

Shakuntala Krishna Institute Of Technology(KD-64)

Bahbalpur Hawaspur kanpur Dehat -209310

COMPUTER FUNDAMENTALS

ER. Shivansh kumar

Assistant Lecturer

Department Of Computer Science



A handwritten signature in red ink, appearing to be 'SKP' with a long horizontal stroke extending to the right.

~ Unit 1st ~

Time period– 1 weak

- Intoduction.
- Characteristics of computer.
- Uses of computer.
- History of computer.
- Generation of computer.
- Block–Diagram of computer.
- Assignment.

:- What is computer ?

. A computer is an electronic device that, Perform, Process, Calculate, and operations based on instructions provided by a user.

It has the ability to accept data(input), Process it, and then produce outputs.

:- Characteristics Of computer ?

- a) Speed.
- b) Storage capacity.
- c) Accuracy.
- d) Reliability.
- e) Versatility.

Speed :- A computer is a very fast machine .it can perform tasks in few second.

Storage capacity :-The capability of storage makes computer very usefull. It saves the effort of input every time.

Accuracy :- Computer are highly accurate machine and they never make mistakes.

Reliability :- The reliability comes from the facts that we can replace many tasks performed by the user with computer.

Versatility :- Computer are useful everywhere in different respects such as in business, for personal use at home, in school for education.

:- Uses Of Computer ?

. People use computer in many ways,

- a. Business.
- b. Homes.
- c. Automobile
- d. Entertainment.

Business:- Computer are used to track inventories with bar codes and scanners, check the credit status of customers, and transfer funds electronically,

Homes:- Tiny computers embedded in the electronic circuitry of most appliances control the indoor temperature, operate home security system,

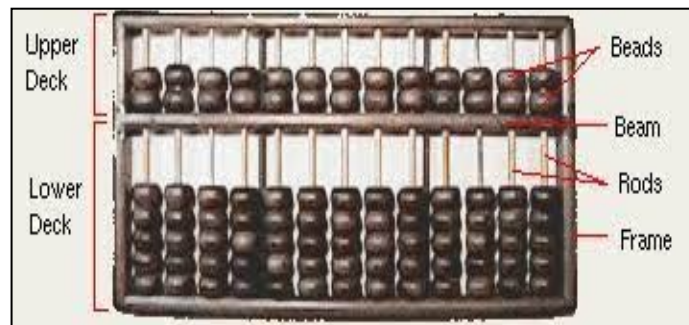
Automobiles:- Regulate the flow off fuel, thereby increasing gas mileage,

Entertainment:- Creating digital sound on stereo systems or computer–animated features from a digitally encoded laser disc.

:-History of Computer ?

Since the creation of man a significant amount of human activities has been ascribed to organizing and processing information so that it could be more easily presented for easy comprehension.

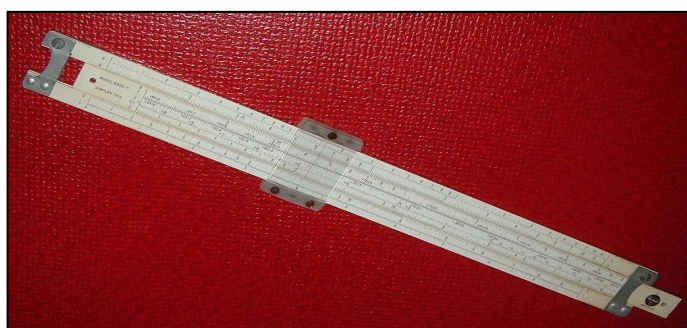
1. **Abacus:**– This is a hand –held device made of beads stung on rods in a frame .



2. **Napier's Bone** :– This was invented by “John Napier(1550–1617)”. It is a mechanical aid to computation that consist of nine such rods with one for each digit 1 through 9. He also invented logarithms which made possible to do division and multiplication by performing addition and subtraction.



3. **Slide Rule:**– By William Oughtred(1575–660). He invented in 1622 but announced it in 1632 this consist of rules on which marking represent logarithms of numbers and also permit calculations.



4. Pascal mechanical calculator(1600) or numerical wheel calculator:

calculator:– Blaise Pascal (1623–1664). In 1642 invented the first adding machine called pascaline.



5. Colmar's Calculator (1820) By Charles Xavier Thomas de Colmar:

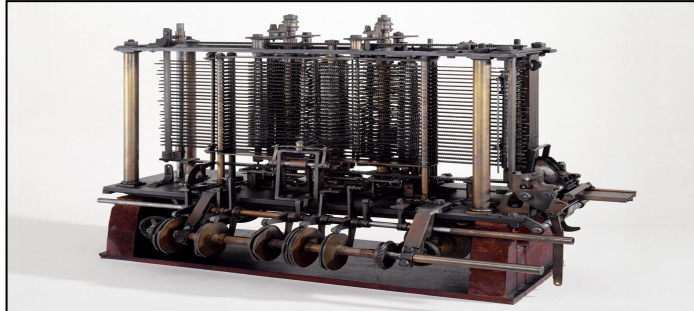
– This presented a more practical approach to computing .



6. Difference Engine (1820):– A difference engine is an automatic mechanical calculator designed to tabulate polynominal function.



7. **Analytical Engine :- (1837).** The analytical engine was a proposed mechanical general purpose computer. It was describe in 1837 as the successor to babbages difference engine.



8. **Mark 1 (1937–1944). “Howard Aiken” :-** First fully automatic calculating machine.



9. **Eniac(electronic numerical integrator and computer) :- “J. presper eckert and john Mauchly” (1943–46).** Was the first electronic, programmable, general purpose digital computer.



10. **Edvac 1946–52.**(electronic discrete variable automatic computer):– was one of the earliest electronic computer.



11. **Edsac**(electronic delay storage automatic calculator):–. Was the earliest British computer and the second electronic digital stored–program computer.



12. **Univac**(universal automatic computer)–(1951):–The Binac built by the eckert–mauchly computer corporation ,was the forst general purpose computer for commercial use, but it was not success. The last UNIVAC computer was produced in 1986.



:– Generation of computer ?

First generation–1940–1956:–(Vacuum tubes)

The first computer used vacuum tube for circuitry and magnetic drum for memory and where often enormous taking up entire rooms. They were very expensive to operate and in addition to using a great deal of electricity generated a lot of heat. First generation computer relied on **machine language** to perform operations and they could only solve one problem at a time. The **UNIVAC** and **ENIAC** was the example of first generation computers.

Second generation–1956–1963:–(Transistors)

Transistors replaced vacuum tube and ushered in the second generation of computers. Second generation computers moved from cryptic binary machine language to symbolic or assembly language which allowed programmers to specify instruction in words. High level languages were also being developed at this time such as early version of **COBOL** and **FORTRAN**. These were also the first computers that stored their instructions in their memory.

Third generation 1964–1971 (Integrated Circuits)

The development of the integrated circuit was the hallmark of the third generation of computers. Transistors were miniaturized and placed on silicon chips called semiconductors which increased the speed and efficiency of computers. Users interacted with third generation of computers through keyboards and monitors and interfaced with an operating system which allowed the device to

run many different application at one time with a central program that monitored the memory..

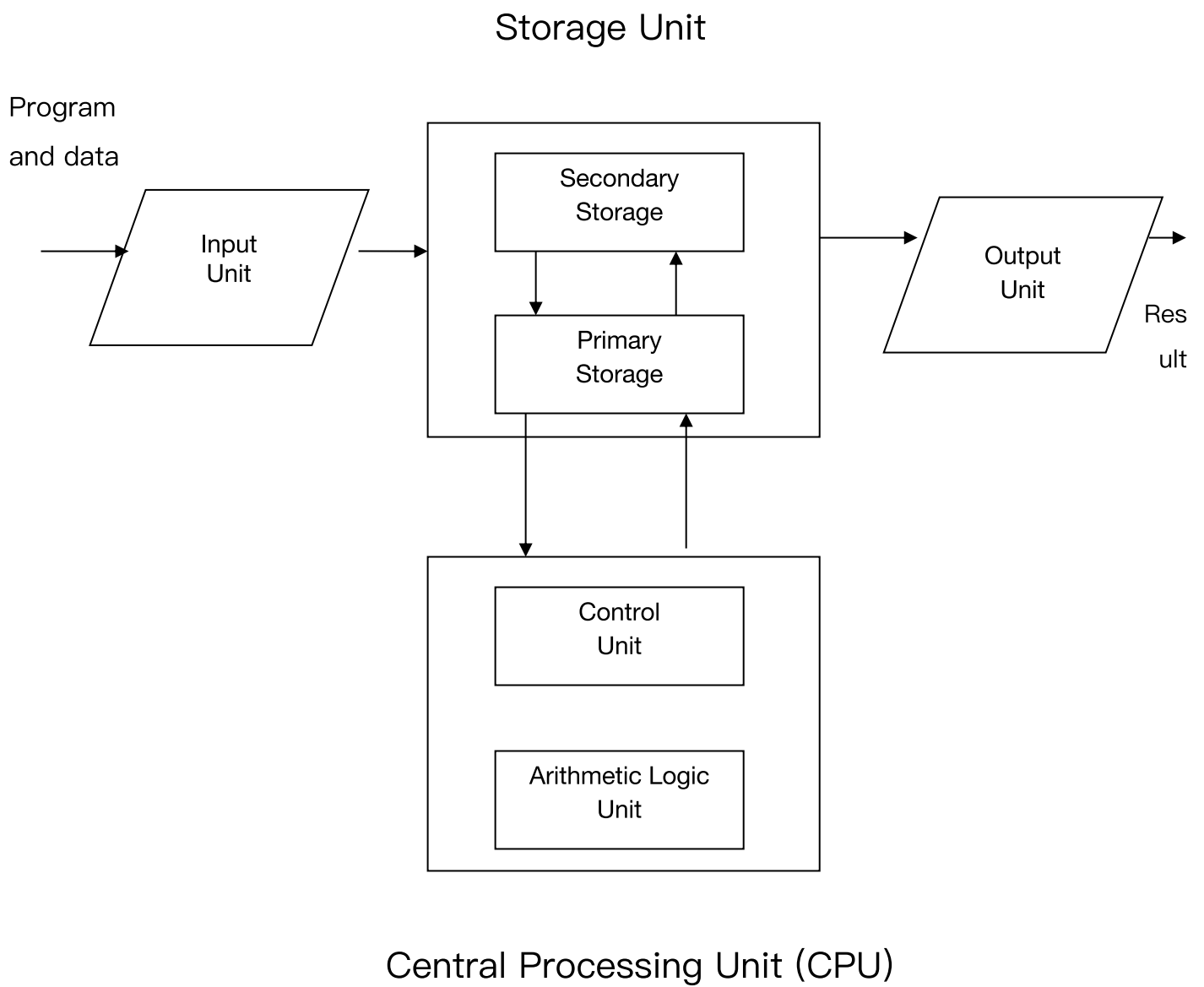
Fourth generation 1971 (microprocessors)

The microprocessors brought the fourth generation computers, as thousands of integrated circuits were built into a single silicon chip. What in the first generation filled an entire room could now fit in the palm of the hand. In 1981 **IBM** introduced its first computer for the home user and in 1984 Apple introduced the Macintosh. As these small computers became more powerful they could be linked together to form networks. Which eventually led to the development of the internet?

