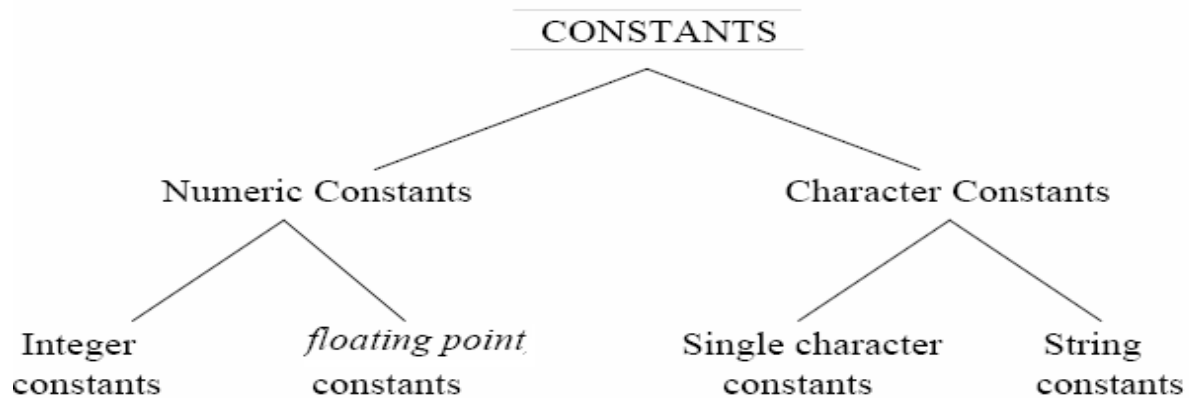
3. Main Function (main()) :

This is the main function of any C program. C compilers start execution from this main().

4. Other functions :-

All other functions called from main() or any other functions can be present after the statements of main().

CONSTANTS

Constant is a memory variable whose value does not change throughout program.

```
                        CONSTANTS

        Numeric Constants              Character Constants

  Integer      floating point      Single character      String
  constants      constants         constants           constants
```

Numeric Constants

A Numeric Constant is made is made up a sequence of numeric digits with optional presence of a decimal point.

Integer Constants

An integer constant refers to a sequence of digits without decimal point. There are three types integers: *decimal, octal*, and *hexadecimal*.

Decimal Constant

E.g.: 123,-321 etc.

Octal Constant

An *octal* integer constant consists of any combination of digits from the set 0 through 7, with a leading 0.

E.g.: 1) 037 2) 0435

Hexadecimal Constant

A sequence of digits preceded by 0x or 0X is considered as *hexadecimal* integer.

They may also include alphabets A to F or a to f.

E.g.: 1) 0X2 2) 0x9F 3) 0Xbcd

Floating point constant

A sequence of numeric digits with decimal points called Floating point constant.

Eg:0.0083, -0.75 etc.

Single Character Constants

A single character constants contains a single character enclosed within a pair of *single* quote marks.

E.g.: '5', 'X', ';'

String Constants

A string constant is a sequence of characters enclosed in *double* quotes. The characters may be letters, numbers, special characters and blank space.

E.g.: "Hello!" , "1987", "?....!"

KEYWORDS

The keywords are also called 'Reserved words' In C Language.

Keywords are the words whose meaning has already been explained to the C compiler and their meanings cannot be changed.

There are 32 keyword in Turbo C :

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

VARIABLES

A *variable* is a data name that may be used to store a data value. A variable may take different values at different times of execution and may be chosen by the programmer in a meaningful way. It may consist of letters, digits and underscore character.

E.g.: 1) Average
       2) Height

Rules for defining variables

* Should not be a reserved word like int etc.
* Should start with a letter or an underscore(_).
* Can contain letters, numbers or underscore.
* No other special characters are allowed including space.
* Variable names are case sensitive i.e. A and a are different.

Variable Declaration:

Syntax

<Data type> <variable name>;

E.g. int a;

Variable Definition

<variable name>=<value>;

E.g. a=10;

Data Types in C

Variable should be declared before it can be used in program.

Data types are the keywords, which are used for assigning a type to a variable.

There are 4 Primary data types in C language. They are:

* **int** – This data type is used to define an integer number (-....-3,-2,-1,0,1,2,3....). A single integer occupies 2 bytes.
* **char** – Used to define characters. A single character occupies 1 byte.
* **float** – Used to define floating point numbers (*single precision*). Occupies 4 bytes.
* **double** – Used for double precision floating point numbers. Occupies 8 bytes.

Data Type	Descripton	Size in bytes	Format
int	Integer value	2	%d
char	Single value	1	%c
float	A real number containing a decimal and/or exponent	4	%f
double	double-precision floating point number giving precision of 14 digits	8	%ld

- An integer data type can be further classified in to signed or unsigned and short or long.

MANAGING INPUT AND OUTPUT OPERATIONS:

