MODULE 1

INTRODUCTION TO COMPUTER

Definition: Computer is an electronic device which accepts input, processes data, stores information and produces output.

Data: Raw facts/ figures

Information: Processed data

For example, if 12-12-92 is the date of birth of a student, then it is data (a raw fact/figure). However, when we process this data (subtract it from the present-date)and say that the age of the student is 23 years, then the outcome is information,

- A computer is an electronic device that performs calculations and operations based on instructions provided by a software or hardware program. It has the ability to accept data (input), process it, and then produce outputs.
- Computers can also store data for later uses in appropriate storage devices, and retrieve whenever it is necessary.
- Modern computers are electronic devices used for a variety of purposes ranging from browsing the web, writing documents, editing videos, creating applications, playing video games, etc.
- They are designed to execute applications and provide a variety of solutions by combining integrated hardware and software components.

CHARACTERISTICS OF COMPUTERS

Computer is an electronic device that performs a function based on a given set of instructions known as a program.

A computer accepts data, processes it, and produces information.

These days, computers have become a crucial part of our everyday lives, and we need computers just like we need televisions, telephones, or other electronic devices at home. Computers are basically meant to solve problems quickly and accurately,.

The characteristics possessed by computers can be listed as follows:

1. **Speed:** Computer has the speed to process all computations in fractions of seconds.
2. **Storage and Reliability**: It can store huge amount of information easily and the computed results
3. are precise and reliable.
4. **Accuracy**: The computer is capable of providing accurate results of the computations.
5. **Versatility**: Determines that the computers can be used in all areas, such as in the field teaching,
6. **Training**, entertainment, accounting, hospitals etc.
7. **Diligence**: Determines that for how many hours computer can work without getting tired, and gives an acceptable output even after running for a long time.
8. **Automation**: Computers can perform a task without user intervaention.
9. **No IQ**: Although, it is the trend to make computers intelligent using Artificial Intelligence concept, they still do not have decision making abilities of their own. i.e, they need user guidance to perform tasks. Hence, they have no IQ.
10. **Economical:** Use of computers reduces man power requirements and they are efficient also. Hence, they save time, energy and money. Example: Sending E-mails

STORED PROGRAM CONCEPT

The following are the key characteristic features of this concept:

- Before any data is processed, instructions are read into memory.
- Instructions are stored in the computer's memory for execution.
- Instructions are stored in binary form (using binary numbers—only 0’ss and 1’s),
- Processing starts with the first instruction in the program, which is copied into a control unit circuit. The control unit executes the instructions.
- Instructions written by the users are performed sequentially until there is a break in the current flow.
- Input/Output and processing operations are performed simultaneously, While data is being read/written, the central processing unit (CPU) executes another program in the memory that is ready for execution,

HISTORY OF COMPUTERS

Early computers were designed not for entertainment but for solving number-crunching problems. These computers were punch-card based computers that took up entire rooms.

Today, our smartphones have much more computing power than that was available in those early computers.

GENERATION OF COMPUTER

The computer of each generation is smaller, faster and more powerful than preceding generation. There are five computer generations,

1. **First Generation (1942-1955)**
    - Hardware Technology First generation computers were manufactured using thousands of vacuum tubes. A vacuum tube is a device made of fragile glass. Memory Electromagnetic relay was used as primary memory and punched cards were used to store data and instructions.
    - Software Technology – Program was done in machine or assembly language.
    - They were the fastest calculating device of those times.
    - Computers were too bulky and required a complete room for storage.
    - Highly unreliable as vacuum tubes emitted a large.
    - amount of heat and burnt frequently
    - Required air-conditioned rooms for installation
    - Costly
    - Difficult to use
    - Required constant' maintenance because vacuum
    - tubes used filaments that had limited life time.
    - Therefore, these computers were prone to frequent
    - hardware failures

    Examples ENIAC, EDVAC, EDSAC, UNIVAC 1, IBM 701

2. **SECOND GENERATION (1955-1964)**
    - Hardware Technology Second generation computers were manufactured using transistors. Transistors were reliable, powerful, cheaper, smaller, and cooler than vacuum tubes.

- Memory Magnetic core memory was used as primary memory; magnetic tapes and magnetic disks were used to store data and instructions. These computers had faster and larger memory than the first generation computers. Software Technology Programming was done in high level programming languages. Batch operating system was used.
- Used for Scientific and commercial applications.
- Faster, smaller, cheaper, reliable, and easier to use than the first generation computers.
- They consumed 1/10th the power consumed by generation computers
- Bulky in size and required a complete room for its installation
- Dissipated less heat than first generation computers but still required air-conditioned rooms
- Costly
- Difficult to use

Examples Honeywell 400, IBM 7030, CDC UNIVAC LARC

3. **THIRD GENERATION (1964-1975)**

- Hardware Technology Third generation computers were manufactured using integrated chips (ICs). ICs consist of several components such as transistors, capacitors, and resistors on a single chip to avoid wired interconnections between components, Minicomputers came into existence,
- Memory Larger magnetic core memory was used as primary memory; larger capacity magnetic tapes and magnetic disks were used to store data and instructions. Software Technology Programming was done in high level programming languages such as FORTRAN, COBOL, Pascal, and BASIC. Time sharing operating system was used. Software was separated from the hardware. This allowed users to invest only in the software they need.
- Used for Scientific, commercial, and interactive online applications
- Faster, smaller, cheaper, reliable, and easier to use than the second generation computers.
- They consumed less power than second generation computers.
- Bulky in size and required a complete room for installation .
- Dissipated less heat than second generation computers but still required air-conditioned rooms.

- Costly.
- Easier to use and upgrade.

Examples 1BM 360/370, PDP-8, PADP-11, CDC6600

4. FOURTH GENERATION (1975-1989)

- The Microprocessor was the most important component. With the help of LSI (Large Scale Integration) and VLSI (Very Large Scale Integration) the entire CPU is on a single chip. The size was reduced and the speed was increased.
- Faster, smaller, cheaper, powerful, reliable, and easier to use than the previous generation computers

Examples 1BM PC. Apple Il, TRS-80, VAX 9000, CRAY1, CRAY-2, CRAY-X/MP

5. FIFTH GENERATION (1989-PRESENT)

- Artificial Intelligence and use of natural languages are the main features of this generations. These systems are expected to interact with users in natural language. Speech recognition and speech output should also be possible. Computers must be able to perform parallel processing. The quad-core and octa-core was also introduced. Neural networks and expert systems have been developed.
- Faster, smaller, cheaper, powerful, reliable, and easier to use than the previous generation computers.
- Speed of microprocessors and the size of memory are growing rapidly.
- High-end features available on mainframe computers in the fourth generation are now available on the microprocessors.
- They consume less power than computers of prior generations.
- Air-conditioned rooms required for mainframes and supercomputers but not for microprocessor.

Examples 1BM_ notebooks, Pentium PCs, SUM workstations, IBM SP/2,

Param supercomputer.

TYPES OF COMPUTERS

Different Types of computers: Digital Computers, Analog Computers and Hybrid computers.

a) Digital Computers:

- They are based on the principle of counting.
- They process information in discrete forms by using binary digits.
- They display information in the form of text, graphics or pictures by using 0s and 1s internally.
- Used in scientific and engineering applications.

b) Analog Computers

- They are based on the principle of measurement.
- They store data in continuous form of physical quantities.
- They are excellent to be used in situations that require data to be measured directly.
- Ex: Devices used to measure current flow, temperature, blood pressure, heart beats, voltage etc.

c) Hybrid Computers

- They are the combination of Digital and Analog computers.
- They use the best features of Digital and Analog computers.
- i.e they combine the speed of analog computers and the memory and accuracy of digital computers.
- They are used mainly in specialized applications where both analog and digital processing is involved like in Petrol bunks, hospitals etc.

CLASSIFICATION OF COMPUTERS

Computers can be broadly classified into four categories based on their speed, amount of data that they can process, and price. These categories are as follows:

1. Supercomputers
2. Mainframe computers
3. Minicomputers
4. Microcomputer

Apart from being classified by generations, computers can also be categorized by their size. The size of acomputer is often an indirect indication of its capabilities.

- **Supercomputers:** These are huge machines having most powerful and fastest processors. It uses multiple CPUs for parallel data processing. Speeds are measured in flops (floating point operations per second). The fastest operates at 34 petaflops. They are used for weather forecasting, analysis of geological data. They have enormous storage, uses more power and generate lot of heat. They are used by government agencies.
- **Mainframes:** These are multi-user machines that support many users using the feature of time sharing. It can run multiple programs even with a single CPU. The processor speed is measured in MIPS (Million instructions per second). It is used to handle data, applications related to organization and online transactions in banks, financial institutions and large corporations.
- **Minicomputers/Midrange computers:** It was introduced by DEC (Digital Equipment Corporation). They can serve hundreds of users and are small enough to partially occupy a room. They are used in smaller organizations or a department of a large one. They are not affordable to be used in home.
- **Microcomputers:** The microcomputer or PC is introduced by Apple and endorsed by IBM. This is a single-user machine powered by a single-chip microprocessor. They are very powerful machines having gigabytes of memory. They are both used in standalone mode and in a network. A microcomputer takes the form of desktop, notebook (laptop) or a netbook (smaller laptop). PCs today are powered by 3 types of OS – windows (7, 8 or 10), Mac OS X (Apple) and Linux. They are used for engineering and scientific applicationsand for software development.

APPLICTAION OF COMPUTER

.1.Word Processing
2. Digital video and audio composition
3. Desktop publishing
4. e-Business
5. Bioinformatics
6. Health care
7. Geographic Information System and Remote Sensing
8. Meteorology

9. Multimedia and Animation
10. Legal System
11. Retail Business
12. Sports
13. Travel and Tourism
14. Simulation
15. Astronomy
16. Education
17. Industry and Engineering
18. Robotics
19. Decision Support Systems
20. Expert System

BASIC ORGINATION OF A COMPUTER

A computer is an electronic device that performs five major operations:

1. Accepting data or instructions (input)
2. Storing data
3. Processing data
4. Displaying results (output).
5. Controlling and coordinating all operations inside a computer In this section, we will discuss all these functions and see how one unit of a computer interacts with another to perform these operations, which shows the interaction between the different units of a computer system.,

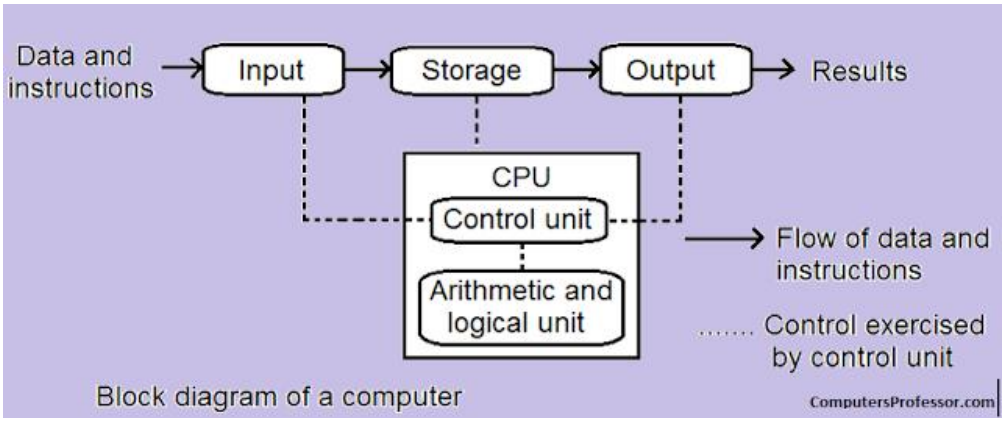


Figure: Block diagram computer

- **INPUT**: This is the process of entering data and instructions (also known as programs) into the computer system. The data and instructions can be entered by using different input devices such as keyboard, mouse, scanner, and trackball. Note that computers understand binary language, which consists of only two symbols (0 and 1), so it is the responsibility of the input devices to convert the input data into binary codes.
- **STORAGE:** Storage is the process of saving data and instructions permanently in the computer so that they can be used for processing, The computer storage space not only stores the data and programs that operate on that data but also stores the intermediate results and the final results of processing,

  A computer has two types of storage areas:

  Primary storage - Primary storage, also known as the main memory, is the storage area that is directly accessible by the CPU at very high speeds. Primary storage space is very expensive and therefore limited in capacity. Another drawback of main memory is that it is volatile in nature; that is, as soon as the computer is switched off, the information stored gets erased. Hence, it cannot be used as a permanent storage of useful data and programs for future use. An example of primary storage is random access memory (RAM).