Fifth generation —present and beyond_(Artificial intelligence)

Fifth generation computing devices based on artificial intelligence, are still in development, though there are some application, such as voice recognition that are being used today. The goal of fifth generation computing is to develop devices that respond to natural language input and are capable of learning and self-organization.

:–BLOCK–DIAGRAM of computer?



~Assignment~

- What do you mean by computer generations ?
- How many computer generation are there ?
- What do you understand by Block–diagram ?
- Write 3 uses of computer ?
- Write about history of computers ?

~ Unit 2nd ~

Time period– 1 weak

- What Is Software.
- How Does Software Works.
- History Of Software.
- Difference Between Application & System Software.
- What is computer Hardware.
- Internal Hardware Components.
- Input Hardware Components.
- Outputs Hardware Components.
- Assignment.

What is software ?

Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer. Early software was written for specific computers and sold with the hardware it ran on. In the 1980s, software began to be sold on floppy disks, and later on CDs and DVDs. Today most software is purchased and directly downloaded over the internet. Software can be found on vendor websites or application service provider websites.

The two main categories of software are.

- Application software
- System software

Application Software:– Application software is a computer software package that performs a specific function for a user, or in some cases, for another application an application can be self-contained, or it can be a group of programs that run the **modern applications** include office suites, graphics software, databases and database management programs, web browsers, software development tools, image editors etc.

System Software:– These software programs are designed to run a computer's application programs and hardware. System

software coordinates the activities and functions of the hardware and software. The OS is the best example of system software it manages all the other computer programs.

Driver software:– Also known as device drivers, this software is often considered a type of system software. Device drivers controls the device that is connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function.

Middleware:– The term *middleware* describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word.

Programming software:– Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

How does software work ?

Application software consists of many programs that perform specific functions for end users, such as writing reports and

navigating websites. Applications can also perform tasks for other applications.

These desktop applications are installed on a user's computer and use the computer memory to carry out tasks. They take up space on the computer's hard drive and do not need an internet connection to work. However, desktop applications must adhere to the requirements of the hardware devices they run on.

Web applications, on the other hand, only require internet access to work; they do not rely on the hardware and system software to run. Consequently, users can launch web applications from devices that have a web browser. Since the components responsible for the application functionality are on the server, users can launch the app from Windows, Mac, Linux or any other OS.

System software sits between the computer hardware and the application software. Users do not interact directly with system software as it runs in the background, handling the basic functions of the computer. This software coordinates a system's hardware and software so users can run high-level application software to perform specific actions. System software executes when a computer system boots up and continues running as long as the system is on.

History of software ?

The term software was not used until the late 1950s. During this time, although different types of programming software were being created, they were typically not commercially available. Consequently, users -- mostly scientists and large enterprises -- often had to write their own software.

The following is a brief timeline of the history of software:

June 21, 1948. Tom Kilburn, a computer scientist, writes the world's first piece of software for the Manchester Baby computer at the University of Manchester in England.

Early 1950s. General Motors creates the first OS, for the IBM 701 Electronic Data Processing Machine. It is called General Motors Operating System, or GM OS.

- **1958.** Statistician John Tukey coins the word *software* in an article about computer programming.
- **Late 1960s.** Floppy disks are introduced and are used in the 1980s and 1990s to distribute software.
- **Nov. 3, 1971.** AT&T releases the first edition of the Unix OS.
- **1977.** Apple releases the Apple II and consumer software takes off.
- **1979.** VisiCorp releases VisiCalc for the Apple II, the first spreadsheet software for personal computers.
- **1981.** Microsoft releases MS-DOS, the OS on which many of the early IBM computers ran. IBM begins selling software, and commercial software becomes available to the average consumer.

- **1980s.** Hard drives become standard on PCs, and manufacturers start bundling software in computers.
- **1983.** The free software movement is launched with Richard Stallman's GNU (GNU is not Unix) Linux project to create a Unix-like OS with source code that can be freely copied, modified and distributed.
- **1984.** Mac OS is released to run Apple's Macintosh line.
- **Mid-1980s.** Key software applications, including AutoDesk AutoCAD, Microsoft Word and Microsoft Excel, are released.
- **1985.** Microsoft Windows 1.0 is released.
- **1989.** CD-ROMs become standard and hold much more data than floppy disks. Large software programs can be distributed quickly, easily and relatively inexpensively.
- **1991.** The Linux kernel, the basis for the open source Linux OS, is released.
- **1997.** DVDs are introduced and able to hold more data than CDs, making it possible to put bundles of programs, such as the Microsoft Office Suite, onto one disk.
- **1999.** Salesforce.com uses cloud computing to pioneer software delivery over the internet.
- **2000.** The term software as a service (SaaS) comes into vogue.
- **2007.** iPhone is launched and mobile applications begin to take hold.

- **2010 to the present.** DVDs are becoming obsolete as users buy and download software from the internet and the cloud. Vendors move to subscription-based models and SaaS has become common.

Difference between Application & System Software ?

System software vs. application software

System software	Application software
General-purpose software that manages basic system resources and processes	Software that performs specific tasks to meet user needs
Written in low-level assembly language or machine code	Written in higher-level languages, such as Python and JavaScript
Must meet specific hardware needs; interacts closely with hardware	Does not take hardware into account and doesn't interact directly with hardware
Installed at the same time as the OS, usually by the manufacturer	User or admin installs software when needed
Runs any time the computer is on	User triggers and stops the program
Works in the background and users don't usually access it	Runs in the foreground and users work directly with the software to perform specific tasks
Runs independently	Needs system software to run
Is necessary for the system to function	Isn't needed for the system to function

What is computer hardware ?

Hardware is the physical part of a computer, which we can touch and feel. Monitor, Keyboard, Mouse and CPU etc are the examples of hardware.

There are three types of hardware such as input device, output device and storage device.

The devices which are used to input data in a computer are called input devices. It consists keyboard, mouse, microphone, scanner etc.

The devices which are used to get the data stored in a computer is called output devices. It consists printer, monitor, speaker etc.

Storage devices are used to store data. There are two type of storage devices–

1. Primary Memory– RAM, ROM
2. Secondary Memory– CD, DVD, Pen drive and Floppy disk etc.

What are internal computer hardware components?

- **Motherboard**. This is a printed circuit board that holds the central processing unit (CPU) and other essential internal hardware and functions as the central hub that all other hardware components run through.
- **CPU**. The CPU is the brain of the computer that processes and executes digital instructions from various programs; its clock speed determines the computer's performance and efficiency in processing data.

- **RAM**. RAM -- or dynamic RAM -- is temporary memory storage that makes information immediately accessible to programs; RAM is volatile memory, so stored data is cleared when the computer powers off.
- **Hard drive**. Hard disk drives are physical storage devices that store both permanent and temporary data in different formats, including programs, OSes, device files, photos, etc.
- **Solid-state drive (SSD)**. SSDs are solid-state storage devices based on NAND flash memory technology; SSDs are non-volatile, so they can safely store data even when the computer is powered down.
- **Optical drive**. Optical drives typically reside in an on-device drive bay; they enable the computer to read and interact with nonmagnetic external media, such as compact disc read-only memory or digital video discs.
- **Heat sink**. This is a passive piece of hardware that draws heat away from components to regulate/reduce their temperature to help ensure they continue to function properly. Typically, a heat sink is installed directly atop the CPU, which produces the most heat among internal components.
- **Graphics processing unit**. This chip-based device processes graphical data and often functions as an extension to the main CPU.

- **Network interface card (NIC)**. A NIC is a circuit board or chip that enables the computer to connect to a network; also known as a *network adapter* or local area network adapter, it typically supports connection to an Ethernet network.

Other computing components, such as USB ports, power supplies, transistors and chips, are also types of internal hardware.

Common input hardware components :–

- **Mouse**. A mouse is a hand–held pointing device that moves a cursor around a computer screen and enables interaction with objects on the screen. It may be wired or wireless.
- **Keyboard**. A keyboard is an input device featuring a standard QWERTY keyset that enables users to input text, numbers or special characters.
- **Microphone**. A microphone is a device that translates sound waves into electrical signals and supports computer–based audio communications.
- **Camera**. A camera captures visual images and streams them to the computer or through a computer to a network device.
- **Touchpad**. A touchpad is an input device, external or built into a laptop, used to control the pointer on a display screen. It is typically an alternative to an external mouse.

- **USB flash drive.** A USB flash drive is an external, removable storage device that uses flash memory and interfaces with a computer through a USB port.
- **Memory card.** A memory card is a type of portable external storage media, such as a Compact Flash card, used to store media or data files.

output hardware components :-

- **Monitor.** A monitor is an output device similar to a TV screen that displays information, documents or images generated by the computing device.
- **Printer.** Printers render electronic data from a computer into printed material.
- **Speaker.** A speaker is an external audio output device that connects to a computer to generate a sound output.
- **Headphones, earphones, ear buds.** Similar to speakers, these devices provide audio output that's audible only to a single listener.

~Assignment~

- What do you understand by software ?
- Parts of software ?
- Uses of software ?
- Difference between software and hardware ?
- What do you understand by hardware?

Unit 3rd

- Types of computer.
- Programming language.
- Programming environment.
- Computer Organization.
- Drives.
- Files.
- Directories
- Alu. CPU. Cu.
- Assignment.

Types of computer?

There are 5 types of computer.

- Super Computer.
- Mainframe computers.
- Mini computers.
- Work stations.
- Micro computers.

A. Super computers: – Super computer are the fastest calculating devices ever they invented. A desktop micro computer process in micro second. And a super computer can operate at speed in nanosecond one thousand to one million times fast as micro computer. Super computer help in analyzing weather pattern , like pollution, shedding light, global warming and the depletion of the earth's ozone layer.



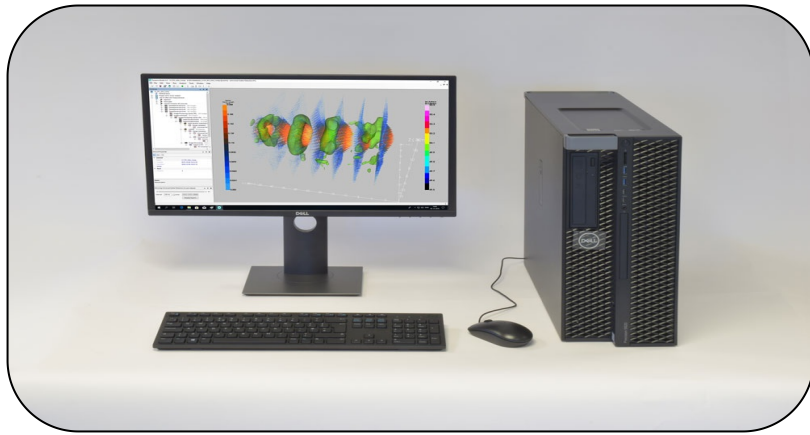
B. Mainframe computer: – mainframe fall into higher category of computer. They are specially designed proprietary circuits instead of just single microprocessor as their CPU. They are higher connectivity large data processing capabilities.