- There are a number of I/O functions in C, based on the data type. The input/output functions are classified in two types.
   - Formatted functions
   - Unformatted functions
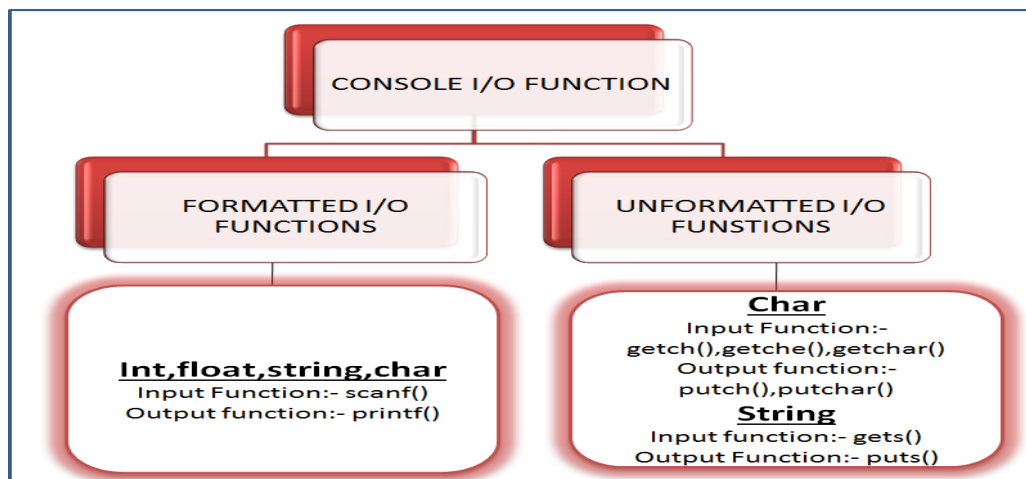
Formatted I/O function
- With the formatted functions, the input or output is formatted as per our requirement.
- All the I/O function are defined in stdio.h header file.
- Header file should be included in the program at the beginning.

printf() function
- This function displays output with specified format.
- It requires format conversion symbol or format string and variables names to the print the data.
- The list of variables are specified in the printf() statement.
- The values of the variables are printed as the sequence mentioned in printf().
- The format string symbol and variable name should be the same in number and type.
   Syntax
     printf("control string", varialbe1, variable2,..., variableN);
- The control string specifies the field format such as %d, %s, %f and variables as taken by the programmer

scanf() function

- scanf() function reads all the types of data values.
- It is used for runtime assignment of variables.
- The scanf() statement also requires conversion symbol to identify the data to be read during the execution of the program.
- The scanf() stops functioning when some input entered does not match format string.

  Syntax :

  scanf("control string", &varialbe1, &variable2,..., &variableN);

-
- Scanf statement requires '&' operator called address operator.
- The address operator prints the memory location of the variable.
- scanf() statement the role of '&' operator is to indicate the memory location of the variable, so that the value read would be placed at that location.

OPERATORS IN C

- An operator is a symbol that specifies an operation to be performed on the operands.
- E.g.: a + b where + is an operator and a, b are operands
- C is very rich in built-in operators, classified into following number of categories:
  - ➢ Arithmetic operators.
    - ❖ Unary operators.
    - ❖ Binary operators
  - ➢ Relational operators.
  - ➢ Logical operators.
  - ➢ Bitwise operators.
  - ➢ Assignment operators.
  - ➢ Increment/Decrement operators.
  - ➢ Conditional operators.

Arithmetic operator

- It is used to carry out arithmetic operations like addition, subtraction etc,

- **E.g.: + , - , * , / etc.**

Operator	Meaning
+	Addition or unary plus
-	Subtraction or unary minus
*	Multiplication
/	Division
%	Modulus

- There are 2 types of arithmetic operators in C:

- **Unary operators:** These are the arithmetic operators that require only one operand.

  E.g.: -a, +b.

- **Binary operators:** These are the arithmetic operators that require two operands.

Arithmetic operator precedence:
- Precedence means priority.
- When an expression contains many operators, the operations are carried out according to the priority of the operators.
- The higher priority operations are solved first.

OPERATORS	PRECEDENCE
Unary + or -	1st
* / %	2nd
+ -	3rd

Relational operators:

- These operators are used to distinguish two values depending on their relations.
- These operators provide the relationship between two expressions.
- If the relation is true it returns a value 1 otherwise 0 for false.

Operator	Meaning
>	Is greater than
>=	Is greater than or equal to
<	Is less than
<=	Is less than or equal to
==	Is equal to
!=	Is not equal to

Logical operators

- C language supports three logical operators. They are- Logical AND (&&), Logical OR (||) and Logical NOT (!).
- The logical expressions are evaluated from left to right.

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A || B
0	0	0
0	1	1
1	0	1
1	1	1

A	!A
0	1
1	0

Bitwise operators

- Bitwise operators perform operations at bit level. These operators include: bitwise AND, bitwise OR, bitwise XOR and shift operators.

Operator	Meaning
&	Bitwise AND
		Bitwise OR
^	Bitwise Exclusive
<<	Shift left
>>	Shift right

Assignment operators

- The assignment operator is responsible for assigning values to the variables.
- The equal sign (=) is the fundamental assignment operator.
  E.g. : a=5;
- C also supports other assignment operators that provide shorthand ways to represent common variable assignments.

Increment and decrement operators:

- C provides two operators for incrementing and decrementing variables.
- The increment operator ++ adds 1 to its operand, while the decrement operator -- subtracts 1.
- The ++ and -- may be used either before the variable called pre increment/decrement, or after the variable called post increment/ decrement.
- E.g. ++a (i.e. Pre increment)
      --a (i.e. Pre decrement)
    a++ (i.e. Post increment)
    a-- (i.e. Post decrement)
- In both cases, the effect is to increment a. But the expression ++a increments a before its value is used, while a++ increments a after its value has been used.

Conditional operator:
- The conditional operator operator (?:) is just like an if .. else statement that can be written within expressions.
- The syntax of the conditional operator is
-           exp1 ? exp2 : exp3
- Here, exp1 is evaluated first. If it is true then exp2 is evaluated and becomes the result of the expression, otherwise exp3 is evaluated and becomes the result of the expression. For example,
-   large = (a > b) ? a : b
- Conditional operator is also known as ternary operator as it takes *three* operands.

Expression

- Operators, functions, constants and variables are combined together to form expressions.
- Consider the expression A + B * 5. Where, +, * are operators, A, B are variables, 5 is constant and A + B * 5 is an expression.