  Secondary storage - Also known as auxiliary memory, this memory is just the opposite of primary memory, It overcomes all the drawbacks of the primary storage area. It is cheaper, non-volatile, and used to permanently store data and programs of those jobs that are not being currently executed by the CPU. An example is the magnetic disk used to store data, such as C and D drives, for future use.
- **OUTPUT :** Output is the process of giving the result of data processing to the outside world. The results are given through output devices such as monitor, and printer.

Since the computer accepts data only in binary form and the result of processing is also in binary form, the result cannot be directly given to the user.

The output devices, therefore, convert the results available in binary codes into a human-readable language before displaying it to the user.

Control The control unit (CU) is the central nervous system of the entire computer system. It manages and controls all the components of the computer system. It is the CU that decides the manner in which instructions will be executed and operations performed.

It takes care of the step-by-step processing of all operations that are performed in the computer.

- **Note** that the CPU is a combination of the arithmetic logic unit (ALU) and the CU.

- **PROCESSING :** The process of performing operations on the data as per the instructions specified by the user (program) is called processing.

Data and instructions are taken from the primary memory and transferred to the ALU, which performs all sorts of calculations.

The intermediate results of processing may be stored in the main memory, as they might be required again.

When the processing completes, the final result is then transferred to the main memory. Hence, the data may move from main memory to the ALU multiple times before the processing is over.

LAB SESSION—INSIDE THE COMPUTER

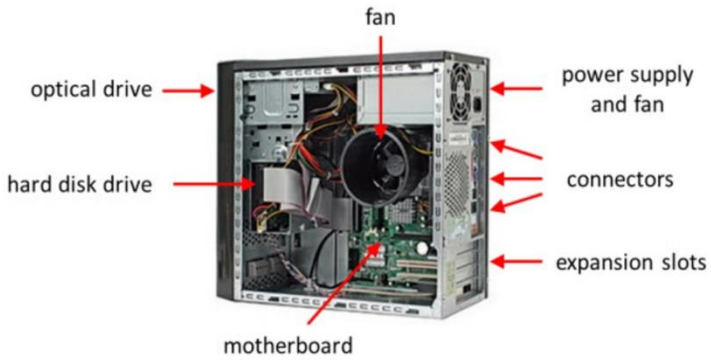


Figure: Computer Case and its parts

The following are some of the major parts of the computer:

- **CPU** : The CPU is the brain of the computer. It performs all calculations and controls the devices connected to the computer system. The faster the CPU, the quicker programs can process the instructions.
- **RAM:** A fast CPU is of no use if the computer does not have sufficient RAM. It is the computer's memory which stores information used by applications that are currently being executed by the CPU..
- **Hard disk drive (HDD):** The HDD of the computer is the secondary memory of the computer system where information is stored permanently. All types of data,

documents, and programs are stored on the hard disk. The larger the hard disk, the more the amount of data that can be stored on the drive.

- **Video card**: The video card is a board that plugs into the motherboard of the computer and generates images for display.
- **Sound card** : As with video cards, sound cards are expansion boards that are used to enable a computer to manipulate sound. For example, sound cards allow the users to plug in speakers and a microphone.
- **Modem :** A modem (modulator-demodulator) is a device that enables the computer to use a telephone line to communicate and connect to the Internet,
- **Network card** : A network card is used to connect the computer cither to other computers or to the Internet.
- **Fans :** There are one or more fans inside the computer to keep the air moving and the computer cool.
- **Cables :** There are multiple wires inside the computer that are flat, ribbon-like cables. They are used to provide power and communication to the various parts inside the computer.

MOTHERBOARD

The motherboard, also known as the mainboard or the parent board is the primary component of a computer. It is used to connect all the components of the computer. The motherboard is a printed circuit that has connectors for expansion cards, memory modules, the processor, etc.

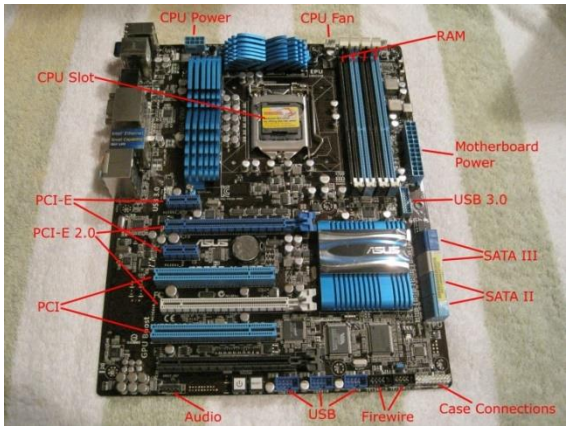


FIGURE : Computer's Motherboard

CHARACTERISTICS OF A MOTHERBOARD

A motherboard can be classified depending on the following characteristics:

- Form factor - Form factor refers to the motherboard's geometry, dimensions, arrangement, and electrical requirements.
- Chipset - The chipset is an electronic circuit that basically coordinates data transfers between the different components of the computer (such as the processor and memory). In order to enhance the computer's upgradeability, one must choose a motherboard that has the latest chipset integrated in it.
- Integrated components - Some of the motherboard's components are integrated into its printed circuitry. These include the following:

- The chipset is a circuit that controls the majority of the computer's resources such as the bus interface with the processor, cache memory, RAM, and expansion cards.
- CMOS clock and battery
- BIOS
- System bus and expansion bus

- CMOS clock and battery - The real-time clock (or RTC) is 4 circuit that is used to synchronize the computer's signals. When the computer is switched off, the power supply stops providing electricity to the motherboard. You must have observed that when we turn on the system, it always displays the correct time. This is because an electronic circuit, called the complementary metaloxide semiconductor (CMOS) chip, saves some system information, such as the time, date, and other essential system settings,
- Type of processor socket used - The processor (also called the microprocessor) is the brain of the computer, The processor is characterized by its speed or frequency, which is the rate at which it executes instructions.

  The slot on the motherboard into which the processor is inserted is called the processor socket or slot.
- Input-Output connectors - The basic input/output system (BIOS) is an interface between the operating system and the motherboard. The BIOS is stored in the read-only memory (ROM), which cannot be rewritten. The BIOS uses data stored in the CMOS chip to know about the system's hardware configuration. To enter BIOS setup, the user must press the DEL key. Fl and F2 keys can also be used.

- RAM connectors - RAM is the primary storage area that stores data while the computer is running.
- Expansion slots - Expansion slots are compartments into which expansion cards can be inserted. Such curds render new features or enhance the computer's performance. For example, the AGP slot (also known as Accelerated Graphic Port) is a fast port used for graphics cards.

INPUT AND OUTPUT DEVICES

INPUT DEVICES

An input device is used to feed data and instructions into ‘a computer, In the absence of an input device, a computer would have only been a display device, In this section, we will read about some of the widely used input devices,

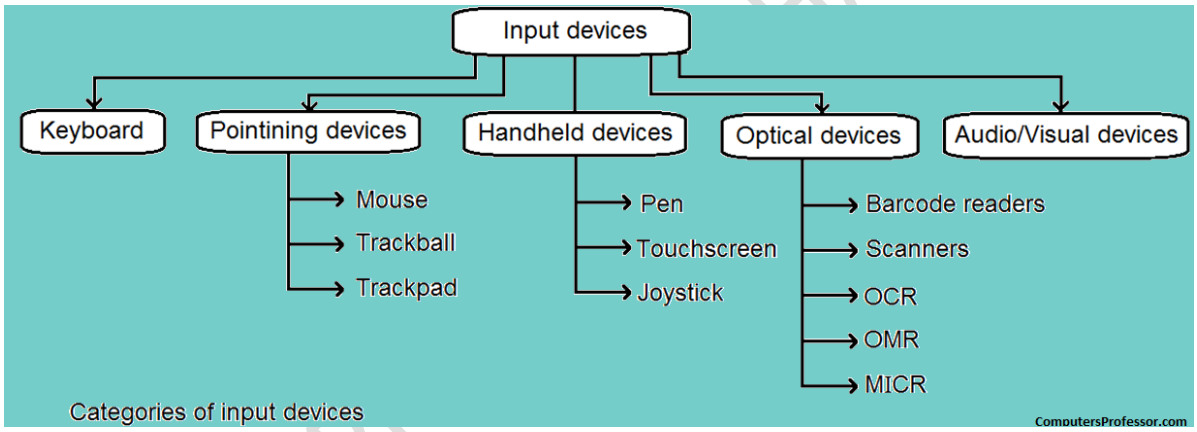


Figure : Categorizes input devices into different groups.

- **KEYBOARD** - The keyboard is the main input device for computers. Computer keyboards look very similar to the keyboards of typewriters, with some additional keys.
  - Using a keyboard, the user can type & document, use keystroke shortcuts, access menus, play games, and perform numerous other tasks. Most keyboards have between 80 and 110 keys, which include the following:

Typing keys - These include the letters of the alphabet. The layout of the keyboard is known as QWERTY for its first six letters. The QWERTY pattern has been a standard right from the time computer keyboards were introduced.

Numeric keys - These include a set of keys, arranged in the same configuration found on calculators to speed up data entry of numbers. When the Num Lock key is set to ON, the user can type numbers, dot, or input the symbols /, *, ~, and +. When the Num Lock key is set to OFF, the numeric keys can be used to move the cursor on the screen,

Function keys - These are used by applications and operating systems to input specific commands, They are often placed on the top of the keyboard in a single row. Function keys can be programmed so that their functionality varies from one program to another,

Control keys - These are used to handle control of the cursor and the screen. Four arrow keys are arranged input and Output Devices.

Hard copy devices an inverted T-type fashion between the typing and the numeric keys, and are used to move the cursor on the screen in small increments.

In addition to the arrow keys, there are other cursor keys (or navigational keys), such as:

- + Home and End to move the cursor to the beginning and end of the current line, respectively
- Page Up and Page Down to move the cursor up and down by one screen at a time, respectively.
- + Insert to enter a character between two existing characters
- + Delete to delete a character at the cursor position

Other common control keys on the keyboard include Control (Ctrl), Alternate (Alt), Escape (Esc), Print Screen, Pause, the Windows or Start key (Microsoft Windows logo), and a shortcut key. The shortcut key is used to access the options available by pressing the right mouse button. The Esc key cancels the selected option, and the Pause key suspends a command/process in progress. Finally, the Print Screen key captures everything on the screen as an image. The image can be pasted into any document.