C. Mini computer :– Mini computer are relatively larger and faster computer. Apart from single user environment, they also support multiuser environment.



D. Work stations :– Workstations are specialized, single user computers with many of the features of as personal computer but with the processing power of a minicomputer the powerful machine are popular among scientist engineers, graph artists, animators and programmers users who need a great deal of number crunching power workstations personal computers. Workstation making them perfect for advanced design, modeling, animation and video–editing.



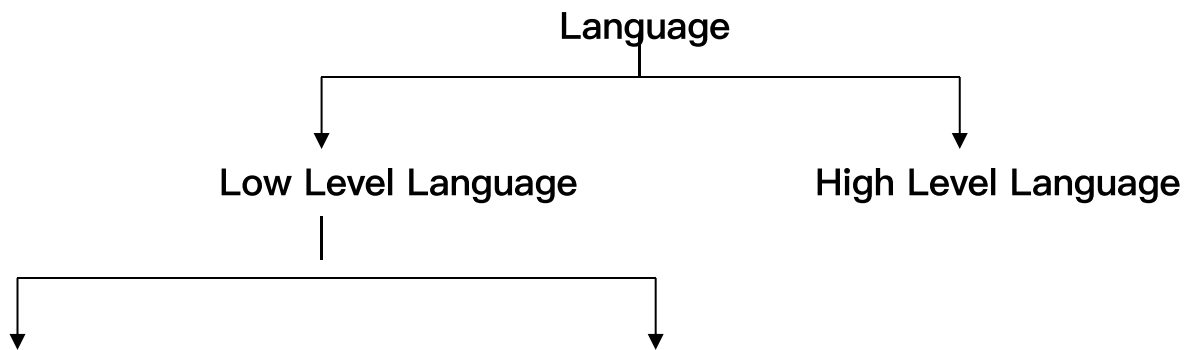
E. Micro computer :- A micro computer is smallest, least expensive of all computers. The main differences between microcomputer and mainframe or mini computer are that microcomputer have smallest memory and less power, are physically smaller and permit fewer peripherals to be attached. Microcomputers are also called personal computers.



Programming Languages .

There are two types of language.

- Low Level Language.
- High Level Language.



Machine Language

Assembly Language

Machine language: – It contains the strings of binary number (0's and 1's) and is machine dependent. It means that the machine language for every processor is different . An instruction prepared in any machine language has two part format, as shown below.

Opcode (Operation code)	Operand
-----------------------------------	----------------

The first part is the operation which tells the computer what function to perform. Every computer has an operation code for each of its function.

The second part of the instructions is the operand and it tells the computer where to find or store the data or other instruction.

Advantages:–

- Very efficient.
- Requires less storage space.

Disadvantage:–

- Machine dependent.
- Cost is high.
- Programming is difficult.

Assembly language: – In this type of language replaced the string the binary numbers with short acronyms corresponding to the fundamental function that the computer processor is able to perform. Each mnemonic of assembly language has a unique machine code.

Assembly language or symbolic language is a language that uses symbols instead of numbers. A translator program that translates an assembly code with the computer machine code is called as assembler.

Advantage:–

- They are time saving and reduce details.
- They are easier to debug.

Disadvantage:–

- The coding to assembly language is time consuming.

High Level language: – To overcome the difficulties of low level languages, the high level language was developed. These languages, instead of being machine based, are oriented more toward solving the problem.

Advantage:–

- Easier to maintain
- They are not machine oriented.
- They provide better documentations.

S R	High level languages	Assembly languages	Machine languages
1	These are easily understandable	Less understandable than high level language more than machine language	Program are less understandable
2	Program are portable	Not portable to processor to same interpreter only	Not portable only processor of same architecture
3	Debugging is easier	Debugging is more complex than in high level languages	Not good as high level language and assembly language.

Most Commonly used programming languages:–

1. **Python:** – Python is one of the most widely used user friendly programming language it is an open source and easy to learn programming language.

Advantage:–

- Python is easy to read and easy to understand and easy to write.
- It integrates with other programming language like C++ and java.

Disadvantage:–

- Python is not suitable for developing mobile application and game.
- Python work with the interpreter that's why it is slower than other programming language like C and C++.

2. **JAVA:** – java is a simple, secure, platform-independent, reliable, architecture-neutral high level programming language developed by sun Microsystems in 1995.

Advantage:–

- Java is easy to write compile learn and debug as compare to other programming languages.
- It is capable of performing multiple task same time.

Disadvantage:–

- Java consume more memory in slower than other programming languages

3. **C:** – c is a popular simple and flexible general purpose computer programming language “**Dennis M Ritchie**” developed it in 1972.

Advantage:–

- It is used to perform complex calculations and operations.

Disadvantage:–

- In the c programming language it is difficult to find error.

4. **C++:-** is one of the thousand of programming language that be used to developed software. C++ programming language are developed by “BJARNE STROUSTRUP”
In 1980.

Advantage:-

- C++ is a simple and portable structured programming language.

Disadvantage:-

- C++ cannot support garbage collection.

5. **C# :-** Is a modern general purpose and object oriented programming language used with **XML** based web services on the **Dot net** platform.

Advantage:-

- C# is a modern type safe easy fast and open source programming language.

Disadvantage:-

- C# is less flexible because it is completely based on Microsoft dot net framework.

6. **PHP:-** Php stand for “**hypertext preprocessor**”. It is an open source powerful server-side scripting language mainly used for create static and dynamic websites
It is developed by “**Rasmus laired in 1994**”.

Advantage:–

- PHP is a more secure and easy to use programming language.
- It supports powerful online libraries.

Disadvantage:–

- PHP is not capable of handling a larger number of application and not suitable for large application.

Programming environments:–

Programming environments is the collection of tools used in the development of software.

This collection may consist.

- A file system
- A text editor
- A linker
- A compiler
- Integrated tools.

These tools may be access through a uniform interface (GUI).

1. **Microsoft visual studio.NET:–** which is a large collection of software development tools, used through a windows interface. It is used to develop software in following languages.

- C#.
- Visual basic .NET.
- Jscript (MS JavaScript version).
- J# (MS java version).
- Managed C++.

2. **Net beans:**– Net Beans IDE is a free open source, integrated development environment (IDE). Developed by “**Oracle corporation and sun micro system**” that enables you to develop web application and mobile application.
3. **Turbo C, C++:**– Turbo c is a discontinued integrated development environment and compiler for the C programming language. Introduced in 13 may 1987 and developed by “**Borland**”.
4. **Dreamweaver:**– Adobe Dreamweaver is a website creation program that allows you to build and publish web pages almost anywhere with software that support HTML and Css and Java script.
5. **Arduino:**– Arduino is an open source electronics platform based on easy to use hardware and software. Developed by “Massimo Banzi” in 2005.

Drives:–

A **drive** is a location ([medium](#)) that is capable of storing and reading information that is not easily removed, like a [disk](#) or [disc](#). All drives store [files](#) and [programs](#) used by your computer. For example, when you write a letter in a [word processor](#), the program is loaded from the [hard drive](#). When you save the document, it's saved to the hard drive or other disk or drive. The picture is an example of different drives listed in Microsoft Windows [My Computer](#).



In the picture, drive A: is the [floppy drive](#), C: is the primary [hard drive](#), D: and E: are [partitions](#), and F: is the [CD-ROM](#) drive. The CD-ROM drive is usually the last [drive letter](#). In most situations, the hard drive is the C: drive, and a CD-ROM or other disc drive is the D: drive.

Types of computer drives

- Bernoulli drive (obsolete)
- Disc drives: Blu-ray, CD-R, CD-ROM, CD-RW, and DVD.
- Flash drive.
- Floppy disk drive (obsolete)
- Hard drive
- Local drive
- LS-120 (obsolete)
- Network drive
- RAM disk
- SSD
- SuperDisk (obsolete)

- Tape drive
- USB drive
- Virtual drive
- Zip drive (obsolete)

Fixed drive?

A **fixed drive** is any drive inside the computer that is not easily removable or portable. For example, most hard drives inside the computer are considered a fixed drive.

Portable drive and removable disk?

A **portable drive** and **removable disk** is any drive or disk that can be transported between computers. The most common portable drives today are the USB card readers, USB jump drives, and USB external hard disk drives.

Files:–

A **file** is an object on a [computer](#) that stores [data](#), [information](#), settings, or commands used with a computer [program](#). On a computer there are three types of files, [application files](#), [data files](#), and [system files](#).

In a [GUI](#) (graphical user interface), such as [Microsoft Windows](#), files display as [icons](#) that relate to the program that opens the file. For example, all [PDF](#) icons appear the same and open in [Adobe](#) Acrobat or the reader [associated](#) with PDF files. If a program is associated with a program, [double-clicking](#) the icon opens it in the default program.

How are files created?

A file is created using a computer software program. For example, to create a text file you would use a text editor, to create an image file you would use an image editor, and to create a document you would use a word processor.

- How to create a computer file.
- How to open, view, and edit the contents of a file on a computer.

Where are files stored?

Computer files are stored on a drive (e.g., the hard drive), disc (e.g., DVD), and a diskette (e.g., floppy disk) and may be in a folder (directory) on that medium.

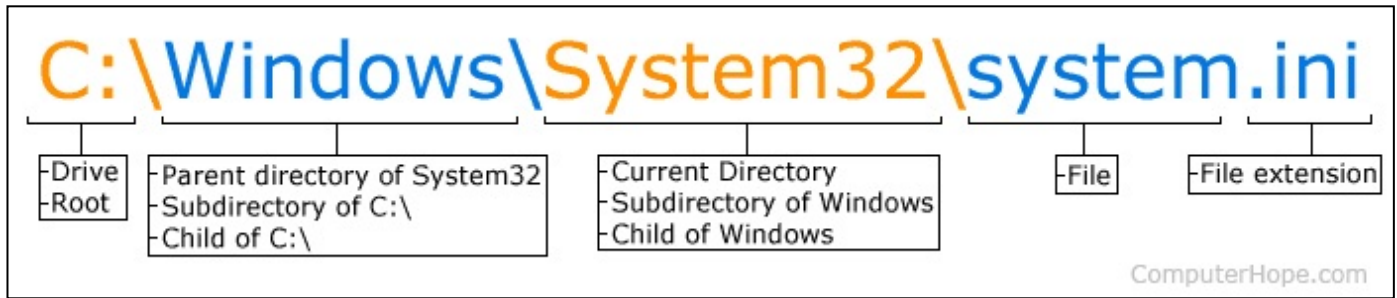
Directories:–

A **directory** is a location for storing files on your computer. Directories are found in a [hierarchical file system](#), such as [Linux](#), [MS-DOS](#), [OS/2](#), and [Unix](#).