Type conversion and type casting

- Type conversion and type casting of variables refers to changing a variable of one data type into another.
- Type conversion is done implicitly, and casting has to be done explicitly by the programmer.
- Type conversion is done when the expression has variables of different data types. So to evaluate the expression, the data type is promoted from lower to higher level where the hierarchy of data types can be given as: double, float, long, int, short and char.
- For example, type conversion is automatically done when we assign an integer value to a floating point variable.
  For example:
    float x;
    int y = 3;
    x = y;
     Now, x = 3.0,
- Type casting is also known as forced conversion. It is done when the value of a higher data type has to be converted in to the value of a lower data type. For example, we need to explicitly type cast an integer variable into a floating point variable.

```
float salary = 10000.00;
int sal;
sal = (int) salary;
```

- Typecasting can be done by placing the destination data type in parentheses followed by the variable name that has to be converted.

Decision Control statements:

- Decision control statements are used to alter the flow of a sequence of instructions.
- These statements help to jump from one part of the program to another depending on whether a particular condition is satisfied or not.
- These decision control statements include:
  ➢ If statement
  ➢ If else statement
  ➢ If else if statement
  ➢ Switch statement

IF STATEMENT

- If statement is the simplest form of decision control statements that is frequently used in decision making. The general form of a simple if statement is shown in the figure.
- First the test expression is evaluated. If the test expression is true, the statement of if block (statement 1 to n) are executed otherwise these statements will be skipped and the execution will jump out from if.
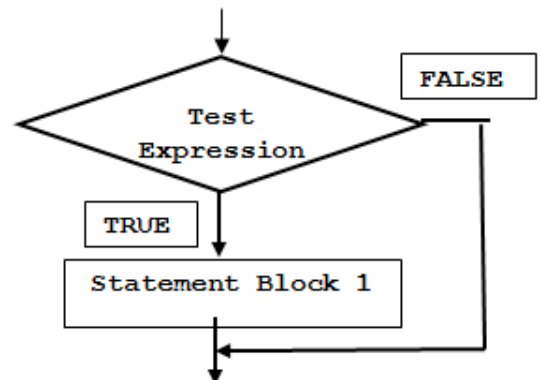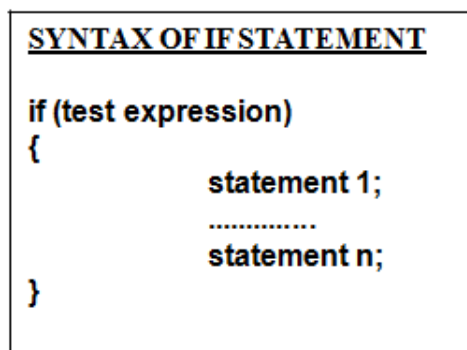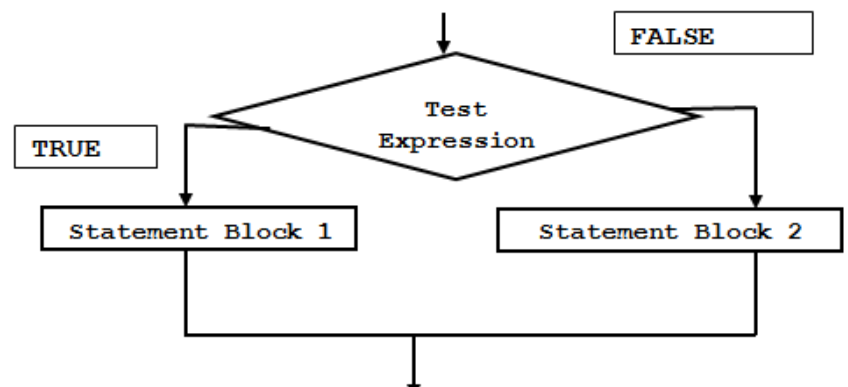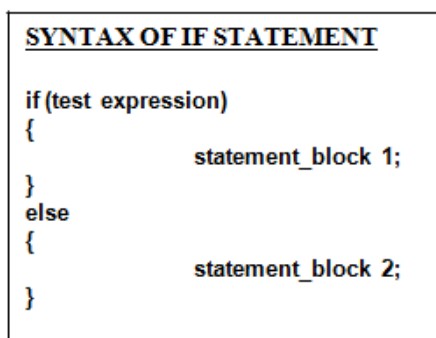

Example:
```
int m=40,n=40;
 if (m == n)
 {
    printf("m and n are equal");
 }
```

IF ELSE STATEMENT

- In the if-else construct, first the test expression is evaluated. If the expression is true, statement block 1 is executed and statement block 2 is skipped. Otherwise, if the expression is false, statement block 2 is executed and statement block 1 is ignored.
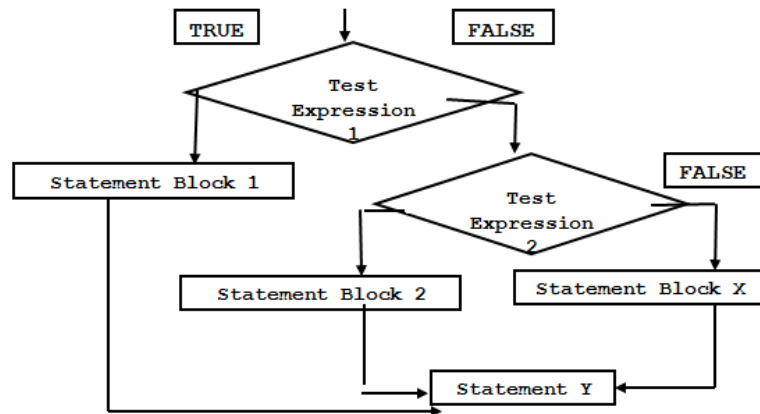
Example:

```
int m=40,n=20;
 if (m == n) {
    printf("m and n are equal");
 }
 else {
     printf("m and n are not equal");
 }
```