➢ POINTING DEVICES

A pointing input device enables the users to easily control

- the movement of the pointer to select items on a display screen,
- to select commands from commands menu, to draw graphics, etc.

Some examples of pointing devices include mouse, trackball, light pen, joystick, and touchpad.

- **Mouse** - The mouse is an input device that was invented by Douglas Engelbart in 1963. It is the key input device used in a graphical user interface (GUI). It can be used to handle the pointer easily on the screen to perform various functions such as opening a program or file. The earliest form has a rotating ball and two buttons. The left button is used for selecting (by clicking once) and execute (by clicking twice). The right button is used to check and change attributes. The optical mouse uses infrared laser/LED, the wireless mouse uses radio frequency technology
  - ✓ *Point* - Placing the pointer over the word or the object on the screen by moving the mouse on the desk is termed as pointing.
  - ✓ *Drag* - Dragging means pointing to a desired location while pressing the left button.
  - ✓ *Scroll* - The scroll wheel, which is placed in between the left and right buttons of the mouse, is used to vertically scroll through long documents,

    Some of the popular mouse types are as follows:

  - ❖ Mechanical mouse - This type of mouse has a rubber or metal ball at its bottom and an electronic circuit containing sensors, When the mouse is moved over a flat surface, the sensors detect the direction of movement of the ball. The electronic circuit translates the movement into signals and feeds it as input to the computer.
  - ❖ Optical mouse - The optical mouse is more advanced than the mechanical mouse, It contains a ball inside. The movement of the mouse is detected using laser technology, by using optical sensors.
  - ❖ Cordless mouse - A cordless or wireless mouse is not connected to the computer. The movement of the mouse is detected using radio waves or infrared light waves.
- **Trackball** - is a pointing device that is used to control the position of the cursor on the screen, It is usually used in notebook computers, where it is placed on the keyboard. The trackball is nothing but an upside-down mouse where the ball rotates in place within a socket. The user rolls the ball to position the cursor at an appropriate position on the screen and then clicks one of the buttons
- **Touchpad** - A touchpad (or trackpad) is a small, flat, rectangular stationary pointing device with a sensitive surface of 1.5—2 square inches, Touchpads are widely used in laptops, and are in built on the laptop keyboards, They can also be attached to a PC or

be used with personal digital assistants (PDAs) and iPods. The working of a touchpad is similar to that of a mouse or a trackball.

HANDHELD DEVICES

A handheld device is a pocket-sized computing device with a display screen and touch input and/or a miniature keyboard, Some common examples of handheld devices include smartphones, PDAs, handheld game consoles, and portable media players (c.g., iPods).

- ✓ Joystick - A joystick is a cursor control device widely used in computer games and computeraided design (CAD) computer-aided manufacturing (CAM) applications.
- ✓ Stylus - A stylus is a pen-shaped input device used to enter information or write on the touchscreen of a handheld device.
- ✓ Touchscreen - A touchsereen is a display screen that can identify the occurrence and position of a touch inside the display region. The user can touch the screen either by using a finger or a stylus.

➢ OPTICAL DEVICES

Optical devices, also known as data-scanning devices, use light as a source of input for detecting or recognizing different objects such as characters, marks, codes, and images. These devices convert these objects into digital data and send it to the computer for further processing. Some optical devices that are discussed in this section include barcode readers, image scanners, optical character recognition (OCR) devices, optical mark readers (OMR), and magnetic ink character recognition (MICR) devices.

- ✓ Barcode Reader - A barcode reader is a handheld input device that is used to capture and read information stored in a barcode. It consists of a scanner, a decoder, and a cable used to connect the reader to a computer,
- ✓ Image Scanner - A scanner is a device that captures images, printed text, and handwriting, from different sources such as photographic prints, posters, and magazines and converts them into digital images for editing and display on computers,

- ✓ Optical Character Recognition (OCR) Device - Optical character recognition is the process of converting printed materials into text or word processing files that can be easily edited and stored.
- ✓ Optical Mark Recognition (OMR) Device - Optical mark recognition is the process of electronically extracting data from marked fields, such as checkboxes and fill-in fields, on printed forms.
- ✓ Magnetic Ink Character Reader - Magnetic ink character reader (MICR) is used to verify the legitimacy of paper documents, especially bank checks, It consists of magnetic ink printed characters that can be recognized by high-speed magnetic recognition devices The printed characters provide important information (such as cheque number, bank Pay DATE: as) RUPEES
- ✓ Magnetic ink character recognition - A cheque containing magnetic ink characters printed on it routing number, account number, and, in some cases, the amount on the cheque) for processing to the receiving party.

AUDIOVISUAL INPUT DEVICES

All computers are multimedia-enabled that is, computers not only allow one to read or write text, but also enable the user to record songs, view animated movies, etc. Hence, in addition to having a keyboard and a mouse, audio-video devices have become a necessity today.

- Audio Devices - Audio devices are used to either capture or create sound. They enable computers to accept music, speech, or sound effects for recording and/or editing.
- Video Input Devices - Video input devices are used to capture video from the outside world into the computer. Here, the term video means moving picture along with sound (as in television).

OUTPUT DEVICES

Any device that outputs/gives information from a computer can be called an output device.

- The output devices are electromechanical devices that accept digital data (in the form of 0s and 1s) from the computer and convert them into human-understandable language.
- Computers are multimedia-enabled, the information from computers is usually output in cither visual or auditory format.
- Monitors and speakers are two widely used output devices. These devices provide instant feedback to the user's input.
  For example, monitors display characters as they are typed.
- Similarly, speakers play a song instantly when the user selects one from a playlist. Other examples of output devices include printers, plotters, and projectors.

➢ CLASSIFICATION OF OUTPUT DEVICES

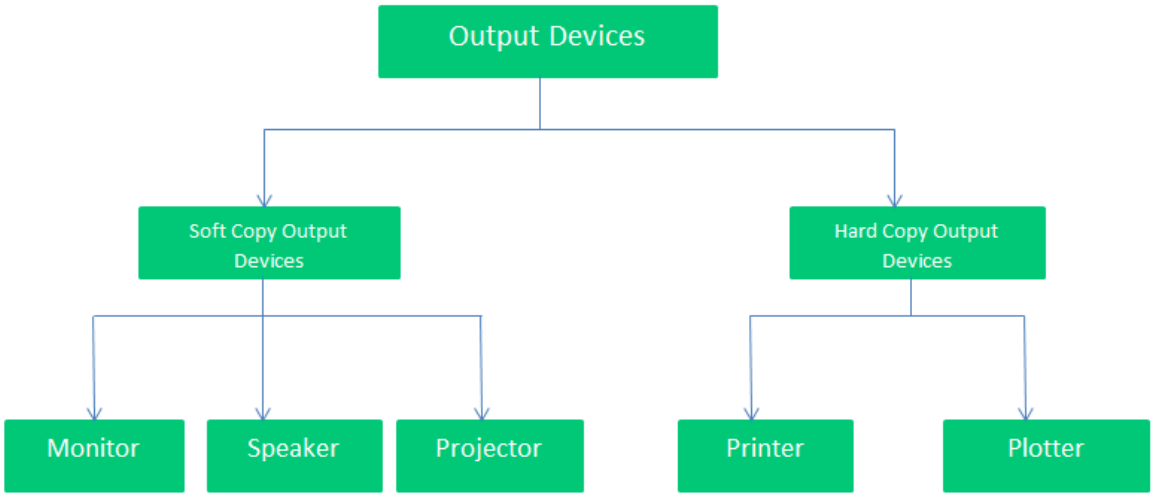


Fig: Classification of Output Device

❖ **Soft Copy Devices -** Soft copy output devices are those that produce an electronic version of an output—for example, a file that is stored on a hard disk, CD, or pen drive—and is displayed on the computer screen (monitor).

Features of a soft copy output include the following:

- The output can be viewed only when the computer is on.
- The user can easily edit soft copy output.

- Soft copy cannot be used by people who do not have a computer.
- Scarching for data in a soft copy is easy and fast.
- Electronic distribution of material as soft copy is cheaper, It can be done easily and quickly.

✓ Monitors

- The monitor is a soft copy output device used to display video and graphics information generated by the computer through the video card.
- Computer monitors are similar to television screens but they display information at a much higher quality.
- The monitor is connected to cither the VGA or the digital video interface (DVI) port on the video card (on the motherboard or separately purchased).
- Monitors come in three variants—cathode ray tube (CRT), liquid crystal display (LCD), and plasma.
- While CRT monitors look much like traditional televisions and are very huge in size, LCD monitors on the other hand are thinner, offering equivalent graphics quality.
- However, these days, LCD monitors are replacing CRT monitors as they are cheaper and occupy less space on the desk. Most monitors have a size range of 15’ to 21’ or more (where size is defined as a diagonal measurement from one corner of the screen to the other).

FIGURE ; Types Of Monitor

CRT MONITOR

- CRT monitor CRT monitors work by firing charged electrons at a phosphorus film.

- When electrons hit the phosphor-coated screen, they glow, thereby enabling the user to see the output.
- In a CRT, the cathode (negative terminal) is a heated filament that is placed in a vacuum created inside a glass tube.
- The ray is a stream of electrons that come out from a heated cathode into the vacuum, While electrons are negative, the anode, on the other hand, is positive, so it attracts the electrons coming out of the cathode.
- That is, the focusing anode focuses the stream of electrons to form a tight beam that is then accelerated by an accelerating anode.
- This tight, high-speed beam of electrons flies through the vacuum in the tube and hits the flat screen at the other end of the tube.
- This screen is coated with phosphor, which glows when struck by the beam, thereby displaying a picture, which the user sees on the monitor.

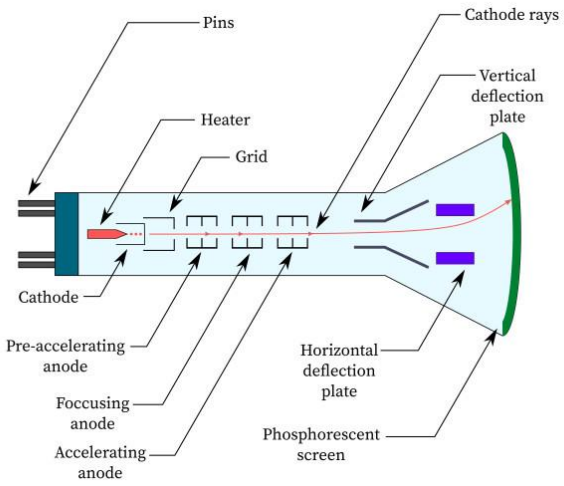


FIGURE : WORKING OF CRT MONITOR

Advantages

- CRT monitors provide images of good quality (bright as well as clear).
- CRT monitors are cheapest when compared to LCD and plasma monitors.
- The images are clear even when you try to view it from an angle.

Disadvantages

- CRT monitors occupy a large space on the desk.
- They are bigger in size and weight and therefore difficult to move from one place to another when compared with other types of monitors.
- Power consumption is higher than the other monitors.

LCD MONITOR

- An LCD monitor (shown in Figure 2.19) is a thin, flat, electronic visual display that uses the light modulating properties of liquid crystals, which do not emit light directly.
- LCD screens are used in a wide range of applications ranging from computer monitors, televisions, instrument panels, aircraft cockpit displays, signage, etc,, 10 consumer devices such as video players, gaming devices, clocks, watches, calculators, and telephones.
- LCD technology is based on the principle of blocking light.
- The LCD consists of two pieces of polarizing filters (or substrates) that contain a liquid crystal material between them.