Pictured is an example of output from the Windows/DOS [tree command](#). It shows all the local and [subdirectories](#) (e.g., the "big" directory in the "cdn" directory). When looking at this overview, the [current directory](#) is the [root directory](#) of the C: drive. It's called the "root" directory because there is nothing beneath it, and the other directories "branch" from it. If you are using an operating system with multiple user accounts, the directory may also be referred to as a [home directory](#).

Overview of a directory and path

Below is an example of what a directory path would look like in MS-DOS.



In the picture, **C:** is the drive letter and the current directory is **System32**, which is a subdirectory of the **Windows** directory.

ALU:–

An arithmetic–logic unit is the part of a [central processing unit](#) that carries out arithmetic and logic operations on the [operands](#) in computer [instruction words](#).

In some processors, the ALU is divided into two units: an arithmetic unit (AU) and a logic unit (LU). Some processors contain more than one AU -- for example, one for fixed–point operations and another for floating–point operations.

In computer systems, floating–point computations are sometimes done by a [floating–point](#) unit (FPU) on a separate chip called a numeric [coprocessor](#).

How does an arithmetic–logic unit work?

Typically, the ALU has direct input and output access to the processor controller, main memory (random access memory or RAM in a personal computer) and input/output devices. Inputs and outputs flow along an electronic path that is called a bus.

The input consists of an instruction word, sometimes called a machine instruction word, that contains an operation code or "opcode," one or more operands and sometimes a format code. The operation code tells the ALU what operation to perform and the operands are used in the operation.

For example, two operands might be added together or compared logically. The format may be combined with the opcode and tells, for example, whether this is a fixed-point or a floating-point instruction.

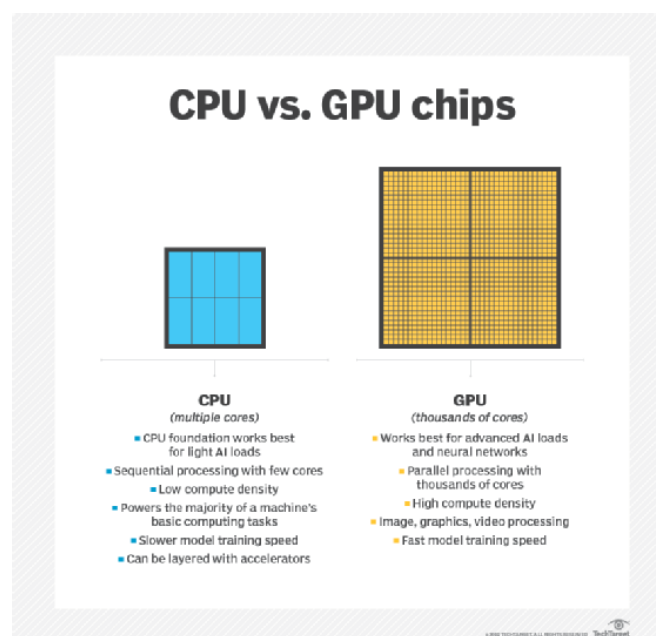
The output consists of a result that is placed in a storage register and settings that indicate whether the operation was performed successfully. If it isn't, some sort of status will be stored in a permanent place that is sometimes called the machine status word.

What type of functions do ALUs support?

In computer science, ALUs serve as a combinational digital circuit that performs arithmetic and bitwise operations on binary numbers. This is a foundational building block of arithmetic logic circuits for numerous types of control units and computing circuits including central processing units (CPUs), and graphics processing units.

Long before modern PCs, ALUs first helped to support microprocessors and transistors in the 1970s.

The following are a few examples of bitwise logical operations and basic arithmetic operations supported by ALUs:



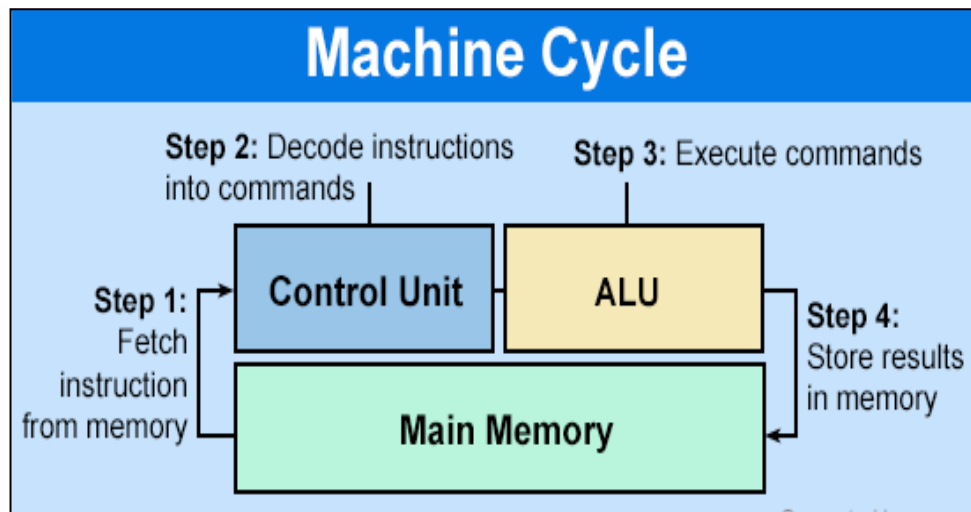
- **Addition.** Adds A and B with carry-in or carry-out sum at Y.
- **Subtraction.** Subtracts B from A or vice versa with the difference at Y and carry-in or carry-out.
- **Increment.** Where A or B is increased by one and Y represents the new value.
- **Decrement.** Where A or B is decreased by one and Y represents the new value.
- **AND.** The bitwise logic AND of A and B is represented by Y.
- **OR.** The bitwise logic OR of A and B is represented by Y.
- **Exclusive-OR.** The bitwise logic XOR of A and B is represented by Y.

ALU shift functions cause A or B operands to shift, either right or left, with the new operand represented by Y. Complex ALUs utilize barrel shifters to shift A or B operands by any number of bits in a single operation.

CU:–

A **control unit** or **CU** is circuitry that directs operations within a computer's processor. It lets the computer's logic unit, memory, and both input and output devices know how to respond to instructions received from a program. Examples of devices that utilize control units include CPUs and GPUs.

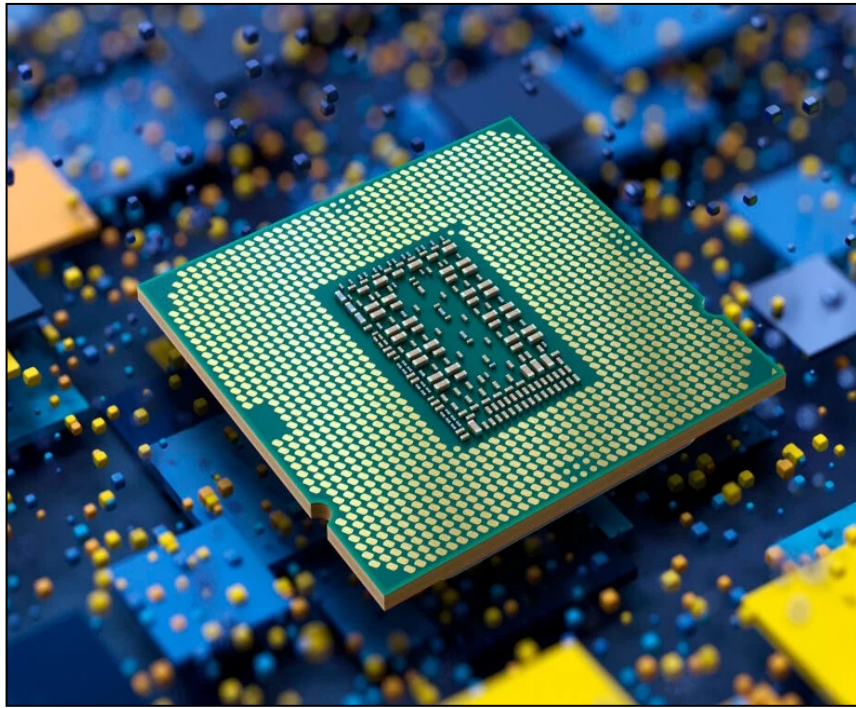
A control unit works by receiving input information that it converts into control signals, which are then sent to the central processor. The computer's processor then tells the attached hardware what operations to carry out. The functions that a control unit performs are dependent on the type of CPU, due to the variance of architecture between different manufacturers. The following diagram illustrates how instructions from a program are processed.



CPU:–

Stands for "Central Processing Unit." The CPU is the primary [component](#) of a computer that processes instructions. It runs the [operating system](#) and [applications](#), constantly receiving [input](#) from the user or active software programs. It processes the data and produces [output](#), which may be stored by an application or displayed on the screen. is the [electronic circuitry](#) that executes [instructions](#) comprising a [computer program](#). The CPU performs basic [arithmetic](#), logic, controlling, and [input/output](#) (I/O) operations specified by the instructions in the program. This contrasts with external components such as [main memory](#) and I/O circuitry,^[1] and specialized processors such as [graphics processing units](#)

The CPU contains at least one [processor](#), which is the actual chip inside the CPU that performs calculations. For many years, most CPUs only had one processor, but now it is common for a single CPU to have at least two processors or "processing cores." A CPU with two processing cores is called a [dual-core](#) CPU and models with four cores are called [quad-core](#) CPUs. High-end CPUs may have six (hexa-core) or even eight (octo-core) processors. A computer may also have more than one CPU, which each have multiple cores. For example, a [server](#) with two hexa-core CPUs has a total of 12 processors.



~Assignment~

- What do know about programming language ?
- Write down about Alu ,Cu,Cpu ?
- What do you know about Files ?
- What do you know about Drives ?

~ Unit 4th ~

- Binary number system.
- Octal number system.
- Decimal number system.
- Hexa decimal number system.
- Binary to decimal.
- Binary to octal.
- Binary to hexa decimal.
- Algorithm.
- Flow chart.
- Assignment.

Number Systems

The language we use to communicate with each other is comprised of words and characters. We understand numbers, characters and words. But this type of data is not suitable for computers. Computers only understand the numbers.

So, when we enter data, the data is converted into electronic pulse. Each pulse is identified as code and the code is converted into numeric format by ASCII. It gives each number, character and symbol a numeric value (number) that a computer understands. So to understand the language of computers, one must be familiar with the number systems.

The Number Systems used in computers are:

- Binary number system
- Octal number system
- Decimal number system
- Hexadecimal number system

Binary number system

It has only two digits '0' and '1' so its base is 2. Accordingly, In this number system, there are only two types of electronic pulses; absence of electronic pulse which represents '0' and presence of electronic pulse which represents '1'. Each digit is called a bit. A group of four bits (1101) is called a nibble and group of eight bits (11001010) is called a byte. The position of each digit in a binary number represents a specific power of the base (2) of the number system.