IF ELSE IF STATEMENT

- C language supports if else if statements to test additional conditions apart from the initial test expression. The if-else-if construct works in the same way as a normal if statement.

```
SYNTAX OF IF-ELSE STATEMENT

if ( test expression 1)
{
        statement block 1;
}
else if ( test expression 2)
{
        statement block 2;
}
..........................
else if (test expression N)
{
        statement block N;
}
else
{
        Statement Block X;
}
Statement Y;
```


Example:

```
void main()
{
 int numb;
 printf("Type any Number : ");
  scanf("%d", &n);
 if(n > 0)
 {
   printf("%d is the positive number", n);
 }
 else if(n < 0)
   printf("%d is the Negative number", n);
 else
   printf("%d is zero",n);
}
```

SWITCH CASE

- A switch case statement is a multi-way decision statement. Switch statements are used:
  - When there is only one variable to evaluate in the expression.
  - When many conditions are being tested for.
- Switch case statement advantages include:
  - Easy to debug, read, understand and maintain.
  - Execute faster than its equivalent if-else construct.

Syntax:
```
switch (expression)
{
    case value1:
            Statement block-1:
            break;
    case value2:
            Statement block-2:
            break;
    ....
    ...
    case value-n:
            statement block-n:
            break;
    default:
            default statement;
            break;
}
```

Example:
```
switch(grade)
{
        case 'A':
            printf("\n Excellent");
            break;
        case 'B':
            printf("\n Good");
            break;
        case 'C':
            printf("\n Fair");
            break;
        default:
            printf("\n Invalid Grade");
}
```

ITERATIVE STATEMENTS/ LOOP STATEMENTS

- Iterative statements are used to repeat the execution of a list of statements, depending on the value of an integer expression.
- Different types of loops are:
  - ➢ While loop
  - ➢ Do-while loop
  - ➢ For loop

WHILE LOOP

- The while loop is used to repeat one or more statements while a particular condition is true.
- In the while loop, the condition is tested before any of the statements in the statement block is executed.
- If the condition is true, only then the statements will be executed otherwise the control will jump to the immediate statement outside the while loop block.
- We must constantly update the condition of the while loop.

Syntax:
```
while (condition)
{
    statement_block;
}
```
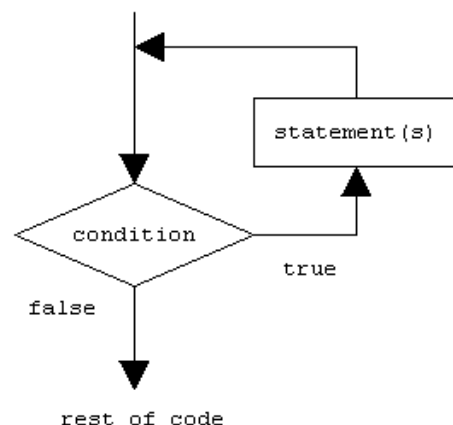
Example:
```
int i = 0;
    while(i<=10)
    {
            printf("\n %d", i);
            i = i + 1;
    }
```

DO WHILE LOOP
- The do-while loop is similar to the while loop. The only difference is that in a do-while loop, the test condition is tested at the end of the loop.
- The body of the loop gets executed at least one time (even if the condition is false).
- The do while loop continues to execute whilst a condition is true. There is no choice whether to execute the loop or not. Hence, entry in the loop is automatic there is only a choice to continue it further or not.

Syntax:
> **do**
>
> **{**
>
> > **statement_block;**
>
> **} while (condition);**

Example:

```
int i = 0;
    do
    {
            printf("\n %d", i);
            i = i + 1;
    } while(i<=10);
```
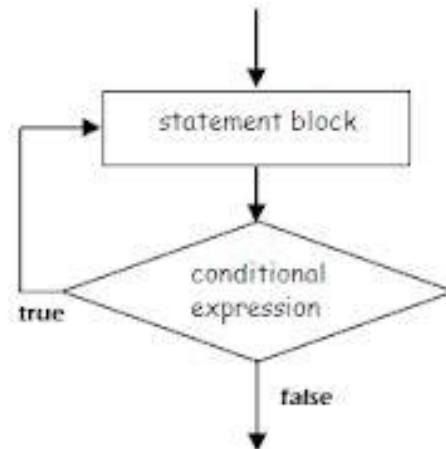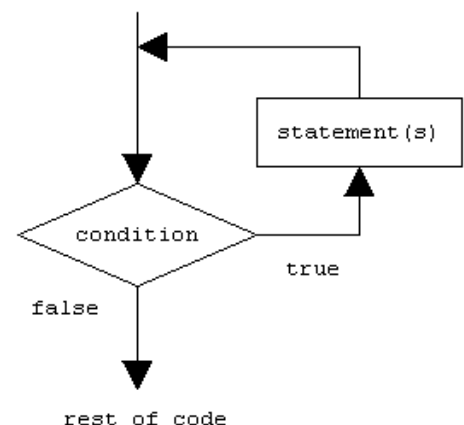
FOR LOOP
- For loop is used to repeat a task until a particular condition is true.

- When a for loop is used, the loop variable is initialized only once.
- With every iteration of the loop, the value of the loop variable is updated and the condition is checked. If the condition is true, the statement block of the loop is executed else, the statements comprising the statement block of the for loop are skipped and the control jumps to the immediate statement following the for loop body.
- Updating the loop variable may include incrementing the loop variable, decrementing the loop variable.

syntax
> **for (initialization; condition; increment/decrement)**
> > **{**
> > > **statement block;**
> >
> > **}**

Example:

```
for(i=0; i<= n; i++)
    printf("\n %d", i);
```

BREAK STATEMENT

In C, the break statement has two uses.