- A backlight creates light, which is made to pass through the first substrate. Simultaneously, the electrical currents cause the liquid crystal molecules to align, thus allowing varying levels of light to pass through to the second substrate and create the colours, and hence images are seen on the screen.

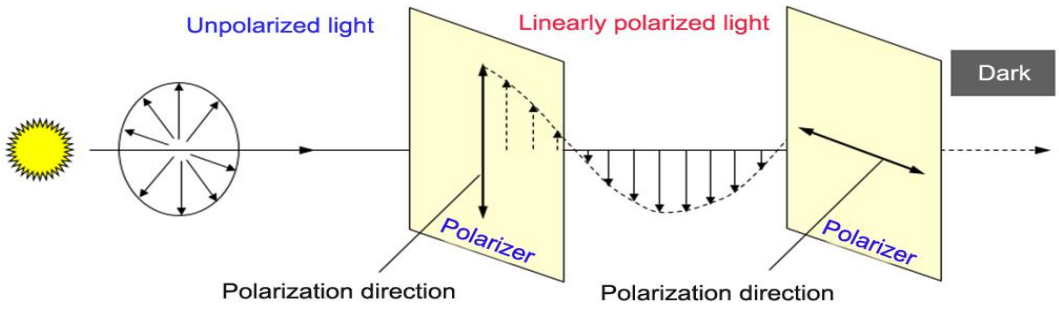


FIGURE : CRT MONITOR WORKING

Advantages

- LCD monitors are very compact and lightweight.
- They consume less power.
- They do not suffer from geometric distortion.
- There is little or no flicker of images (depending on the backlight technology used).
- They are more reliable than CRTs.
- They can be made in almost any size or shape.

- They cause less eye fatigue.

Disadvantages

- They are more expensive than CRTs.
- Images are not very clear when tried to view from an angle.

PLASMA MONITOR

Plasma monitors are thin and flat monitors widely used in televisions and computers. The plasma display contains two glass plates that have hundreds of thousands of tiny cells filled with xenon and neon gases. The address clectrode and the transparent display electrode are sandwiched between the glass plates. The display electrode is covered by a magnesium oxide protective layer and is arranged in horizontal rows along the screen, while the address electrodes are arranged in vertical columns, thereby forming a grid-like structure**.**

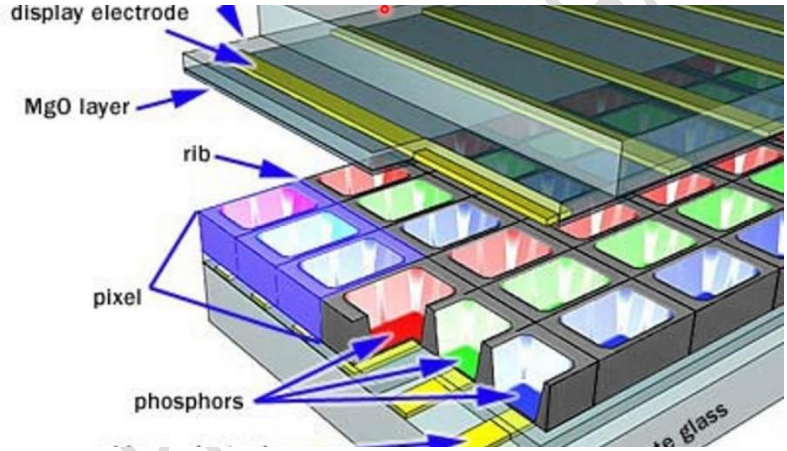


FIGURE : WORKING OF PLASMA MONITOR

- ‘To ionize the gas in a particular cell, the electrodes that intersect at that cell are charged at least thousands of times within a small fraction of a second (charging cach cell in turn).
- Once the intersecting electrodes are charged, an electric current begins to flow through the gas in the cell.
- The current creates a rapid flow of charged particles, thereby stimulating the gas atoms to release ultraviolet photons.
- When these UV photons hit a phosphor atom in the cell, one of phosphor’s electrons jumps to a higher energy level, and the atom heats up. When the electron falls back to its normal level, it release light photon,

Advantages

- The technology used in plasma monitors allows producing a very wide sereen using extremely thin materials,
- Very bright images are formed which look good from almost every angle.
- These monitors are not heavy and are thus easily portable.

Disadvantages

- These monitors are very expensive.
- They have a high power consumption.
- Since the images are phosphor-based, at times, they may suffer from flicker

PROJECTORS

- A projector is a device that takes an image from a video source and projects it onto a screen or another surface.
- These days, projectors are used for a wide range of applications, varying from home theater systems for projecting movies and television programs onto a screen much larger than even the biggest available television, to organizations for projecting information and presentations onto screens large enough for rooms filled with many people.
- Projectors also allow users to change/adjust some features of the image such as brightness, sharpness, and colour settings, similar to the features available in a standard television. Projectors are now available in a variety of different shapes and sizes, and are produced by many different companies.
- To display the image on a big screen, the projector first displays that image onto a small screen inside the projector itself, which is then projected onto the final scrcen using bright light and a lens. The lens is shaped in such a way that it takes the small image and turns it into a dramatically larger one.

FIGURE : PROJECTOR

Projectors can be broadly classified into two categories depending on the technology they use.

LCD projector - LCD projectors make use of their own light to display the image on the screen/wall. These proj are based on LCD technology. To use these projectors, the room must be first darkened, else the image formed will be blurred.

Digital light processing (DLP) projector - DLP projectors use a number of mirrors to reflect the light. When using the DLP projector, the room may or may not be darkened because it displays a clear image in both situations.

SPEAKERS

- Initially, computers were designed to be used only for scientific purposes, but later, with the advances in technology, computers became so popular and inexpensive that they are now used in almost every sphere of our day to-day lives.
- Today, all business and home users demand audio capabilities from their computers. For this purpose, speakers were developed in different sizes and shape: and with different powers and sound quality, With all these types of speakers, the user can enjoy music, movie, or a game, and the voice will be spread through the entire room.
- With good quality speakers, the voice will also be audible even to people sitting in another room or even to neighbours.
- Every computer has a built-in speaker, an external speaker disables this lower-fidelity built-in speaker. Speakers available in the market have a wide range of quality and prices.
- The normal computer speakers are small, plastic, and have mediocre sound quality. Other speakers are available that have equalization features such as bass and treble controls.
- Users can also use a lead to connect their computer's sound output to an existing stereo system to give much better results than the small, low-cost computer speakers.

FIGURE: AUDIO DEVICES

HARD COPY DEVICES

- Hard copy output devices are those that produce a physical form of output.
- For example, the content of a file printed on paper is a form of hard copy output. The features of hard copy output include
- A computer is not needed to see the output editing and incorporating the edits in the hard copy is difficult.
- Hard copy output can be easily distributed to people who do not have a computer.
- Searching for data in a hard copy is a tiring and difficult job.
- Distribution of hard copy is not only costly but slower as well.

PRINTERS

- A printer is a device that takes the text and graphics information obtained from a computer and prints it on a paper.
- Printers are available in the market in various sizes, speeds, sophistication, and costs, Usually, more expensive printers are used for higher-resolution colour printing.
  - ✓ Colour - Colour printouts are needed for presentations, maps, and other pages where colour is part of the information. Colour printers can also be set to print only in monochrome. These printers are more expensive, so if the users do not have a specific need for colour and usually take a lot of printouts, they will find a black-and-white printer cheaper to operate.
  - ✓ Resolution - The resolution of a printer sharpness of text and images rendered on paper.
  - ✓ Speed - Speed means number of pages that are printed in one minute. Colour printing is even slower.

- ✓ Memory - Most printers have a small amount of memory (for example, | MB), which can be expanded by the user. Having more memory enhances the speed of printing.

Printers can be broadly classified into two groups: impact and non-impact printers

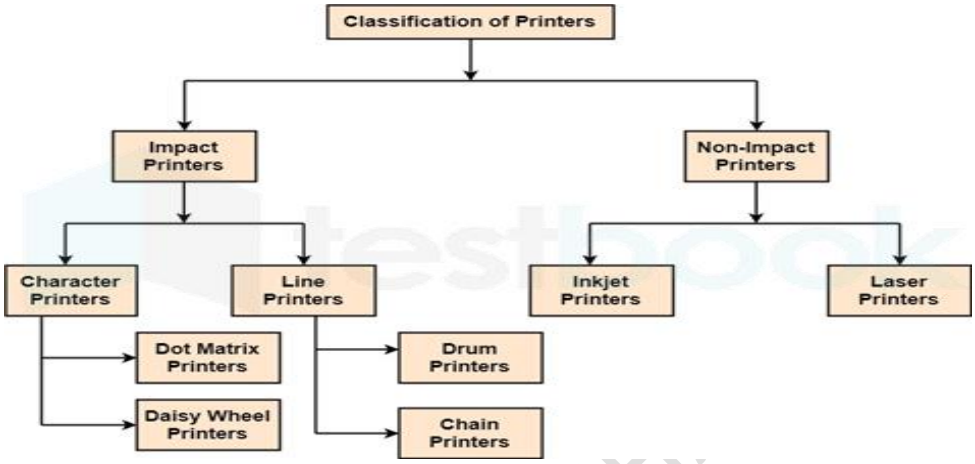


FIGURE : CLASSIFICATION OF PRINTERS

Impact Printer - These printers print characters by striking an inked ribbon against the paper. Examples of impact printers include dot matrix printers, daisy wheel printers, and most types of line printers

Advantages

- ✓ These printers enable the user to produce carbon copies.
- ✓ They are cheap.

Disadvantages

- ✓ Impact printers are slow.
- ✓ They offer poor print quality, especially in the case of graphics.
- ✓ They can be extremely noisy
- ✓ They can print only using the standard font

Dot matrix printer - A dot matrix printer prints characters and images of all types as a pattern of dots. This printer has a printhead (or hammer) that consists of pins representing the character or image. The printhead runs back and forth, or in an up-and-down motion on the page and prints by striking an ink-soaked cloth ribbon against the paper, much like the print mechanism of a typewriter.

Daisy wheel printer - Daisy wheel printers use an impact printing technology to generate high quality output comparable to typewriters, and are three times faster. However, today, daisy wheel technology is found only in some electronic typewriters.

The key benefit of using a daisy wheel printer is that the print quality is high, as the exact shape of the character hits the ribbon to leave an impression on the paper.

Line printer - A line printer is a high-speed impact printer in which one typed line is printed at a time. The speed of a line printer usually varies from 600 to 1200 lines per minute, or approximately 10-20 pages per minute. Because of their high speed, line printers are widely used in data centers and in industrial environments.

Non-impact printer - Non-impact printers are much quieter than impact printers, as their printing heads do not strike the paper. They offer better print quality, faster printing, and the ability to create prints that contain sophisticated graphics. Non-impact printers use cither solid or liquid cartridge-based ink, which is cither sprayed, dripped, or electrostatically drawn onto the page. The main types of non-impact printers are inkjet, laser, and thermal printers.

Advantages

- Non-impact printers produce prints of good quality, and hence render sophisticated graphics.
- They are noiseless.
- They are fast.
- They can print text in different fonts.

Disadvantages

- These printers are expensive.
- The ink cartridges used by them are also costly.
-

Inkjet printer - Inkjet printers have made rapid technological advances in recent years. The colour inkjet printers have succeeded in making colour printing an affordable option even for home users.