Octal number system

It has eight digits (0, 1, 2, 3, 4, 5, 6, 7) so its base is 8. Each digit in an octal number represents a specific power of its base (8). As there are only eight digits, three bits ($2^3=8$) of binary number system can convert any octal number into binary number. This number system is also used to shorten long binary numbers. The three binary digits can be represented with a single octal digit.

Decimal number system

This number system has ten digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) so its base is 10. In this number system, the maximum value of a digit is 9 and the minimum value of a digit is 0. The position of each digit in decimal number represents a specific power of the base (10) of the number system. This number system is widely used in our day to day life. It can represent any numeric value.

Hexadecimal number system

This number system has 16 digits that ranges from 0 to 9 and A to F. So, its base is 16. The A to F alphabets represent 10 to 15 decimal numbers. The position of each digit in a hexadecimal number represents a specific power of base (16) of the number system. As there are only sixteen digits, four bits ($2^4=16$) of binary number system can convert any hexadecimal number into binary number. It is also known as alphanumeric number system as it uses both numeric digits and alphabets.

Binary to Decimal conversion?

The decimal number is equal to the sum of binary digits (d_n) times their power of 2 (2^n):

$$\text{decimal} = d_0 \times 2^0 + d_1 \times 2^1 + d_2 \times 2^2 + \dots$$

Example #1

Find the decimal value of 111001_2 :

binary number:	1	1	1	0	0	1
power of 2:	2^5	2^4	2^3	2^2	2^1	2^0

$$111001_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 57_{10}$$

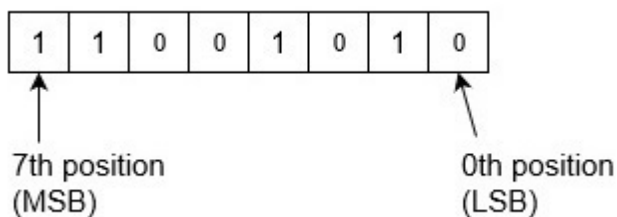
Example-2

Find the decimal value of 100011_2 :

binary number:	1	0	0	0	1	1
power of 2:	2^5	2^4	2^3	2^2	2^1	2^0

$$100011_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 35_{10}$$

Example-3 – Convert binary number 11001010 into decimal number. Since there is no binary point here and no fractional part. So,



Binary to decimal is,

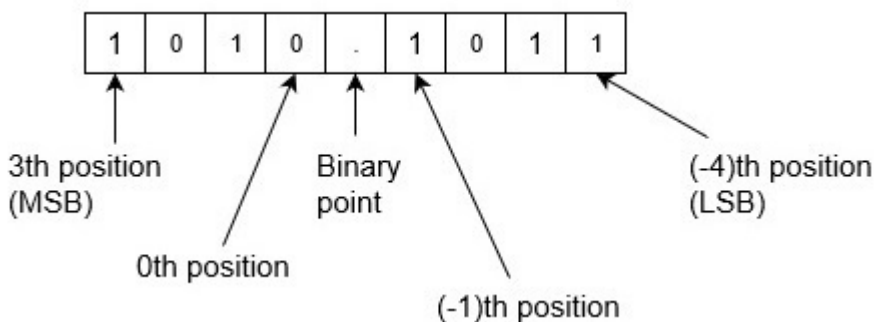
$$= (11001010)_2$$

$$= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 128 + 64 + 0 + 0 + 8 + 0 + 2 + 0$$

$$= (202)_{10}$$

Example-4 – Convert binary number 1010.1011 into decimal number. Since there is a binary point here with fractional part. So,



Binary to decimal is,

$$= (1010.1011)_2$$

$$= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$

$$= 8 + 0 + 2 + 0 + 0.5 + 0 + 0.125 + 0.0625$$

$$= (10.6875)_{10}$$

Decimal to binary conversion ?

Decimal to Binary Conversion



Step 1: Divide the given number **13** repeatedly by 2 until you get '0' as the quotient

$$\begin{array}{lcl} 13 \div 2 = 6 & \text{(Remainder 1)} & \\ 6 \div 2 = 3 & \text{(Remainder 0)} & \\ 3 \div 2 = 1 & \text{(Remainder 1)} & \\ 1 \div 2 = 0 & \text{(Remainder 1)} & \end{array}$$

Step 2: Write the remainders in the reverse order **1 1 0 1**

$$\therefore 13_{10} = 1101_2$$

(Decimal) (Binary)

Division by 2	Quotient	Remainder
$13 \div 2$	6	1 (LSB)
$6 \div 2$	3	0
$3 \div 2$	1	1
$1 \div 2$	0	1 (MSB)

- **Example 1:** Convert 174_{10} to binary.

Division by 2	Quotient	Remainder
$174 \div 2$	87	0 (LSB)
$87 \div 2$	43	1
$43 \div 2$	21	1
$21 \div 2$	10	1
$10 \div 2$	5	0
$5 \div 2$	2	1
$2 \div 2$	1	0
$1 \div 2$	0	1 (MSB)

Solution: For decimal to binary conversion, let us first divide the given number by 2 and note down the remainders as shown in the following table.

After noting the remainders, we write them in the reverse order such that the Most Significant Bit (MSB) is written first, and the Least Significant Bit is written in the end. Hence, the binary equivalent for the given decimal number 174_{10} is 10101110_2 .

- **Example 2:** Convert the following decimal number to binary: 156_{10}

Division by 2	Quotient	Remainder
$156 \div 2$	78	0 (LSB)
$78 \div 2$	39	0
$39 \div 2$	19	1
$19 \div 2$	9	1
$9 \div 2$	4	1

$4 \div 2$	2	0
$2 \div 2$	1	0
$1 \div 2$	0	1 (MSB)

Solution: To convert 156_{10} to the binary number system, let us divide it repeatedly by 2 and note the remainders as shown below.

Let us write the remainders in the reverse order. Hence, the binary equivalent for the given decimal number 156_{10} is 10011100_2 .

Binary to octal conversion ?

Example 1: Convert 1010101_2 to octal

Solution:

Given binary number is 1010101_2

First, we convert given binary to decimal

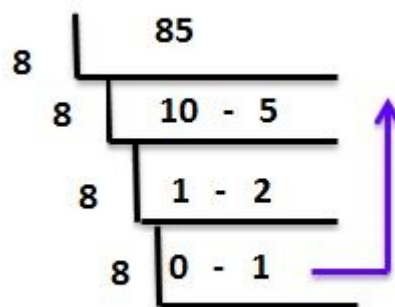
$$1010101_2 = (1 * 2^6) + (0 * 2^5) + (1 * 2^4) + (0 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0)$$

$$= 64 + 0 + 16 + 0 + 4 + 0 + 1$$

$$= 64 + 21$$

$$010101_2 = 85 \text{ (Decimal form)}$$

Now we will convert this decimal to octal form



Therefore, the equivalent octal number is 125_8 .

Example 2: Convert 01101_2 to octal

Solution:

Given binary number is 01101_2

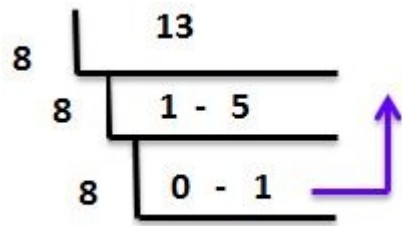
First we convert given binary to decimal

$$01101_2 = (0 * 2^4) + (1 * 2^3) + (1 * 2^2) + (0 * 2^1) + (1 * 2^0)$$

$$= 0 + 8 + 4 + 0 + 1$$

$01101_2 = 13$ (Decimal form)

Now we will convert this decimal to octal form



Therefore, the equivalent octal number is 15₈.

Method 2: Direct Method of Octal to Binary Conversion

Another method, surely an easier and less complicated one, is to directly convert a number from octal to binary by referring to the octal to binary table given above. In this method, we convert every digit of the given octal number to its binary equivalent by referring to the above chart. For example, using this method we can directly write 3₈ as 011₂, 5₈ as 101₂, etc.

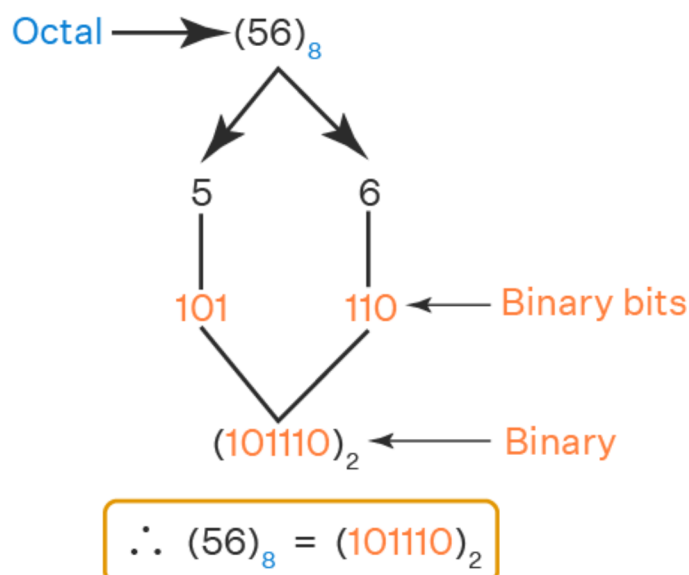
Let us convert the same number 56₈ from octal to binary using the direct method. There are two digits in this number, 5 and 6. Referring to the above octal to binary chart:

$$5_8 \rightarrow 101_2$$

$$6_8 \rightarrow 110_2$$

By combining these two, we will get, $56_8 = 101110_2$. Observe the image given below for your reference.

Octal to Binary Conversion



Using Grouping

Since, there are only 16 digits (from 0 to 7 and A to F) in hexadecimal number system, so we can represent any digit of hexadecimal number system using only 4 bit as following below.

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

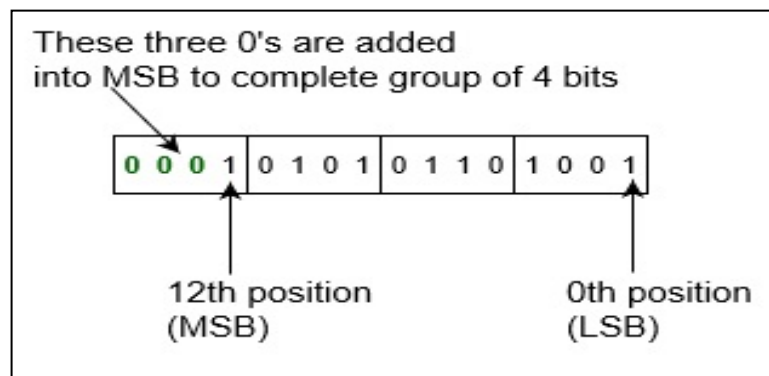
So, if you make each group of 4 bit of binary input number, then replace each group of binary number from its equivalent hexadecimal digits. That will be hexadecimal number of given number. Note that you can add any number of 0's in leftmost bit (or in most significant bit) for integer part and add any number of 0's in rightmost bit (or in least significant bit) for fraction part for completing the group of 4 bit, this does not change value of input binary number.

So, these are following steps to convert a binary number into hexadecimal number.