- It terminates a statement sequence in a switch statement.
- It can be used to exit from a loop.

Syntax:

break;

Example:
```c
#include <stdio.h>
void main()
{
   int i;
   for(i=0;i<10;i++)
    {
       if(i==5)
        {
               printf("\nComing out of for loop when i = 5");
               break;
        }
    printf("%d ",i);
     }
}
```

Output:

0 1 2 3 4

Coming out of for loop when i = 5

Continue statement :

Continue statement is used to continue the next iteration of for loop, while loop and do-while loops. So, the remaining statements are skipped within the loop for that particular iteration.

Syntax :

continue;

Example:
```c
int i;
   for(i=0; i<= 10; i++)
        {       if (i==5)
                   continue;
              printf("\t %d", i);
        }
```

goto statement :

•goto statements is used to transfer the normal flow of a program to the specified label in the program.

Syntax:
```c
{
   …….
   go to label;
   …….
   …….
   LABEL:
   statements;
}
```

Example:
```c
int num, sum=0;
read:
    printf("\n Enter the number. Enter 999
            to scanf("%d", &num);

    if (num != 999)
    {
            if(num < 0)
                    goto read;
            sum += num;
            goto read;
    }
    printf("\n Sum of the numbers entered by
            the user is = %d", sum);
```

ADVANCED FEATURES OF C

Functions:
- A function is a self contained block of code that performs a particular task.
- A complex problem is easier to solve by dividing it into several smaller parts, each of which can be solved by itself.
- Any C program is a collection of functions.
- A function takes some data as input, perform some operation on that data and then return a value.
- Any C program must contain at least one function, which is main().
- There is no limit on the number of functions that might be present in a C program.
  For example:
  ```
  main()
  {
          message();
          printf("I am in main");
  }
  message()
  {
          printf("I am in message");
  }
  ```

Advantages of Functions
- Functions separate the concept (what is done) from the implementation (how it is done).
- Functions make programs easier to understand.
- Functions can be called several times in the same program, allowing the code to be reused.

Types of functions
In C language function is divided in to two types:
1. Predefined functions /standard functions/ Built-in functions/Library functions:
2. User-Defined functions.

1. Predefined functions /standard functions/ Built-in functions/Library functions:
- Predefined functions provided by C that are used by programmers without having to write any code for them.
- These are the functions that have been pre-defined by C and put into standard C libraries. i.e.: printf( ), scanf( ), sqrt() , pow() , log() , sin() , getch() , clrscr() etc.
- What we need to do to use them is to include the appropriate header files. Example: #include <stdio.h>, #include <math.h> etc.

2. User-Defined functions:
- Functions that are written by the programmers themselves to carry out various individual tasks.
- It is easier to plan and write our program if we divide it into several functions instead of writing a long piece of code inside the main function.
- A function is reusable and therefore prevents us (programmers) from having to unnecessarily rewrite what we have written before.
- In order to write and use our own function, we need to do the following:
  - ➤ create a function prototype (i.e. declare the function)
  - ➤ define the function somewhere in the program (i.e. function implementation)
  - ➤ call the function whenever it needs to be used.

Function declaration:
- The declaration of function needs to be done before the first call of the function.
- A function declaration specifies the name, return type, and arguments of a function. This is also called the function prototype.
- To be a prototype, a function declaration must establish types for the function's arguments.
- Having the prototype available before the first use of the function allows the compiler to check that the correct number and types of arguments are used in the function call.
- main () is user-defined functions.
- The prototype has the same syntax as the function definition, except that it is terminated by a semicolon following the closing parenthesis and therefore has no body.

Function Definition:
- It is also called as function implementation
- A function definition is where the actual code for the function is written. This code will determine what the function will do when it is called.

- A function definition consists of the following elements:
  - ➢ A function return data type (return value).
  - ➢ A function name.
  - ➢ An optional list of formal parameters enclosed in parentheses (function arguments).
  - ➢ function body.
- General form of any function definition is:

      return-type function-name(argument declarations)
      {
              declarations and statements
      }
- Return-type refers to the data type of the value being returned from the function. If the return type is omitted, int is assumed.
- The values provided to a function for processing are the arguments.
- The set of statements between the braces is called as the function body.
- Any function not returning anything must be of type 'void'.
- If the function does not receive any arguments from the calling function, 'void' is used in the place of the arguments.

Passing Parameters to the Function

In function parameter can be passed in two ways:
1. Call by Value
2. Call by Reference

1. Call by Value:
- In C, all functions are passed "by value" by default.
- This means that the called function is given the values of its arguments in temporary variables rather than the originals.
- For ex. main()

      {
              int a=4,b=5;
              sum(a,b);
              printf("Sum = %d",a+b);
      }
      sum(int a,int b)
      {
              a++;
              b++;
              printf("Sum = %d",a+b);
      }

2. Call by Reference:
- When necessary, it is possible to arrange a function which can modify a variable in a calling routine.
- The caller must provide the address of the variable to be set (generally, a pointer to a variable), and the called function must declare the parameter to be a pointer and access the variable indirectly through it.
- For ex: main()

      {
              int a=4,b=5;
              sum(&a,&b);
              printf("Sum = %d",a+b);
      }
      sum(int *a,int *b)
      {
              (*a)++;
              (*b)++;
              printf("Sum = %d",(*a)+(*b));
      }

Function Call:
- Any function (including main) could utilize any other function definition that exist in a program – hence it is said to call the function (function call).
- To call a function (i.e. to ask another function to do something for the calling function), it requires the FunctionName followed by a list of actual parameters (or arguments), if any, enclosed in parenthesis.