Laser printer - A laser printer is a non-impact printer that works at very high speeds and produces high-quality text and graphics. It uses the technology used in photocopier machines. When a document is sent to the printer, the following steps take place:

- ✓ A laser beam 'draws' the document on a drum (which is coated with a photo-conductive material) using electrical charges.
- ✓ After the drum is charged, it is rolled in a toner (a dry powder type of ink).
- ✓ The toner sticks to the charged image on the drum,
- ✓ The toner is transferred onto a piece of paper and fused to the paper with heat and pressure.
- ✓ After the document is printed, the electrical charge is removed from the drum and the excess toner is collected.

PLOTTERS

- A plotter is a printing device that is usually used to print vector graphics with high print quality.
- They are widely used to draw maps, in scientific applications, and computer-aided engineering (CAE). Architects use plotters to draw blueprints of the structures they are working on.
- A plotter is basically a printer that interprets commands from a computer to make line drawings on paper with one or more automated pens.
- Since plotters are much more expensive than printers, they are used only for specialized applications.

There are two different types of plotters are drum and flatbed.

Drum plotter –

- ✓ A drum plotter is used to draw graphics on paper that is wrapped around a drum. This type of plotter is usually used with mainframe and minicomputer systems.
- ✓ The drum plotter works by rotating the drum back and forth to produce vertical motion.
- ✓ The pen, which is mounted on a carriage, is moved across the width of the paper.
- ✓ Hence, the vertical movement of the paper and the horizontal movement of the pen create the required design under the control of the computer.

- ✓ Drum plotters can make multicolour drawings by using pens with different coloured inks. Moreover, drum plotters support very large plot sizes with paper widths of up to 1 meter,

Flatbed plotter –

- ✓ In a flatbed plotter, the paper is spread on the flat rectangular surface of the plotter, and the pen is moved over it.
- ✓ Flatbed plotters are less expensive, and are used in many small computing systems.
- ✓ The size of the plot is limited only by the size of the plotter's bed.
- ✓ In this type of plotter, the paper does not move; rather, plotting is done by moving an arm that moves a pen over the paper.

DESIGNING EFFICIENT PROGRAMS

- A programming paradigm is a fundamental style of programming that defines how the structure and basic elements of a computer program will be built.
- The style of writing programs and the set of capabilities and limitations that a particular programming language has depends on the programming paradigm it supports.
- Some programming languages strictly follow a single paradigm, others may draw concepts from more than one.

These paradigms, in sequence of their application, can be classified as follows:

Monolithic programming:

- Programs written using monolithic programming languages such as assembly language and BASIC consist of global data and sequential code.
- The global data can be easily accessed and modified (knowingly or mistakenly) from any part of the program, thereby posing a serious threat to its integrity.
- A sequential code is one in which all instructions are executed in the specified sequence
- Therefore, all the actions required to complete a particular task are embedded within the same application itself.
- This not only makes the size of the program large but also makes it difficult to debug and maintain.

- For all these reasons, monolithic programming language is used only for very small and simple applications where reusability ts not a concer.

Procedural Programming

- In procedural languages, a program is divided into subroutines that can access global data.
- To avoid repetition of code, each subroutine performs a well-defined task.
- A subroutine that needs the service provided by another subroutine can call that —_ subroutine. Therefore, with 'jump'. "goto', and 'call' instructions, the sequence of execution of instructions can be altered.

Advantages

- The only goal is to write correct programs.
- Programs are easier to write as compared to monolithic programming

Disadvantages

- No concept of reusability.
- Requires more time and effort to write programs Programs are difficult to maintain
- Global data is shared and therefore may get altered (mistakenly)

Structured Programming

- Structured programming, also referred to as modular programming, was first suggested by mathematicians, Corrado Bohm and Guiseppe Jacopini in 1966.
- It was specifically designed to enforce a logical structure on the program to make it more efficient and easier to understand and modify.
- Structured programming was basically defined to be used in large programs that require large development team to develop different parts of the same program.
- Structured programming employs a top-down approach in which the overall program structure is broken down into separate modules
- For large and complex programs, the overall program structure may further require the need to break the modules into subsidiary pieces.
- This process continues until an individual piece of code can be written easily.
- Almost every modern programming language similar to C, Pascal, etce.,

Advantages

- The goal of structured programming is to write correct programs that are easy to understand and change.
- Modules enhance programmers' productivity by allowing them to look at the big picture first and focus on details later.
- With modules, many programmers can work on a single, large program, with each working on a different module.
- A structured program takes less time to be written than other programs. Modules or procedures written for one program can be reused in other programs as well.
- Each module performs a specific task.
- A structured program is easy to debug.

Disadvantages

- Not data-centered
- Global data is shared and therefore may get inadvertently modified
- Main focus is on functions

Object-oriented Programming (OOP)

- The object-oriented paradigm is task-based and data-based.
- In this paradigm, all the relevant data and tasks are grouped together in entities known as objects.
- The procedural or structured programming paradigm considers this list as merely a collection of data.
- Any program that accesses this list must have some procedures or functions to process this list. For example, to find the largest number or to sort the numbers in the list, we need specific procedures or functions to do the task, Therefore, the list is a passive entity as it is maintained by a controlling program rather than having the responsibility of maintaining itself.
- However, in the object-oriented paradigm, the list and the associated operations are treated as one entity known as an object.

DESIGNING EFFICIENT PROGRAMS

Phases In Software Development Life Cycle (SDLC)

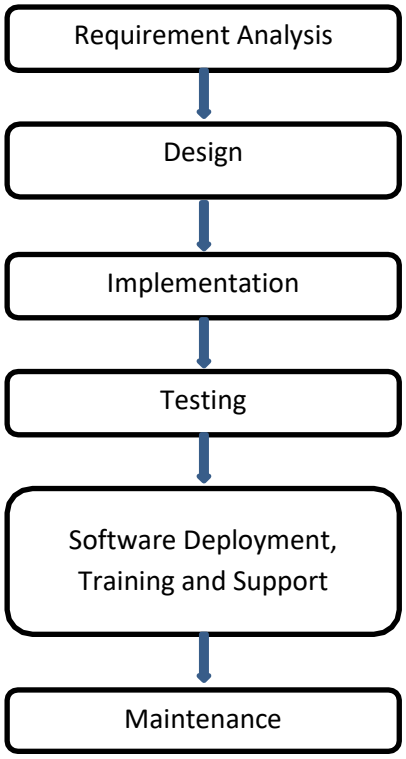


FIGURE : Phases in software development life cycle

1. Requirement Analysis

- In this phase, user's requirements are gathered to know why software is to be built.
- Every identified requirement is documented to avoid any doubts or uncertainty about the functionality of programs.
- The functionality, capability, performance and availability of hardware and software components will be analyses in this phase.

2. Design

- The requirements collected in the previous phase act as inputs to design phase.
- In this phase, a plan of actions is made before actual development process starts.
- In this phase, the core structure of the software is divided into modules.
- The solution of the problem is specified for each module in the form of algorithms or flowcharts
- This phase specifies how the software is built.

3. Implementation

- In this phase, the designed algorithms or flowcharts are converted into program code using any high level language.
- The program codes are tested by programmers to ensure correctness.
- This phase is also known as construction or code generation phase.
- While developing the code, the development team checks whether the software is compatible with the available hardware and other software components specified in the document provided in requirement specification phase.

4. Testing

- In this phase, all modules are together tested to ensure that the overall system works fine as a whole product.
- Though the individual modules are tested separately by the programmers in implementation phase, there are chances of errors creeping in during integration of modules to form overall software.
- In this phase, the software is tested using all possible varieties of data to ensure that the software works correctly as expected by the user‟s requirements specification of first phase.

5. Software Deployment, training and support

- Once the software is tested, and approved by users, it is installed or deployed in production environment.
- It is crucial to have training classes for the users of the software as a part of deployment phase.
- The support of the development team is essential in case of any issues occur during the use of software.

6. Maintenance

- It is an ongoing activity to cope up with new problems or new requirements.
- A new code may have to be added which was not considered during design phase.
- If the maintenance cost exceeds, sometimes it is better to build new code.

ALGORITHMS

In general terms, an algorithm provides a blueprint to writing a program to solve a particular problem. It is considered to be an effective procedure for solving a problem in a finite number of steps. That is, a well-defined algorithm always provides an answer, and is guaranteed to terminate.

- ✓ It is a step by step process to solve specific problem.
- ✓ An algorithm is composed of a finite set of steps each of which may require one or more operations.

Algorithms are mainly used to achieve software reuse Once we have an idea or a blueprint of a solution, we can implement it in any high-level language, such as C, C++, Java, and so on.

In order to qualify as an algorithm, a sequence of instructions must possess the following characteristics :

- Be precise
- Be unambiguous
- Not even a single instruction must be repeated infinitely
- After the algorithm gets terminated, the desired result must be obtained.

Control Structures Used In Algorithm

An algorithm he may involve decision speaking, an algorithm may structures, namely, sequence, decision.

Sequence - Sequence means that each step of the algorithm is executed in the specified order. This algorithm performs the steps in a purely sequential order.

Decision - Decision statements are used when the outcome of the process depends on some condition. For example, if x = y, then print “EQUAL”.

Repetition - Repetition, which involves executing one or more steps for a number of times, can be implemented using constructs such as the while, do-while, and for loops. These loops execute one or more steps until some condition is true.

FLOWCHARTS

- A flowchart is a graphical or symbolic representation of a process.
- It is basically used to design and document virtually complex processes to help the viewers to visualize the logic of the process.
- When designing a flowchart, each step in the process is depicted by a different symbol and is associated with a short description.

The symbols used in a flowchart include the following

- ✓ **Start and end symbols** are also known as the terminal symbols and are represented as circles, ovals, or rounded ingles. Terminal symbols are always the first and the last symbols in a flowchart.

- ✓ **Arrows** depict the flow of control of the program. They illustrate the exact sequence in which the instructions are executed.

- ✓ **Generic processing step,** also called as an activity, is represented using a rectangle. Activities include instructions such as add a to b, save the result Therefore, a processing symbol represents arithmetic and data movement instructions.

- ✓ **Input/Output symbols** are represented parallelogram and are used to get inputs from the users or display the results to them.

- ✓ **A conditional or decision symbol** is represented using a diamond, It is basically used to depict a Yes/No question or a True/False test.

- ✓ **Labelled connectors** are represented by an identifying label inside a circle and are used in complex or multisheet diagrams to substitute for arrows. For each label, the 'outflow' connector must have one or more 'inflow' connectors. A pair of identically labelled connectors is used to indicate a continued flow when the use of lines using a becomes confusing.