- Take binary number
- Divide the binary digits into groups of four (starting from right) for integer part and start from left for fraction part.
- Convert each group of four binary digits to one hexadecimal digit.

This is simple algorithm where you have to grouped binary number and replace their equivalent hexadecimal digit.

Example-1 – Convert binary number 1010101101001 into hexadecimal number. Since there is no binary point here and no fractional part. So,



Therefore, Binary to hexadecimal is,

$$= (1010101101001)_2$$

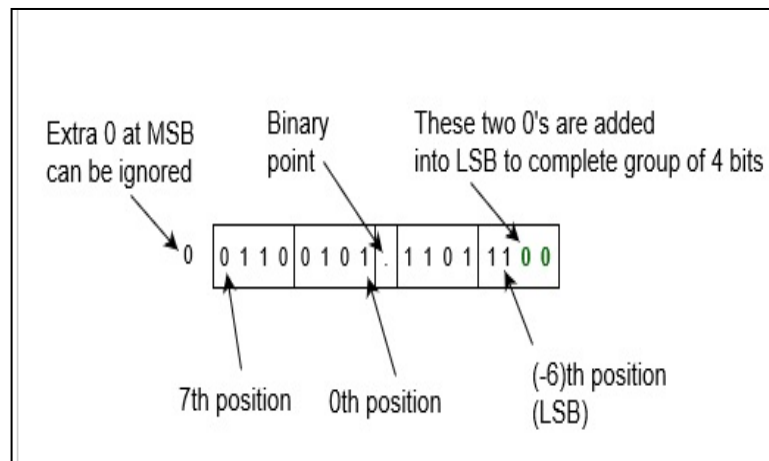
$$= (1\ 0101\ 0110\ 1001)_2$$

$$= (0001\ 0101\ 0110\ 1001)_2$$

$$= (1\ 5\ 6\ 9)_{16}$$

$$= (1569)_{16}$$

Example-2 – Convert binary number 001100101.110111 into hexadecimal number. Since there is binary point here and fractional part. So,



Therefore, Binary to hexadecimal is,

$$= (001100101.110111)_2$$

$$= (0\ 0110\ 0101\ .\ 1101\ 1100)_2$$

$$= (0110\ 0101\ .\ 1101\ 1100)_2$$

$$= (6\ 5\ .\ D\ C)_{16}$$

$$= (65.DC)_{16}$$

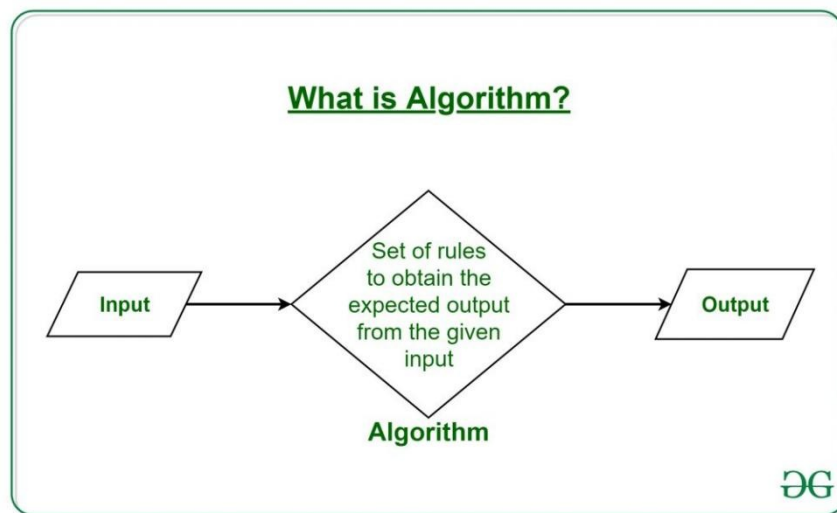
These are above simple conversions binary number to hexadecimal number.

~Unit 5th~

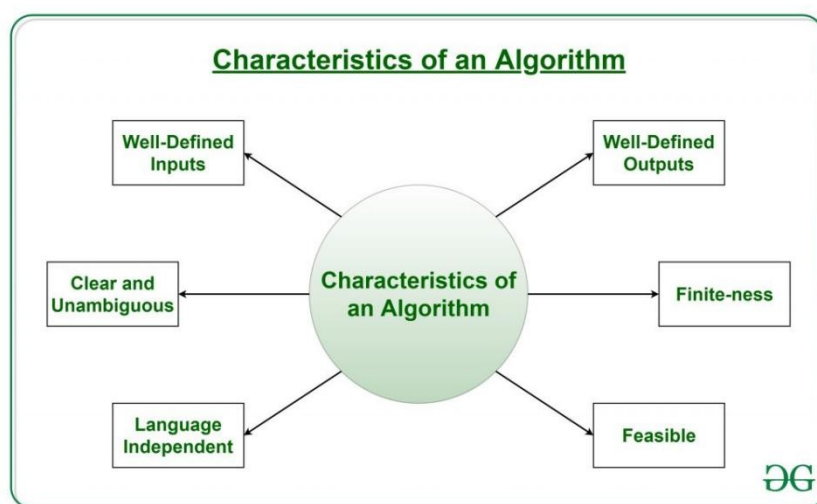
- What is algorithms.
- Characteristics of algorithms.
- Advantage of algorithms.
- What is flow chart.
- Characteristics of flow chart.
- Assignment.

What is an Algorithm? Algorithm Basics

The word **Algorithm** means ” A set of finite rules or instructions to be followed in calculations or other problem–solving operations ” Or ” A procedure for solving a mathematical problem in a finite number of steps that frequently involves recursive operations”.



What are the Characteristics of an Algorithm?



As one would not follow any written instructions to cook the recipe, but only the standard one. Similarly, not all written instructions for programming is an

algorithms. In order for some instructions to be an algorithm, it must have the following characteristics:

- **Clear and Unambiguous:** The algorithm should be clear and unambiguous. Each of its steps should be clear in all aspects and must lead to only one meaning.
- **Well-Defined Inputs:** If an algorithm says to take inputs, it should be well-defined inputs. It may or may not take input.
- **Well-Defined Outputs:** The algorithm must clearly define what output will be yielded and it should be well-defined as well. It should take at least 1 output.
- **Finite-ness:** The algorithm must be finite, i.e. it should terminate after a finite time.
- **Feasible:** The algorithm must be simple, generic, and practical, such that it can be executed with the available resources. It must not contain some future technology or anything.
- **Language Independent:** The Algorithm designed must be language-independent, i.e. it must be just plain instructions that can be implemented in any language, and yet the output will be the same, as expected.

Properties of Algorithm:

- It should terminate after a finite time.
- It should produce at least one output.
- It should take zero or more input.
- It should be deterministic means giving the same output for the same input case.
- Every step in the algorithm must be effective i.e. every step should do some work.

Types of Algorithms:

There are several types of algorithms available. Some important algorithms are:

1. **Brute Force Algorithm**: It is the simplest approach for a problem. A brute force algorithm is the first approach that comes to finding when we see a problem.

2. **Recursive Algorithm**: A recursive algorithm is based on recursion. In this case, a problem is broken into several sub-parts and called the same function again and again.

3. **Backtracking Algorithm**: The backtracking algorithm basically builds the solution by searching among all possible solutions. Using this algorithm, we keep on building the solution following criteria. Whenever a solution fails we trace back to the failure point and build on the next solution and continue this process till we find the solution or all possible solutions are looked after.

4. **Searching Algorithm**: Searching algorithms are the ones that are used for searching elements or groups of elements from a particular data structure. They can be of different types based on their approach or the data structure in which the element should be found.

5. **Sorting Algorithm**: Sorting is arranging a group of data in a particular manner according to the requirement. The algorithms which help in performing this function are called sorting algorithms. Generally sorting algorithms are used to sort groups of data in an increasing or decreasing manner.

6. **Hashing Algorithm**: Hashing algorithms work similarly to the searching algorithm. But they contain an index with a key ID. In hashing, a key is assigned to specific data.

7. **Divide and Conquer Algorithm**: This algorithm breaks a problem into sub-problems, solves a single sub-problem and merges the solutions together to get the final solution. It consists of the following three steps:

- Divide

- Solve
- Combine

8. Greedy Algorithm: In this type of algorithm the solution is built part by part. The solution of the next part is built based on the immediate benefit of the next part. The one solution giving the most benefit will be chosen as the solution for the next part.

9. Dynamic Programming Algorithm: This algorithm uses the concept of using the already found solution to avoid repetitive calculation of the same part of the problem. It divides the problem into smaller overlapping subproblems and solves them.

10. Randomized Algorithm: In the randomized algorithm we use a random number so it gives immediate benefit. The random number helps in deciding the expected outcome.

To learn more about the types of algorithms refer to the article about “Types of Algorithms”.

Advantages of Algorithms:

- It is easy to understand.
- An algorithm is a step-wise representation of a solution to a given problem.
- In Algorithm the problem is broken down into smaller pieces or steps hence, it is easier for the programmer to convert it into an actual program.

Disadvantages of Algorithms:

- Writing an algorithm takes a long time so it is time-consuming.
- Understanding complex logic through algorithms can be very difficult.
- Branching and Looping statements are difficult to show in Algorithms(**imp**).

How to Design an Algorithm?

In order to write an algorithm, the following things are needed as a pre-requisite:

1. The **problem** that is to be solved by this algorithm i.e. clear problem definition.
2. The **constraints** of the problem must be considered while solving the problem.
3. The **input** to be taken to solve the problem.
4. The **output** to be expected when the problem is solved.
5. The **solution** to this problem, is within the given constraints.

Then the algorithm is written with the help of the above parameters such that it solves the problem.

Example: Consider the example to add three numbers and print the sum.

- **Step1:Fulfilling the pre-requisites**

As discussed above, in order to write an algorithm, its pre-requisites must be fulfilled.

1. **The problem that is to be solved by this algorithm:** Add 3 numbers and print their sum.
2. **The constraints of the problem that must be considered while solving the problem:** The numbers must contain only digits and no other characters.
3. **The input to be taken to solve the problem:** The three numbers to be added.
4. **The output to be expected when the problem is solved:** The sum of the three numbers taken as the input i.e. a single integer value.
5. **The solution to this problem, in the given constraints:** The solution consists of adding the 3 numbers. It can be done with the help of '+' operator, or bit-wise, or any other method.

Algorithm to add 3 numbers and print their sum:

1. START.
2. Declare 3 integer variables num1, num2 and num3.

3. Take the three numbers, to be added, as inputs in variables num1, num2, and num3 respectively.
4. Declare an integer variable sum to store the resultant sum of the 3 numbers.
5. Add the 3 numbers and store the result in the variable sum.
6. Print the value of the variable sum.
7. END.

Unit 6

What is text processing?

The term text processing refers to the automation of analyzing electronic text. This allows machine learning models to get structured information about the text to use for analysis, manipulation of the text, or to generate new text.

Text processing is one of the most common tasks used in machine learning applications such as language translation, sentiment analysis, spam filtering, and many others.