- If the function requires some arguments to be passed along, then the arguments need to be listed in the bracket ( ) according to the specified order. For example:

```
void Calc(int, double, char, int);
void main(void)
{
    int a, b;
    double c;
    char d;
    …
    Calc(a, c, d, b);
}
```

Recursion of Function:

- Recursion of Function is a programming technique in which a function calls itself.
- A recursive function calls itself repeatedly, with different argument values each time.

C program to calculate factorial using recursion

```
#include<stdio.h>
int fact(int);
int main()
{
  int num,f;
  printf("\nEnter a number: ");
  scanf("%d",&num);
  f=fact(num);
  printf("\nFactorial of %d is: %d",num,f);
  return 0;
}

int fact(int n)
{
  if(n==1)
     return 1;
  else
     return(n*fact(n-1));
}
```

Type of Recursion:

- There are basically two types of recursions in programming.
    1. Direct Recursion
    2. Indirect Recursion

1. **Direct Recursion:** A direct recursion is when a function calls itself from within its own body. This form is the most common among programming.
2. **Indirect Recursion :** An indirect recursion happens when first function calls second function from its body and second function calls first function from its body.


(a) Direct Recursion.

(b) Indirect Recursion.
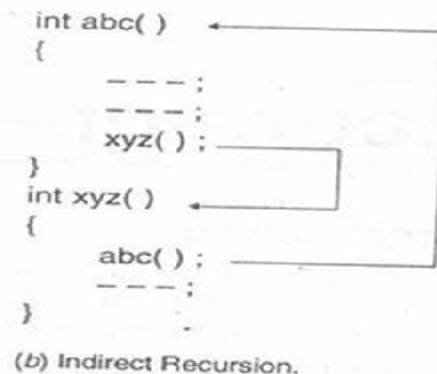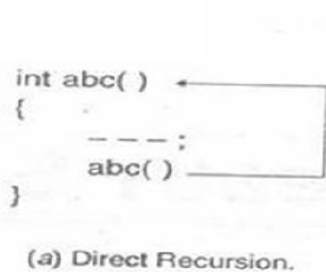
Difference between Recursion and Iteration

RECURSION	ITERATIONS
It is a programming technique in which a function calls itself.	The block of statement executed repeatedly using loops.
Recursion Uses selection structure.	Iteration uses repetition structure.
Recursion is usually slower then iteration.	Iteration is faster than recursion.
Recursion uses more memory than iteration.	Iteration consumes less memory.
Recursion makes code smaller	Iteration makes code longer.

SCOPE OF VARIABLES:
- In C, all constants and variables have a defined scope.
- Scope means the accessibility and visibility of the variables at different points in the program.
- A variable or a constant in C has four types of scope: block, function, file and program scope.
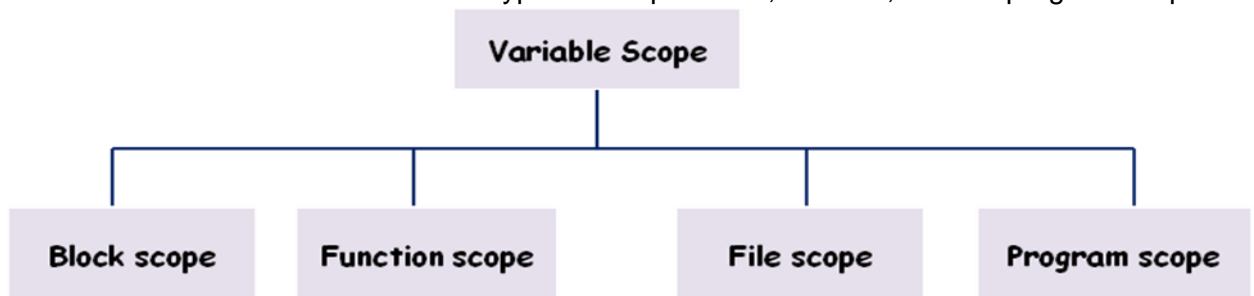
1.BLOCK SCOPE :
- A statement block is a group of statements enclosed within an opening and closing curly brackets ({ }).
- If a variable is declared within a statement block then, as soon as the control exits that block, the variable will cease to exist. Such a variable also known as a local variable is said to have a block scope.

2.FUNCTIONSCOPE:
Function scope is applicable only with goto label names. That is the programmer can not have the same label name inside a function.

3.PROGRAM SCOPE:
- If you want that functions should be able to access some variables which are not passed to them as arguments, then declare those variables outside any function blocks.
- Such variables are commonly known as global variables. Hence, global variables are those variables that can be accessed from any point in the program.

4.FILE SCOPE:
- When a global variable is accessible until the end of the file, the variable is said to have file scope.
- To allow a variable to have file scope, declare that variable with the static keyword before specifying its data type, like this:
                static int x = 10;
- A global static variable can be used anywhere from the file in which it is declared but it is not accessible by any other files.
- Such variables are useful when the programmer writes his own header files.

STORAGE CLASS
- The storage class determines the part of the memory where the variable would be stored.
- The storage class also determines the initial value of the variable.
- It is used to define the scope and lifetime of variable.
- There are two storage location in computer: CPU Registers and Memory
- A value stored in a CPU register can always be accessed faster than the one that is stored in memory.

TYPES OF STORAGE CLASSES

There are four types of storage classes in C:
  i.   Automatic storage class
  ii.   Register storage class
  iii. Static storage class
  iv.   External storage class

Automatic Storage Class

- In automatic storage class auto Keyword is used to declare the variable.
- Storage area of variable is memory.
- Default initial value of variable is garbage value.
- Scope of variable is local to the block in which the variable is defined.
- Life of variable exits till the control remains within the block in which the variable is defined.