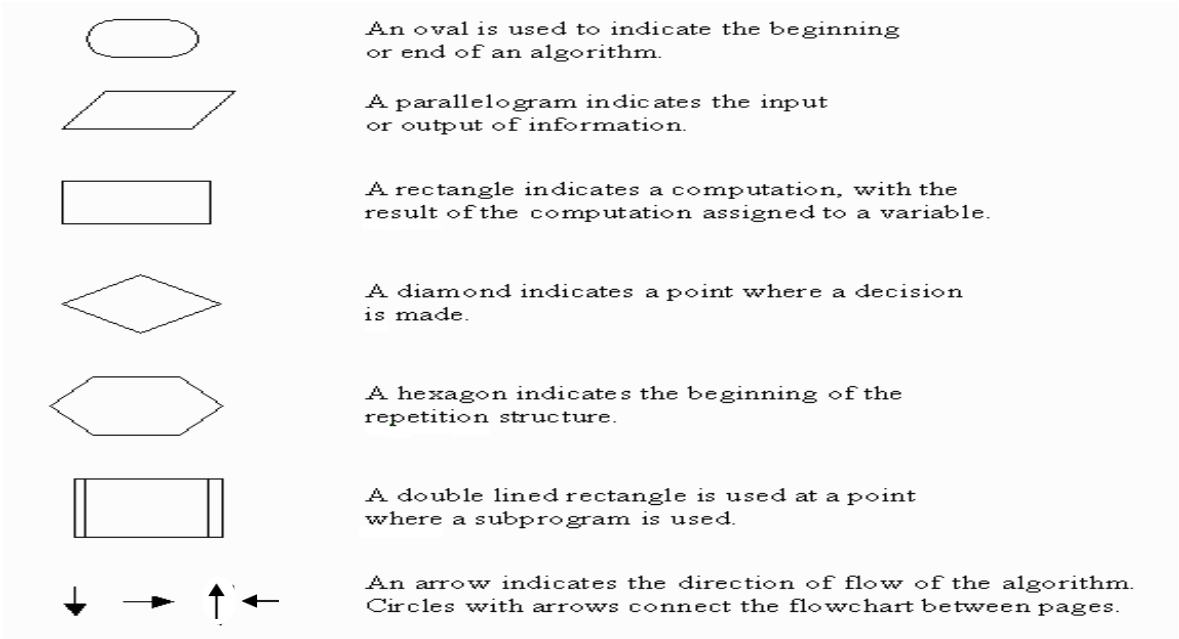
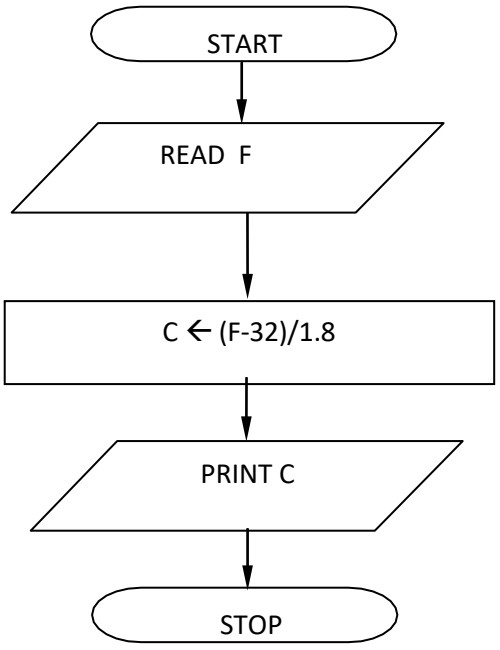


FIGURE : SYMBOLS USED IN A FLOWCHART

Examples on Algorithms and Flowcharts:

Write an algorithm to accept temperature in Fahrenheit scale andconvert into Celsius scale and draw a flowchart.



Formula:- C = (F-32)/1.8

where F is Fahrenheit value and C is Celsius value.

Algorithm : Temperature conversion from Fahrenheit value to Celsius value.

Algorithm: Temperature Conversion

Step 1: [Input temperature in Fahrenheit scale] Input F

Step 2: [Compute centigrade temperature] C ⇓ (F-32) / 1.8

Step 3: [Print the result] Write —Equivalent temperature in Celsius scale is —,C

Step 4: [End of algorithm] End

PSEUDOCODE

- Pseudocode is a compact and informal high-level description of an algorithm that uses the structural conventions of a programming language.
- Pseudocodes are an outline of a program that can easily be converted into programming statements.
- They consist of short English phrases that explain specific tasks within 4 program's algorithm. They should not include keywords in any specific computer language.
- It is basically meant for human reading rather than machine reading, so it omits the details that are not essential for humans. Such details include variable declarations, system-specific code, and subroutines.
- This helps even non-programmers to understand the logic of the designed solution. There are no standards defined for writing a pseudocode, because a pseudocode is not an executable program.
- Flowcharts can be considered as graphical alternatives to pseudocodes, but require more Space on paper.

EXAMPLE

Write a pseudocode for calculating the price of a product after adding the sales tax to its original price.

Solution

1. Read the price of the product
2. Read the sales tax rate
3. Calculate sales tax = price of the item x; sales tax rate
4. Calculate total price = price of the product + sales tax
5. Print total price
6. End

Variables: price of the item, sales tax rate, sales tax, total price

TYPES OF ERRORS

While writing programs, very often we get errors in our programs. These errors if not removed will cither give erroneous output or will not let the compiler to compile the program. These errors are broadly classified under four groups : Run-time, Compile-time, Linker and Logical errors

Types of Errors

1. **Run-time Errors** - As the name suggests, run-time errors occur when the program is being run executed. Such errors occur when the program performs some illegal operations like
   - ✓ dividing a number by zero
   - ✓ opening a file that already exists
   - ✓ lack of free memory space
   - ✓ finding square or logarithm of negative numbers Run-time errors may terminate program execution,

   so the code must be written in such a way that it handles all sorts of unexpected errors rather terminating it unexpectedly.

2. **Compile-time Errors** - Again us the name implies, compile-time errors occur at the time of compilation of the program.Such errors can be further classified as follows:

✓ **Syntax Errors** Syntax errors are generated when rules of a programming language are violated.

✓ **Semantic Errors** Semantic errors are those errors which may comply with rules of the programming language but are not meaningful to the compiler.

3. **Logical Errors** - Logical errors are errors in the program code that result in unexpected and undesirable output which is obviously not correct. Such errors are not detected by the compiler, and programmers must check their code line by line or use a debugger to locate and rectify the errors. Logical errors occur due to incorrect statements,

4. **Linker Errors** These errors occur when the linker is not able to find the function definition for a given prototype

TESTING AND DEBUGGING APPROACHES

Testing

- Testing is an activity that is performed to verify correct behaviour of a program.
- Ideally testing should be conducted at all stages of program development.
- However, in the implementation stage, the following three types of tests can be conducted they are

Unit Tests - Unit testing is applied only on a single unit or module to ensure whether it exhibits the expected behavior.

Integration Tests - These tests are a logical extension of unit tests. In this test, two units that have already been tested are combined into a component and the interface between them is tested. This process is repeated until all the modules are tested together. The main focus of integration testing is to identify errors that occur when the units are combined.

System Tests - system testing checks the entire system. For example, if our program code consists of three modules then each of the module is tested individually using unit tests and then system test is applied to test this entire system as one system

Debugging

Debugging, on the other hand, is an activity that includes execution testing and code correction. The main aim of debugging is locating errors in the program code. Once the errors are located, they are then isolated and fixed to produce an error-free code.

Different approaches applied for debugging a code includes:

Brute-Force Method - In this technique, a printout of CPU registers and relevant memory locations is taken, studied, and documented. It is the least efficient way of debugging a program and is generally done when all the other methods fail.

Backtracking Method - It is a popular technique that is widely used to debug small applications. It works by locating the first symptom of error and then tracing backward across the entire source code until the real cause of error is detected.

However, the main drawback of this approach is that with increase in number of source code lines, the possible backward paths become too large to manage.

Cause Elimination - In this approach, a list of all possible causes of an error is developed. Then relevant tests are carried out to eliminate each of them. If some tests indicate that a particular cause may be responsible for an error then the data are refined to isolate the error.

HISTORY OF C LANGUAGE

The history of C starts with a language called BCPL (Beginners Combined Programming Language) developed by Martin Richards.

- In 1970, Mr. Ken Thompson, a System Engineer at AT & T Bell Laboratories, USA, wrote an earlier version of C language for the UNIX operating system. It is a modified version of the BCPL.
- Therefore to distinguish his language from BCPL, he called the language as B language, the first letter of BCPL.
- The B language was modified to a greater extent by Mr. Dennis Ritchie at Bell Labs. This modified version of B is called as the C language, the second letter of BCPL.
- C was originally designed for UNIX operating system.
- But, later the whole UNIX operating system itself was almost rewritten in C language.
- C is a powerful, general purpose, procedure oriented, structured programming language.

INTRODUCTION TO C

- High Level Language (HLL) has been developed to facilitate easy programming of the computers by any ordinary person.
- One such HLL is the C Language.
- It is becoming so popular because C being a High Level language satisfies the varying requirements of programmers.
- It can be used for application program development as well as system program development.

IMPORTANT FEATURES OF C LANGUAGE

1. C is a system programming language which provides flexibility for writing compilers, operating systems, etc.
2. It can also be used for writing the application programs for scientific, engineering and business applications.
3. C is famous for its portability, meaning that program written in C for one computer can be easily executed by another computer with little or no changes.
4. C supports variety of data types like integers, float point numbers, characters, etc.
5. C is a procedure oriented language which is most suited for structured programming practice.
6. It provides a rich set of built in functions.
7. Programs written in C are found to execute faster compared to other languages.

CHARCTERISTICS OF C:

- C is a high-level programming language, which enables the programmer to concentrate on the problem at hand and not worry about the machine code on which the program would be run.
- Small size—C has only 32 keywords. This makes it relatively casy to leam as compared to other languages.
- C makes extensive use of function calls.
- C is well suited for structured programming.
- Unlike PASCAL it supports loose typing (as a character can be treated as an integer and vice versa).
- Structured language as the code can be organized as a collection of one or more functions.
- Quick language as a well written C program is likely to be as quick as or quicker than a program written in any other language.
- C programs make use of operators and data types, they are fast and efficient. For example, a program written to increment a value from Uses of C.
- Facilitates low level (bitwise) programming.
- Supports pointers to refer computer memory, arrays, structures, and functions.
- C is a core language as many other programming languages (like C++, Java, Perl, etc.) are based on C.

- C is a portable language, i.e., a C program written for one computer can be run on another computer with little or no modification.
- C is an extensible language as it enables the user to add his own functions to the C library.
- C is often treated as the second best language for any given programming task.

USES OF C

C is a very simple language that is widely used by software professionals around the globe. The uses of C language can be summarized as follows:

- C language is primarily used for system programming.
- The portability, efficiency, the ability to access specific hardware addresses, and low runtime demand on system resources make it a good choice for implementing operating systems and embedded system applications.
- C has been so widely accepted by professionals that compilers, libraries, and interpreters of other programming languages are often written in C.
- For portability and convenience reasons, C is sometimes used as an intermediate language for implementation of other languages.
- Basically, C was designed as a programming language and was not meant to be used as a compiler target language.
- Therefore, although C can be used as an intermediate language it is not an ideal option. This led to the development of C-based intermediate languages such as C.
- C is widely used to implement end-user applications.

STRUCTURE OF C

A C program is composed of preprocessor commands, a global declaration section, and one or more functions.

And to write a C program, we first need to write the code. For this, open a text editor. If you are a Windows user you may use Notepad and if you prefer working on UNIX/Linux.

[Comments /Documentation Section] [Pre-processor Directives/Link Section][Global Declaration Section]

[Function Declaration Section]
main( )

{

Declaration Section;
Executable part;

}

Subprogram section
[Function1] [Function2]

.

.

[Function n]

Example:

```
/* Sample C Program */
void main( )
{
    int a, b;
    clrscr( );
    printf("Enter two numbers");
    scanf("%d%d", &a, &b);
    sum = a + b;
    printf("Sum = %d", sum);
}
```

Structure of C Language Includes the Following Sections

1. **The Documentation Section:**
   - It consists of a set of comment lines giving the name of the program, and other details which the programmer would like to use later.
   - The comments are enclosed in a C program using /* and */ .
2. **The Preprocessor directive or Link section:**
   - It provides instruction to the compiler to link some functions or do some processing prior to the execution of the program.
   - It is also used to define symbolic constants of the program.
3. **Global Declaration Section:**
   - There are some variables that are used in more than one function.
   - Such variables are called global variables and are declared in this section that is outside of all other functions.