How is text processing used?

Topic analysis — This technique interprets and categorizes large collections of text into topics or themes.

Sentiment analysis — This function automatically detects the emotional undertones of text and classifies them as positive, negative, or neutral.

Intent detection — This classification model detects the intent, purpose, or goal of the text. For example, it may determine whether the intent is to gain information, make a purchase, or unsubscribe from the company.

Language classification — This classifies text based on which language it's written in.

What is data processing?

Data processing occurs when data is collected and translated into usable information. Usually performed by a data scientist or team of data scientists, it is important for data processing to be done correctly as not to negatively affect the end product, or data output.

Data processing starts with data in its raw form and converts it into a more readable format (graphs, documents, etc.).

Six stages of data processing

1. Data collection

Collecting data is the first step in data processing. Data is pulled from available sources, including data lakes and data warehouses.

2. Data preparation

Once the data is collected, it then enters the data preparation stage. Data preparation, often referred to as “pre-processing” is the stage at which raw data is cleaned up and organized for the following stage of data processing.

3. Data input

The clean data is then entered into its destination (perhaps a CRM like Salesforce or a data warehouse like Redshift), and translated into a language that it can understand.

4. Processing

During this stage, the data inputted to the computer in the previous stage is actually processed for interpretation. Processing is done using machine learning algorithms, though the process itself may vary slightly depending on the source of data being processed (data lakes, social networks, connected devices etc.

5. Data output/interpretation

The output/interpretation stage is the stage at which data is finally usable to non-data scientists. It is translated, readable, and often in the form of graphs, videos, images, plain text, etc.).

6. Data storage

The final stage of data processing is storage. After all of the data is processed, it is then stored for future use. While some information may be put to use immediately, much of it will serve a purpose later on. Plus, properly stored data is a necessity for compliance with data protection legislation like GDPR.

What is E-mail?



E-mail is defined as the transmission of messages on the Internet. It is one of the most commonly used features over communications networks that may contain text, files, images, or other attachments.

Email messages are conveyed through email servers; it uses multiple protocols within the TCP/IP suite. For example, SMTP is a protocol, stands for simple mail transfer protocol and used to send messages whereas other protocols IMAP or POP are used to retrieve messages from a mail server.

Popular Computer Networks:

- Local Area Network (LAN)
- Metropolitan Area Network (MAN)
- Wide Area Network (WAN)

Local Area Network (LAN):

As the name suggests, the local area network is a computer network that operates in a small area, i.e., it connects computers in a small geographical area like within an office, company, school, or any other organization.

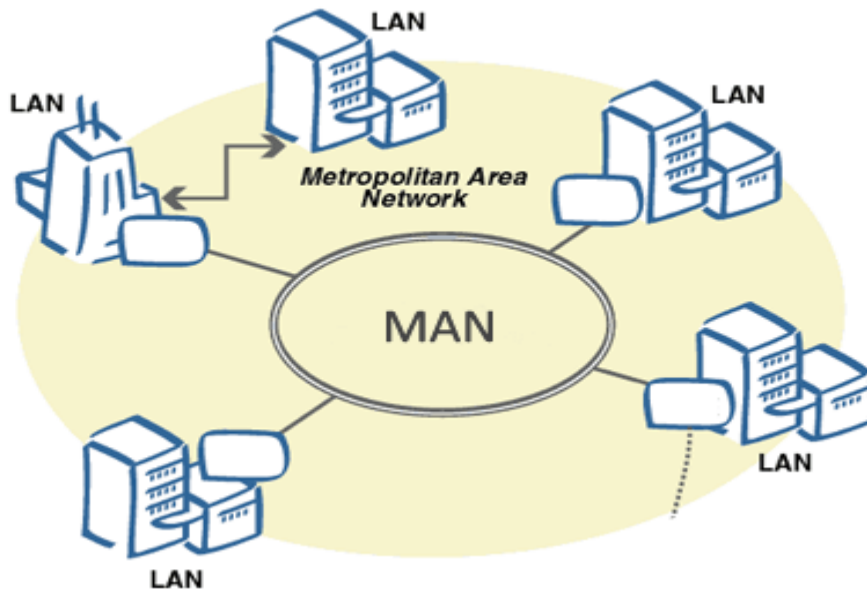
A local area network may be a wired or wireless network or a combination of both. The devices in a LAN are generally connected using an Ethernet cable, which offers an interface to connect multiple devices like router, switches, and computers.

Benefits of LAN:

- It offers a higher operating speed than WAN and MAN.

- It is less expensive and easy to install and maintain.
- It perfectly fulfills the requirement of a specific organization, such as an office, school, etc.
- It can be wired or wireless or a combination of both.
- It is more secure than other networks as it is a small set up that can be easily taken care of.

Metropolitan Area Network (MAN):



MAN is a high-speed network that spreads over a large geographical area such as a metro city or town. It is set up by connecting the local area networks using routers and local telephone exchange lines. It can be operated by a private company, or it may be a service provided by a company such as a local telephone company.

The area covered by MAN is larger than the LAN but smaller than a WAN. Its network ranges from 5 to 50 km. Furthermore, it also provides uplinks for connecting LANs to WANs and the internet

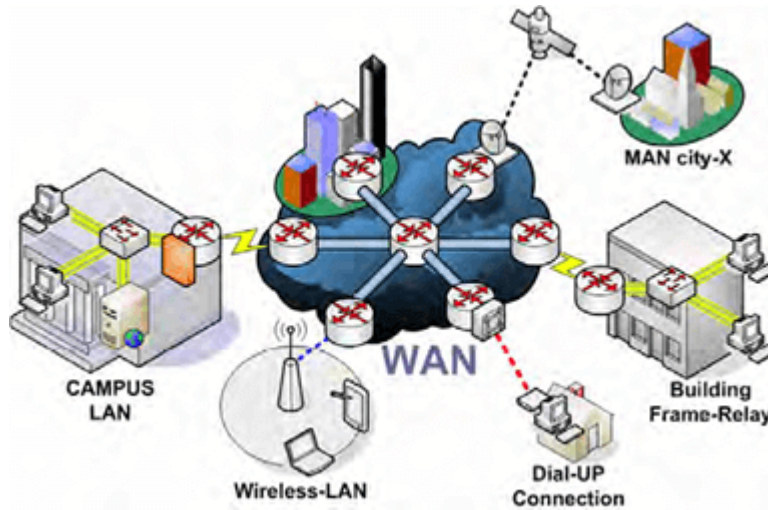
Examples of MAN:

- Cable TV Network
- Telephone service provides that provide high-speed DSL lines

Advantages of MAN:

- **Less Expensive:** It is less expensive to set up a MAN and to connect it to a WAN.
- **High Speed:** The speed of data transfer is more than WAN.
- **Local Emails:** It can send local emails fast.

Wide Area Network (WAN):



WAN extends over a large geographical area. It is not confined within an office, school, city, or town and is mainly set up by telephone lines, fiber optic, or satellite links. It is mostly used by big organizations like banks and multinational companies to communicate with their branches and customers across the world. Although it is structurally similar to MAN, it is different from MAN in terms of its range, e.g., MAN covers up to 50 Kms, whereas WAN covers distances larger than 50 Km, e.g., 1000km or more.

A WAN works by using TCP/IP protocol in combination with networking devices such as switches, routers, firewalls, and modems. It does not connect individual computers; rather, they are designed to link small networks like LANs and MANs to create a large network.

Advantages of a WAN:

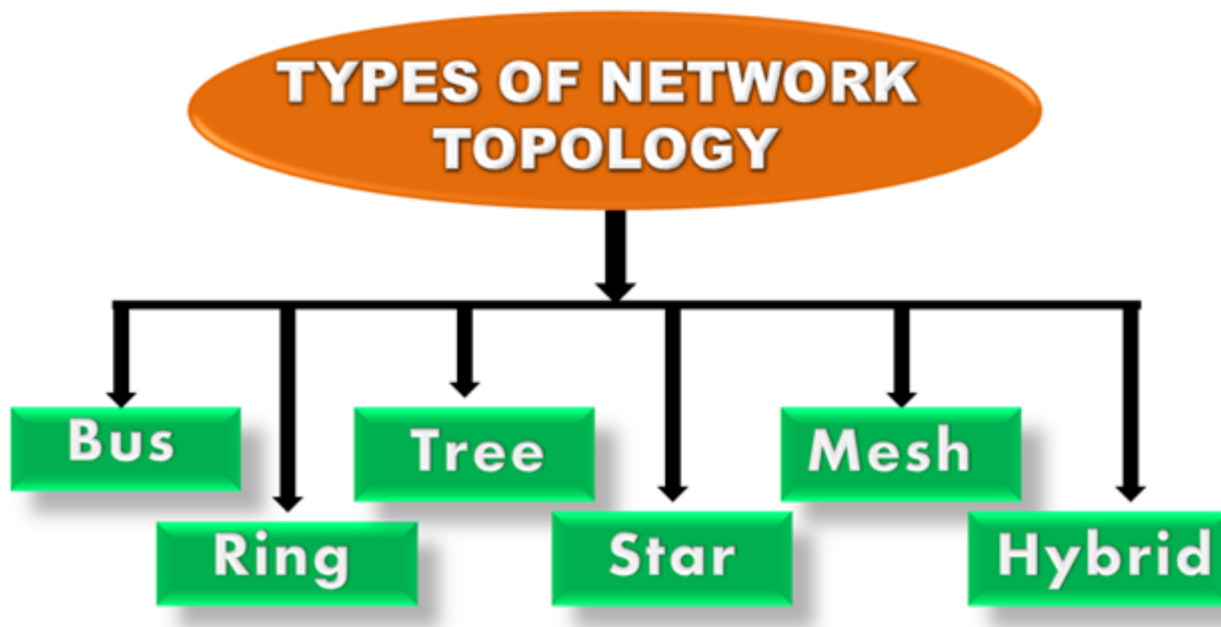
- **Large Network Range:** It spans a large geographical area of 2000 km or more, e.g., from one country to another countries.
- **Centralized data:** It allows your different office branches to use your head office server for retrieving and sharing data. Thus, you don't need to buy email servers, files server and back up servers, etc.

What is Network Topology?

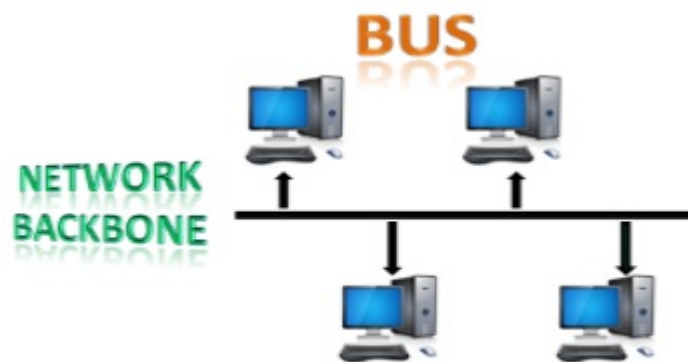
Topology defines the structure of the network of how all the components are interconnected to each other. There are two types of topology: physical and logical topology.