Register Storage Class

- In Register storage class register Keyword is used to declare the variable.
- Storage area of variable is CPU register.
- Default initial value of variable is garbage value.
- Scope of variable is local to the block in which the variable is defined.
- Life of variable exits till the control remains within the block in which the variable is defined.

Static Storage Class

- In static storage class static Keyword is used to declare the variable.
- Storage area of variable is memory.
- Default initial value of variable is zero.
- Scope of variable is local to the block in which the variable is defined.
- Life of the variable persists between different function calls.

External Storage Class

- In extern storage class extern Keyword is used to declare the variable.
- Storage area of variable is memory.
- Default initial value of variable is zero.
- Scope of variable is global means value of variable is visible in all the Program
- Life of the variable persists as long as the program's execution doesn't come to an end.

Array

An array is a collection of same type of elements sharing a common name.

- Arrays can be created from any of the C data-types like int, float, and char. So an integer array can only hold integer values and cannot hold values other than integer.
- When we declare array, it allocates contiguous memory location for storing values.
- In C Array subscripts start at 0 and end one less than the array size .

Types of C arrays:

There are 2 types of C arrays. They are,
  1. Single dimensional array
  2. Multi dimensional array
      - Two dimensional array
      - Three dimensional array, four dimensional array etc…

Single dimensional array:

The array which has single dimension and can be represented by single subscript is called single or one dimensional array.

Declaration of Single dimensional array:

Syntax :

        data-type arr_name[array_size];

Example:

        int roll[8];

Array initialization:

 Syntax:

        data_type arr_name [arr_size]= (value1, value2, value3,….);

Example:

          int roll[8]={12, 45, 32, 23,17, 49, 5,11};

Name	roll[0]	roll[1]	roll[2]	roll[3]	roll[4]	roll[5]	roll[6]	roll[7]
Values	12	45	32	23	17	49	5	11
Address	1000	1002	1004	1006	1008	1010	1012	1014

1-D Array memory arrangement

Program:

WAP in C to accept 10 number, store them in an array and display the array elements.

```c
#include<stdio.h>
void main()
{
    int num[4];
    int i;
    for ( i = 0; i < 10; i++ )
        {
            Printf(Enter numbers:);
            Scanf("%d",&num[i]);
        }
    for ( i = 0; i < 10; i++ )
            printf("\n num[%d] = %d\n", i , num[i]);

}
```

Multi-dimensional arrays

- The array which has more than one dimension and can be represented by more than one subscripts is called multi-dimensional array.

Declaration of Single dimensional array:

Syntax :
type name[size1][size2]...[sizeN];
Example:
float a[2][6][7][4]….[n];

Two dimensional array

Two dimensional arrays can represent the data in the form of rows and columns means taking two subscripts.

Syntax:
data type array name[row size][col. Size];

Example:
int matrix[3][3];

int matrix[5] [5];

matrix[2] [1] = 7

```
int[][] array ={
{1, 2, 3},
{4, 5, 6},
{7, 8, 9},
{10, 11, 12}};
```

Initializing of two dimensional Arrays

To declare, create and initialize a multidimensional array.
For example,
int[][] array = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9}, {10, 11, 12} };

C program for addition of two matrices using arrays source code. Matrix addition in c language:

```c
#include<stdio.h>
Void main()
{
```

```c
int a[3][3],b[3][3],c[3][3],i,j;
printf("Enter the First matrix->");
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&a[i][j]);
printf("\nEnter the Second matrix->");
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        scanf("%d",&b[i][j]);
printf("\nThe First matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",a[i][j]);
}
printf("\nThe Second matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
    printf("%d\t",b[i][j]);
}
for(i=0;i<3;i++)
    for(j=0;j<3;j++)
        c[i][j]=a[i][j]+b[i][j];
printf("\nThe Addition of two matrix is\n");
for(i=0;i<3;i++){
    printf("\n");
    for(j=0;j<3;j++)
        printf("%d\t",c[i][j]);
}
}
}
```

STRING

- A string in C is a array of character.
- It is a one dimensional array type of char.
- Every string is terminated by null character('\0') .
- The predefined functions gets() and puts() in C language to read and display string respectively.
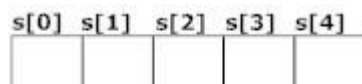
Declaration of strings:

Strings are declared in C in similar manner as arrays. Only difference is that, strings are of char type.

Syntax:

char str_name[str_size];

Example:

char s[5];

s[0]	s[1]	s[2]	s[3]	s[4]

Initialization of strings

char c[]="abcd";

c[0]	c[1]	c[2]	c[3]	c[4]
a	b	c	d	\0
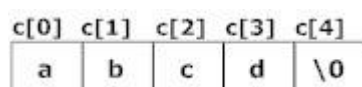
String handling functions

- Many library function are defined under header file <string.h> to perform different tasks.
- Diffrent user defined functios are:
  - ➢ **Strlen()**
  - ➢ **Strcpy()**
  - ➢ **Strcmp()**
  - ➢ **Strcat()**

Strlen():

- The strlen( ) function is used to calculate the length of the string.
- It means that it counts the total number of characters present in the string which includes alphabets, numbers, and all special characters including blank spaces.

Example:
```
char str[] = "Learn C Online";
int strLength;
strLength = strlen(str);    //strLength contains the length of the string i.e. 14
```

Strcpy()

- strcpy function copies a string from a source location to a destination location and provides a null character to terminate the string.