4. The Function declaration Section:

- Here the functions used in the program are declared.
- They are called function prototypes.

5. main () function section :

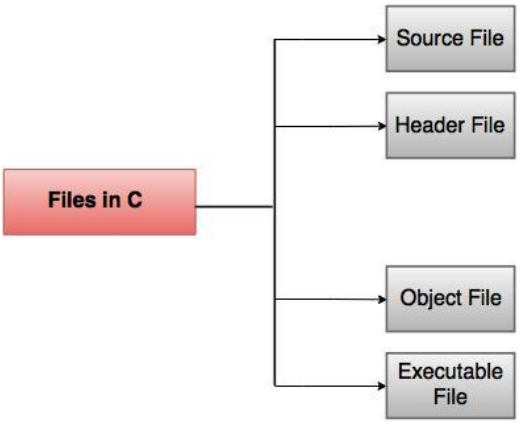
- Every C program must have one main( ) function.
- The execution always begins from main
- This section contains two parts.
- Declaration part and
- Executable part.
- The declaration part declares all the variables used in the executable part.
- There must be at least one statement in the executable part.
- These two parts must appear between the opening and closing braces.
- The program execution begins at the opening brace and ends at the closing brace.
- All statements in the declaration part and executable parts must end with a semicolon.

6. Sub program Section:

- It contains all the user defined functions that are called in the main () function.
- User defined functions are generally placed immediately after the main function, although they may appear in any order.
- All sections except the main () function may be absent when they are not required in any C program

FILES USED IN C PROGRAM

Every C program has four kinds of files associated with it



1. **Source Code Files**
    - The source code file contains the source code of the program. The file extension of any C source code file is '.c'.
    - This file contains C source code that defines the main function and maybe other functions.
    - The main() function is the starting point of execution when you successfully compile and run the program.
    - A C program in general may include even other source code files (with the file extension.c)
2. **Header Files**
    - They have an extension '.h'. They contain the C function declarations and macro definitions that are shared between various source files.

      Advantages of header files:
    - At times the programmer may want to use the same subroutines for different programs.
    - To do this, he would just compile the code of the subroutine once and link to the resulting object file in any file in which the functionalities of this subroutine are required.
    - At times the programmer may want to change or add the subroutines and reflect those changes in all the programs. For doing this, he will have to only change the source file for the subroutines, recompile the source code and then recompile and re-link the program.
    - This tells us that including a header file will make it easier at all levels of the program. If we need to modify anything then changes are made only in the subroutines after which all the changes will be reflected.

Standard header files

C provides us with some standard header files which are available easily.

Common standard header files are:

i) string.h – used for handling string functions.

ii) stdlib.h – used for some miscellaneous functions.

iii) stdio.h – used for giving standardized input and output.

iv) math.h – used for mathematical functions.

v) alloc.h – used for dynamic memory allocation.

vi) conio.h – used for clearing the screen.

The header files are added at the start of the source code so that they can be used by more than one function of the same file.

3. **OBJECT FILES**
   - Object files are generated by the compiler as a result of processing the source code file.
   - Object files contain compact binary code of the function definitions.
   - Linker uses these object files to produce an executable file (.exe file) by combining the object files together.
   - Object files have a '.o' extension, although some operating systems including Windows and MS-DOS have a '.obj' extension for the object file.
4. **EXECUTABLE FILES**
   - This file is generated by the linker.
   - Various object files are linked by the linker for producing a binary file which will be executed directly.
   - They have an '.exe' extension.

COMPILING AND EXECUTING C PROGRAMS

COMPILERS

A compiler is a software that converts the source code to the object code. In other words, we can say that it converts the high-level language to machine/binary language. Moreover, it is necessary to perform this step to make the program executable. This is because the computer understands only binary language.

This process of converting the source code into machine code is called compilation

- C is a compiled language. So once a C program is written, you must run it through a C compiler that can create an executable file to be run by the computer. While the C program is human-readable, the executable file, on the other hand, is a machine-readable file available in an executable form.
- The mechanical part of running a C program begins with one or more program source files, and ends with an executable file, which can be run on a computer.
- The programming process starts with creating 4 source file that consists of the statements of the program written in C language. This source file usually contains ASCII characters and can be produced with a text editor, such as Windows notepad, or in an Integrated Design Environment

- **The source file is then processed by a special program called a compiler, The compiler translates the source code into an object code.** The object code contains the machine instructions for the CPU, and calls to the operating system API (Application Programming Interface).However, even the object file is not an executable file.
- Therefore, in the next step, the object file is processed with another special program called a linker. While there is a different compiler for every individual language, the same linker is used for object files regardless of the original language in which the new program was written, The output of the linker is an executable or runnable file.
- In C language programs, there are two kinds of source files. In addition to the main (.c) source file, which contains executable statements there are also header (.n) source files. These header files should be written as part of the source code for modular C programs.
- The compilation process is done in two steps, In the first step, the pre-processor program reads the source file as text, and produces another text file as output. Source code lines which begin with the # symbol are actually not written in C but in the pre-processor language. The output of the pre-processor is a text file which does not contain any pre-processor statements. This file is ready to be processed by the compiler. The linker combines the object file with library routines (supplied with the compiler) to produce the final executable file.
- In modular programming, the source code is divided into two or more source files. All these source files are compiled separately thereby producing multiple object files.

USING COMMENTS IN C

- ✓ Many a time the meaning or the purpose of the program code is not clear to the reader.
- ✓ It is a good programming practice to place some comments in the code to help the reader understand the code clearly.
- ✓ Comments are just a way of explaining what a program does. It is merely an internal program documentation.
- ✓ The compiler ignores the comments when forming the object file. This means that the comments are non-executable statements.
- ✓ C supports two types of comments: * //is used to comment a single statement. This is known as a line comment. A line comment can be placed on the line and it does not

require to be specially ended as the end of the line automatically ends the line**.** statements.

- ✓ A /* is ended with */ the statements lies with these characters are commented. This type of comment is known as block comment,
- ✓ Note that commented statements are not executed by the compiler**.**

C TOKENS

- Tokens are the smallest or basic units of C program.
- One or more characters are grouped in sequence to form meaningful words.these meaningful words are called as tokens.
- A token is collection of characters.
- Tokens are classified in to 5 types as below

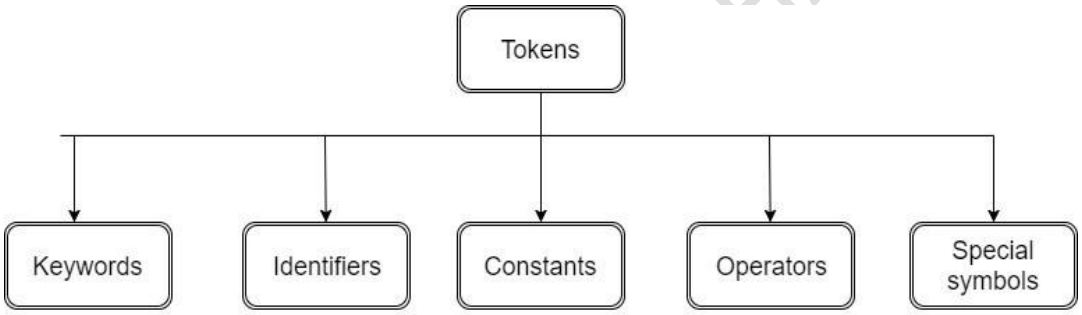


FIGURE : CLASSIFICATION OF C TOKENS

1. Keywords or Reserve words such as float, int, etc.

2. Constants such 1, 15, 5 etc.

3. Identifiers such as name, amount etc.

4. Operators such as +, -, * etc.

5. Special symbols such as comma (,), Semicolon (;) {, }, ? etc and

6. Strings such as “SJMIT”.

Keywords

- ✓ Key words also known as Reserve words of the C language are the words whose
- ✓ meaning is already defined and explained to the C language compiler.
- ✓ Therefore Reserve words cannot be used as identifiers or variable names.
- ✓ They should only be used to carry the pre-defined meaning.
- ✓ For example int is a reserve word.

- ✓ It indicates the data type of the variable as integer.
- ✓ Therefore it is reserved to carry out the specific meaning.
- ✓ Any attempt to use it other than the intended purpose will generate a
- ✓ Compile time error.
- ✓ C language has 32 keywords.
- ✓ Following are some of them – auto, break, case, char, const, continue, default, do, double, else, enum, extern, float, for, goto, if, int, long, return, short , etc

Constants

- ✓ A constant can be defined as a value or a quantity which does not change during
- ✓ the execution of a program.
- ✓ Meaning and value of the constant remains unchanged throughout the execution of
- ✓ the program.
- ✓ These are also called as literals.

C supports following types of constants.

1. Integer Constants

- ➢ An integer constant refers to a sequence of digits.
- ➢ There three types of integer constants. They are,
- Decimal integers,
- Octal integers and
- Hexadecimal integers.
- ➢ Decimal integer constant consists of set of digits from 0 to 9 preceded by an optional + or – sign.
  Ex: 123, -321, 0, 4567, + 78
  Embedded spaces, commas and non-digit characters are not permitted between digits.
- ➢ An octal integer constant consists of any combination of digits from 0 to 7 with a leading 0(zero).
  Ex : 037, 0435, 0567
- ➢ An hexadecimal constant contains sequence of digits preceded by 0x or 0X.
  They may also include alphabets A to F or a to f representing numbers from 10 to 15.
  Ex: 0x9f, 0X4A, 0x521 etc
  The largest integer value that can be stored is machine dependent;
  It is 32767 for 16 bit computers.
  It is also possible to store larger integer constants by appending qualifiers such U, L and UL to the constants. UL to the constants.

2. Floating point Constants or Real Constants

- ➢ The quantities that are represented by numbers with fractional part are called floating point numbers.
  Ex: 0.567, -0.76, 56.78, +247.60

- These numbers are shown in decimal notation, having a whole number followed by a decimal point and the fractional part.
- It is possible to omit digits before the decimal point or digits after the decimal point.
  Ex: 215. , .95, -.76 or +.5
- A real number or a floating point number can also be expressed in exponential notation.
  Ex: 2.15E2
- The general form of exponential notation is mantissa e exponent OR mantissa E exponent
- The mantissa is either a real number or an integer. The exponent is an integer with an optional plus or minus sign.

3. Single Character constants

- A single character constant contains any valid character enclosed within a pair of single quote marks.
- Ex: „5„ „A„ „;„ , „?“
- The character constants have integer values associated with them known as ASCII values.
- For ex: A is having the ASCII value of 65.

4. String constants

A string constant is a sequence of characters enclosed in double quotes. The characters may be alphabets, numbers, special characters and blank space.
Ex: “Hello”, “2002”, “Wel Come”, “ —5+3”

5. Backslash character constants or Escape sequence characters.

- C supports some special backslash character constants that are used in printf() functions.
  For ex: „\n„ stands for new line character.
- Each one of them represents a single character even though it consists of two characters.
- The following are backslash constants.
  \a Audible alert or bell \b back space
  \f form feed \n new line
  \r carriage return \t horizontal tab
  \v vertical tab \‘ single quote
  \” double quote \? Question mark
  \\ backslash \0 null

IDENTIFIERS

- Identifiers are the names given to variables, arrays, functions etc.
- They are the names given to C language constructs.

Rules for defining identifiers:

1. An Identifier consists of sequence of letters, digits, or the underscore

2. The first character must be either a letter or underscore

3. Special characters [Except under score] are not allowed.

4. Keywords or Reserve words are not allowed.

5. Uppercase and lowercase are different.

For ex: Sum, SUM and sum are three distinct variables.

6. Up to 31 characters are significant.

Some Valid examples for identifiers are:

Num num val1 large average x123 value_of _area

Some invalid examples for identifiers are:

1num …….. (Should not begin with number)

1234 ……(Should begin with letter or underscore)

(avg) …….(Parentheses not allowed)

int ……..(Reserve words not allowed)

Average# …… ..(Special symbols not allowed)

Value-of …….. (Hyphen not allowed)

Roll no ……. (Blank space not allowed)

DATA TYPES

- ✓ Data Type: It is the type of data value that is stored in particular memory location.
- ✓ Data types are used to classify the values as integer, real, character, Boolean etc.
- ✓ In C language, data types are used for declaring variables of different types.

1. Primitive Data Types OR Basic Data Types OR Primary Data Types

- Primary data types are the most fundamental and basic building blocks of C programming.
- Derived data types are constructed using them.

- **C language has the four basic data types**

1. **int (Ex: 1,20 and 352),**
2. **char (Ex: „c‟, „A‟and „X‟),**
3. **float (Ex: 25.256789)**
4. **double (Ex: 12.789123450000**).

1. int Data Type

 Integer refers to a whole number with a range of values supported by a particular machine.

 Generally integers occupy one word of storage i.e 16 bits or 2 bytes.

 So its value can range from -32768 to +32767.

Ex: 25, 4235, 180

2. float Data Type

 A floating point number consists of sequence of one or more digits of decimal number system along with embedded decimal point and fractional part if any.

 Computer allocates 32 bits, i.e. 4 bytes of memory for storing float type of variables.

 These numbers are stored with 6 digits of precision for fractional part.

EX: 25.234000, 423.500000, 180.000000

3. double Data Type

 It is similar to the float type.

 It is used whenever the accuracy required to represent the number is more.

 In others words variables declared of type double can store floating point numbers with number of significant digits is roughly twice or double than that of float type.

 It uses 64 bits i.e. 8 bytes of memory giving a precision of 14 decimal digits.

EX: 25.23400000000000, 423.50000000000000, 180.00000000000000

4. char Data Type

 A single character can be defined as char data type.

 These are stored usually as 8 bits i.e 1 byte of memory.

String refers to a series of characters. Strings are declared as array of char types.

Ex: char name[20]; will reserve a memory location to store up to 20

characters.

- ✓ Further, applying qualifiers to the above primary data types yield additional data types.
- ✓ A qualifier alters the characteristics of the data type, such as its sign or size.
- ✓ There are two types of qualifiers namely, sign qualifiers and size qualifiers.
- ✓ signed and unsigned are the sign qualifiers short and long are the size qualifiers.

2. Derived data types:

- ✓ These are the composite data types which are derived from multiple primitive data types.
- ✓ Arrays, pointers, structures and unions are called derived data types in C language.

3. Void data type :

- ✓ Void is an empty data type that has no value. This can be used in functions and pointers.

4. User defined data types :

- ✓ C supports the features "typedef" that allows users to rename an existing data type.
- ✓ This user defined data type can then be used to declare variables:
- ✓ Syntax: typedef data_type new_name;
- ✓ Example: typedef int numbers;
  numbers num1, num2;

5. Enumerated Data Type

- The enumerated data type is defined as follows.
- ✓ Syntax: enum identifier {value1, value2,…., value n};
- The identifier is enumerated data type which can be used declare variables that can have one of the values enclosed within curly braces. [They are also known as enumeration constants].
- Now variables can be declared as enum identifier v1, v2, …., vn;
- The enumerated variables v1, v2,….., vn can have one of the values value1, value2,…., valuen

.

Example:

- enum day {Monday, Tuesday, ……., Sunday};
- enum day week_st, week_end;
- week_st = Monday;
- week_end = Friday;
- if (week_st == Tuesday)
- week_end = Saturday;

DECLARATION OF STORAGE CLASS

- ✓ Variables in C can have storage class in addition to data type.
- ✓ The storage class provides information about location and visibility.
- ✓ Consider the example:
  Here, variable m is called global variable because it is declared before main( ).
  It can be used in all the functions defined in the program.
  It need not be declared in other functions.
  The global variable is also known as external variable.
  The variables i, balance, and sum are called local variables because they are declared inside the functions.
- ✓ Local variables are visible and meaningful only inside the functions in which they are declared.
- ✓ Note that the variable i is declared in both the functions.
- ✓ Any change in the value of i in one function does not affect its value in
- ✓ another function.
- ✓ Example:
- ✓ auto int count;
- ✓ register char ch;
- ✓ static int x;
- ✓ extern long total;
- static and extern variables are automatically initialized to zero.
- auto variables contain undefined values (garbage values) if they are not initialized explicitly.

VARIABLES

- A variable is a memory location that may be used to store a data value.
- The variable keeps changing its value during execution of a program.
- Programmer has to choose the variable name in a meaningful way.
  Ex: num1, num2, average, large

Declaration of variables:

- Every variable must be declared before it is used in the program.
- When a variable is declared, then the memory is reserved for the variable.

Syntax:

- Data_type variable_name1, variable_name2, ……variable nameN ;
  where data type is any basic data type int or float or char or double.

Example: int n, height, count, digit;

float rate, average , y_coordinate, p1;

- ✓ The following declarations of variables are invalid: float, a , b ,c ; (comma after float is not valid)
  int : x; (: is not valid)
  real x, y; (real is not the correct type_name)

Assigning Values to Variables

- ✓ Values can be assigned to variables using assignment operator =.

Syntax:

Variable name = constant / variable / expression;

Ex: a = 10; initial_value = 0;

- ✓ It is also possible to assign a value to a variable at the time of declaration itself.
  Ex: int a=10, b; float x=2.45, y=1.2e-9;
- ✓ The process of giving initial value to the variable is called initialization of the variable.

CONSTANTS

- ✓ Constant is the quantity that does not change its value during execution of a program.
- ✓ Defining Symbolic Constants
  We often use certain unique constants in a program.
  These constants may appear repeatedly in number of places in a program.
  Such constants can be defined and its value can be substituted during the pre-processing stage itself.

Syntax:

#define symbolic-name value of constant

ex:

#define PI 3.14

#define MAX 100

INPUT AND OUTPUT STATEMENTS IN C

- ✓ The C language consists of input statements to read the data to be processed as well as output statements to display the computed results.
- ✓ C language provides a set of library functions or built-in functions, to carry out input and output operations.
- ✓ These library functions are available in a header file called stdio.h.
- ✓ So, for using these library functions, the following preprocessor directive is essential.

include <stdio.h>

The input and output functions supported by C language can be categorized into the following:

- Unformatted Input/Output Functions:
- Formatted Input/Output Functions:

scanf( ) and printf( )

Unformatted Input/Output Functions:

The unformatted input functions supported in C language are as follows :

1. getchar( )

 It is used to read one character from the keyboard.

Syntax:

getchar( );

Example:

char ch;

ch = getchar(); // ch is a character variable.

2. getch()

- It is also used to read one character from the keyboard.
- But, the typed character is not displayed on the screen.

Syntax:

getch( );

Example:

char ch;

ch = getch( ); // ch is a character variable.

3. getche( )

- □ It is also used to read one character from the keyboard.
- □ But, the typed character is displayed on the screen.

Syntax:

getche( );

Example:

char ch;

ch = getche(); // ch is a character variable.

4. gets( )

□ It is used to read a string of characters from the keyboard till the user presses "EnterKey".

Syntax:

gets(str); where str is a string variable.

Example:

char str1[6];

gets(str1);

□ The unformatted output functions supported in C language are as follows:

1. putch( )

This function displays one character on the screen.

Syntax:

putch(ch);

Principles of Programming Using C (22POP13)

Prof. Poral Nagaraj, SJM Institute of Technology, Chitradurga – 577502, Email: nagarajaporaal@sjmit.ac.in

Example:

char ch;

putch(ch); // ch is a character variable.

2. putchar( )

This function also displays the character on the screen.

Syntax:

putchar(ch);

Example:

char ch;

putchar(ch); // ch is a character variable.

3. puts( )

This function displays a string of characters on the screen.

Syntax:

puts(str); where str is a string variable.

Example:

char str1[6];

puts(str1);

Formatted Input/Output Functions:

- scanf( ) and printf( ) are the formatted input and formatted output functions respectively.
- They allow the use of format specifiers (also called format characters or conversion characters) to specify the type of data to be read or printed.

1. scanf( ) function

- It is a formatted input function.
- It is used to read data into variables from the standard input device, i.e., a keyboard.

Syntax:

scanf("format-string", &var1,&var2,……&varn);

where format-string contains only format characters and they give information to the computer about the type of data to be stored in the list of

variables var1,var2……varn.

- The number of format characters must be equal to the number of variable and the type of format characters must be same as the type of variables.
- The list of format characters is shown below table.

Format specifiers	Description
%c	To input/print the character type
%d	To input/print the integer type
%f	To input/print the floating point type
%e or %E	To input/display the floating point with exponent
%g	To input/display the floating point data with or without exponent (Trailing zeroes will not be displayed)
%s	To input/display a string
%lf	To input/display long float or double
%ld	To input/display long integer
%o	To input/display octal integer
%x or %X	To input/display hexadecimal integer
%i	To input/display decimal or octal or hexadecimal integer
%h	To input/display a short integer
%u	To input/display unsigned integer

Example for scanf() function

int p, q;

float num1, num2;

scanf(" %d%d%f%f ", &p, &q, &num1, &num2);

where & is the address operator used in only scanf( ) function.

2. printf( ) function

- It is a formatted output function.
- It is used to display the values of the variables on the standard output device, i.e., a monitor.

Syntax:

printf("format-string", var1,var2,……varn);

where unlike scanf(), the format-string here contains

a. The string to be displayed as it is

b. Format characters and

c. The backslash constants.

- Here also, the number of format characters must be equal to the number of variables and the type of format characters must be same as the type of variables.

Example for printf() function

int num1, num2, fact;

float average;

printf ("%d %d ", num1, num2);

printf(" The Average = %f ", average);

printf (" The factorial of a given number n is = %d \n", fact);

where

- %d, %f are format characters as in the table above.
- "The factorial of a given number n is =" is the text to be displayed as it is.
- \n is the back slash constant also called new line character.

Example program to illustrate scanf( ) and printf( ) statements:

/* Program to show the use of scanf( ) and printf( ) statements */

```
#include<stdio.h>
void main( )
{
 int a,b,c,d;
 float x,y,z,p;
 clrscr( );
 printf("Enter four values);
 scanf("%d %o %x %u", &a, &b ,&c ,&d);
 printf("The first four data are displayed as follows:\n");
 printf("%d %o %x %u \n"
, a, b, c, d);
 scanf("%f %e %e %f", &x, &y, &z, &p);
 printf("The rest of the read data are displayed as follows:\n");
 printf("%f %e %e %f\n", x, y, z, p);
 printf("End of display");
}
```

Suppose the Input for the above program is,
-768 0362 abf6 3856 -26.68 2.8e-3 1.256e22 6.856
The Output of the program would be:
The first four data are displayed as follows:
-768 362 abf6 3856
The rest of the read data are displayed as follows:
-26.680000 2.800000e-03 1.256000e22 6.856000
End of display