Syntax:
```
strcpy(Destination_String,Source_String);
```
Example:
```
char *Destination_String;
char *Source_String = "Learn C Online";
strcpy(Destination_String,Source_String);
printf("%s", Destination_String);
```
Output:
```
Learn C Online
```

Strcmp()

- Strcmp() in C programming language is used to compare two strings.
- If both the strings are equal then it gives the result as zero but if not then it gives the numeric difference between the first non matching characters in the strings.

Syntax:
```
int strcmp(string1, string2);
```
Example:
```
char *string1 = "Learn C Online";
char *string2 = "Learn C Online";
int ret;
ret=strcmp(string1, string2);
printf("%d",ret);
```
Output:
```
     0
```

Strcat()

- The strcat() function is used for string concatenation in C programming language. It means it joins the two strings together

Syntax:
```
strncat(Destination_String, Source_String,no_of_characters);
```
Example:
```
char *Destination_String="Visit ";
char *Source_String = "Learn C Online is a great site";
strncat(Destination_String, Source_String,14);
puts( Destination_String);
```
Output:
```
Visit Learn C Online
```

Program to check wheaher a word is polyndrome or not
```
#include<stdio.h>
#include<string.h>
void main()
{
        char s1[10], s2[10];
        int x;
        gets(s1);
        strcpy(s2,s1);
        strrev(s1);
        x=strcmp(s1,s2);
        if(x==0)
            printf("pallindrome");
        else
        printf("not pallindrome");
        getch();
}
```

Pointers in C
A pointer is a variable containing the address of another variable.
Declaring Pointer Variables
- Like ordinary variables, pointer variable must be declared before they are used.
  Declaration form/Syntax
          datatype *pointer_variable;
- The * tells the compiler that this is pointer variable.
- Pointer_variable will point to the given datatype.

Example:
    int *iptr;

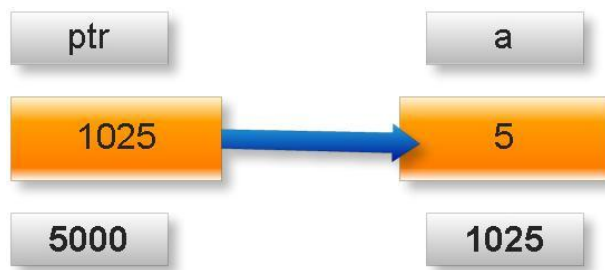Meaning of following simple pointer declaration and definition:
                int a=5;
                int * ptr;
                ptr=&a;
About variable a:
1. Name of variable : a
2. Value of variable which it keeps: 5
3. Address where it has stored in memory : 1025 (assume)
About variable ptr:
4. Name of variable : ptr
5. Value of variable which it keeps: 1025
6. Address where it has stored in memory : 5000 (assume)

ptr	a
1025	5
5000	1025

POINTER EXPRESSIONS AND POINTER ARITHMETIC

- Pointer variables can also be used in expressions. For ex,
  ```
  int num1=2, num2= 3, sum=0, mul=0, div=1;
  int *ptr1, *ptr2;
  ptr1 = &num1, ptr2 = &num2;
  sum = *ptr1 + *ptr2;
  mul = sum * *ptr1;
  *ptr2 +=1;
  div = 9 + *ptr1/*ptr2 - 30;
  ```

- A limited number of arithmetic operations can be perform on pointers. These operations are:

  - ➢ Increment and decrement
  - ➢ Addition and subtraction
  - ➢ Comparison
  - ➢ Assignment

- The increment (++) operator increases the value of a pointer by the size of the data object the pointer refers to. For example, if the pointer refers to the second element in an array, the ++ makes the pointer refer to the third element in the array.

Structures in C

- Structures are user defined data types which is a collection of heterogeneous data.
- It can have integer, float, double or character data in it.
- A structure is declared using the keyword struct followed by a structure name.
- All the variables of the structures are declared within the structure.

syntax.
```
struct struct_name
{
        structure element 1;
        structure element 2;
        ----------
        ----------
        structure element n;
}struct_var_name;
```
Example:
```
struct student
{       int r_no;
char name[20];
char course[20];
float fees;
}s1;
```

Difference Between Array and Stucture :

Array	Structure
Array is a collection of homogeneous data.	Stucture is a collection of heterogeneous data.
Array elements are referred by subscript.	Structure elements are referred by its unique name.

Array elements are accessed by it's position or subscript.	Stucture elements are accessed by its object as '.'(dot) operator.
Array is a derived data type.	Structure is user defined data type.

Syntax
<data_type> array_name[size];
	Syntax
	struct struct_name
	{
	    structure element 1;
	    structure element 2;
	            ---------- 	
	            ---------- 	
	            structure element n;
	}struct_var_nm;

Example:
int rno[5];
	Example:
	struct item_mst
	{
	    int rno;
	    char nm[50];
	}item1;

Unions in C

A union, is a collection of variables of different types, just like a structure. However, with unions, we
can only store information in one field at any one time.
DECLARING A UNION
 Syntax:
    union [*union-type-name]*
    {
            type variable-names;
            type variable-names;
            …
            ...
    }[union variable];
 Example:
    union item
     {
            int m;
            float x;
            char c;
     } code;

Difference between structure and union

Structure	Union
We can access all the members of structure at anytime.	Only one member of union can be accessed at anytime.
Memory is allocated for all variables.	Allocates memory for variable which variable require more memory.
All members of structure can be initialized	Only the first member of a union can be initialized.
'struct' keyword is used to declare structure.	'union' keyword is used to declare union.

Syntax
struct struct_name
{
   structure element 1;
   structure element 2;
      ---------- 	
      ---------- 	
   structure element n;
}struct_var_nm;
	Syntax
	union union_name
	{
	   union element 1;
	   union element 2;
	      ---------- 	
	      ---------- 	
	   union element n;
	}union_var_nm;