Types of Network Topology

Physical topology is the geometric representation of all the nodes in a network. There are six types of network topology which are Bus Topology, Ring Topology, Tree Topology, Star Topology, Mesh Topology, and Hybrid Topology.



1) Bus Topology



- The bus topology is designed in such a way that all the stations are connected through a single cable known as a backbone cable.
- Each node is either connected to the backbone cable by drop cable or directly connected to the backbone cable.

- When a node wants to send a message over the network, it puts a message over the network. All the stations available in the network will receive the message whether it has been addressed or not.
- The bus topology is mainly used in 802.3 (ethernet) and 802.4 standard networks.
- The configuration of a bus topology is quite simpler as compared to other topologies.
- The backbone cable is considered as a "**single lane**" through which the message is broadcast to all the stations.
- The most common access method of the bus topologies is **CSMA** (Carrier Sense Multiple Access).

Advantages of Bus topology:

- **Low-cost cable:** In bus topology, nodes are directly connected to the cable without passing through a hub. Therefore, the initial cost of installation is low.
- **Moderate data speeds:** Coaxial or twisted pair cables are mainly used in bus-based networks that support upto 10 Mbps.

Disadvantages of Bus topology:

- **Extensive cabling:** A bus topology is quite simpler, but still it requires a lot of cabling.
- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.

2) Ring Topology



- Ring topology is like a bus topology, but with connected ends.
- The node that receives the message from the previous computer will retransmit to the next node.

- The data flows in one direction, i.e., it is unidirectional.
- **Token passing:** It is a network access method in which token is passed from one node to another node.
- **Token:** It is a frame that circulates around the network.

Advantages of Ring topology:

- **Network Management:** Faulty devices can be removed from the network without bringing the network down.
- **Product availability:** Many hardware and software tools for network operation and monitoring are available.

Disadvantages of Ring topology:

- **Difficult troubleshooting:** It requires specialized test equipment to determine the cable faults. If any fault occurs in the cable, then it would disrupt the communication for all the nodes.
- **Failure:** The breakdown in one station leads to the failure of the overall network.

3) Star Topology



- Star topology is an arrangement of the network in which every node is connected to the central hub, switch or a central computer.
- The central computer is known as a **server**, and the peripheral devices attached to the server are known as **clients**.
- Coaxial cable or RJ-45 cables are used to connect the computers.
- Hubs or Switches are mainly used as connection devices in a **physical star topology**.
- Star topology is the most popular topology in network implementation.

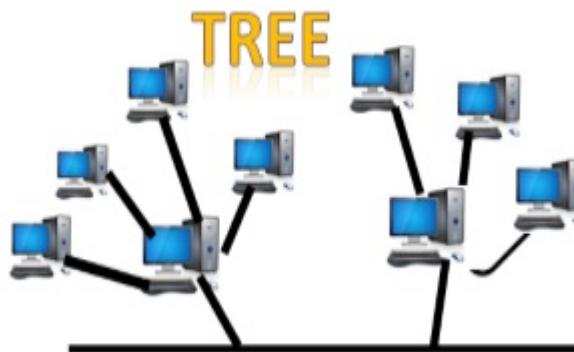
Advantages of Star topology

- **Efficient troubleshooting:** Troubleshooting is quite efficient in a star topology as compared to bus topology. In a bus topology, the manager has to inspect the kilometers of cable. In a star topology, all the stations are connected to the centralized network. Therefore, the network administrator has to go to the single station to troubleshoot the problem.
- **Network control:** Complex network control features can be easily implemented in the star topology. Any changes made in the star topology are automatically accommodated.

Disadvantages of Star topology

- **A Central point of failure:** If the central hub or switch goes down, then all the connected nodes will not be able to communicate with each other.
- **Cable:** Sometimes cable routing becomes difficult when a significant amount of routing is required.

4) Tree topology



- Tree topology combines the characteristics of bus topology and star topology.
- A tree topology is a type of structure in which all the computers are connected with each other in hierarchical fashion.
- The top-most node in tree topology is known as a root node, and all other nodes are the descendants of the root node.
- There is only one path exists between two nodes for the data transmission. Thus, it forms a parent-child hierarchy.

Advantages of Tree topology

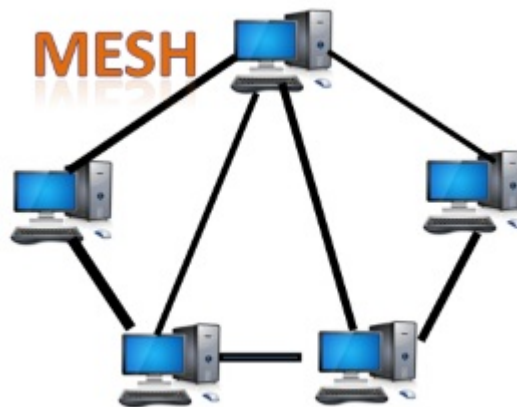
- **Support for broadband transmission:** Tree topology is mainly used to provide broadband transmission, i.e., signals are sent over long distances without being attenuated.

- **Easily expandable:** We can add the new device to the existing network. Therefore, we can say that tree topology is easily expandable.
- **Easily manageable:** In tree topology, the whole network is divided into segments known as star networks which can be easily managed and maintained.

Disadvantages of Tree topology

- **Difficult troubleshooting:** If any fault occurs in the node, then it becomes difficult to troubleshoot the problem.
- **High cost:** Devices required for broadband transmission are very costly.

5) Mesh topology

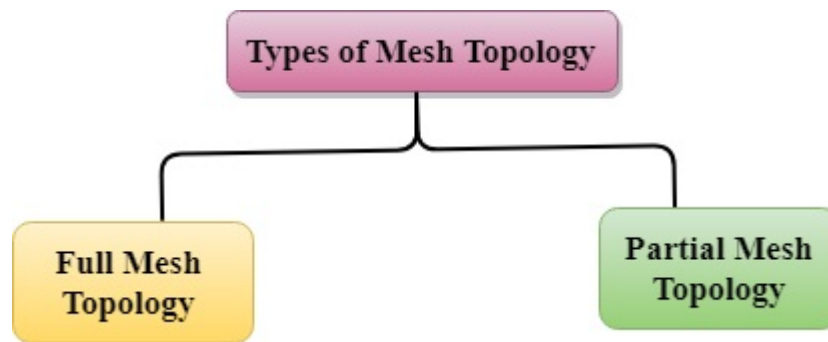


- Mesh technology is an arrangement of the network in which computers are interconnected with each other through various redundant connections.
- There are multiple paths from one computer to another computer.
- It does not contain the switch, hub or any central computer which acts as a central point of communication.
- The Internet is an example of the mesh topology.
- Mesh topology is mainly used for WAN implementations where communication failures are a critical concern.
- Mesh topology is mainly used for wireless networks.
- Mesh topology can be formed by using the formula:
Number of cables = $(n*(n-1))/2$;

Where n is the number of nodes that represents the network.

Mesh topology is divided into two categories:

- Fully connected mesh topology
- Partially connected mesh topology



- **Full Mesh Topology:** In a full mesh topology, each computer is connected to all the computers available in the network.
- **Partial Mesh Topology:** In a partial mesh topology, not all but certain computers are connected to those computers with which they communicate frequently.

Advantages of Mesh topology:

Reliable: The mesh topology networks are very reliable as if any link breakdown will not affect the communication between connected computers.

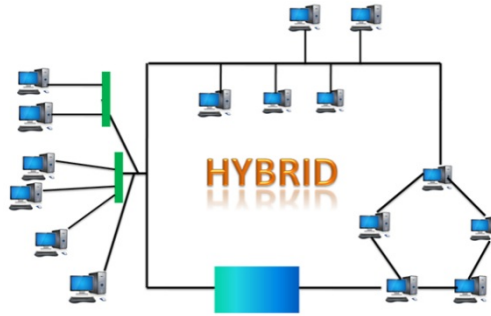
Fast Communication: Communication is very fast between the nodes.

Easier Reconfiguration: Adding new devices would not disrupt the communication between other devices.

Disadvantages of Mesh topology

- **Cost:** A mesh topology contains a large number of connected devices such as a router and more transmission media than other topologies.
- **Management:** Mesh topology networks are very large and very difficult to maintain and manage. If the network is not monitored carefully, then the communication link failure goes undetected.

6) Hybrid Topology



- The combination of various different topologies is known as **Hybrid topology**.
- A Hybrid topology is a connection between different links and nodes to transfer the data.
- When two or more different topologies are combined together is termed as Hybrid topology and if similar topologies are connected with each other will not result in Hybrid topology.

Advantages of Hybrid Topology

- **Reliable:** If a fault occurs in any part of the network will not affect the functioning of the rest of the network.
- **Scalable:** Size of the network can be easily expanded by adding new devices without affecting the functionality of the existing network.

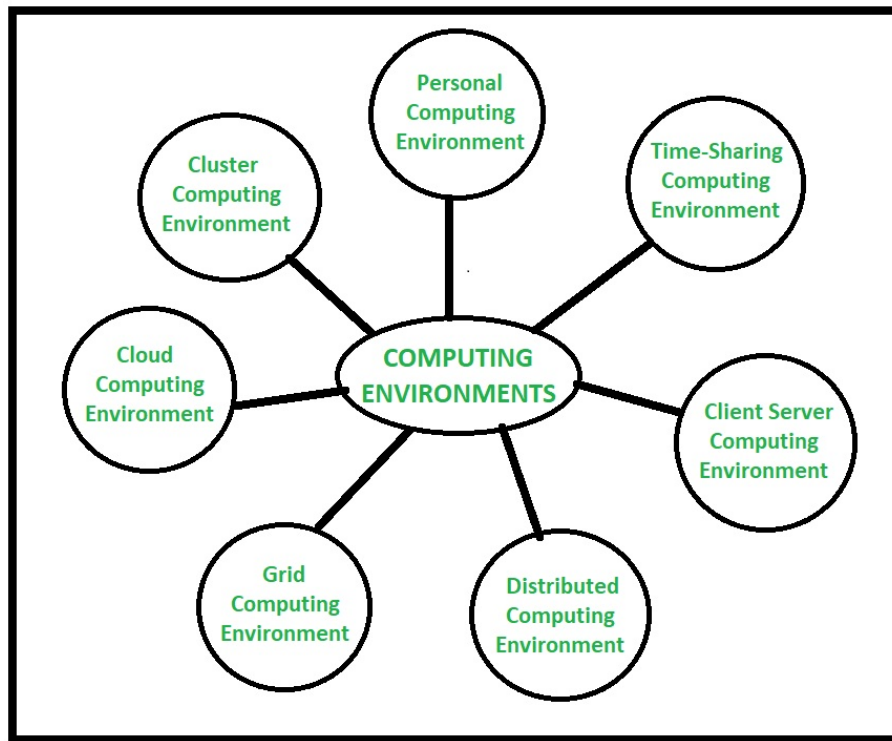
Disadvantages of Hybrid topology

- **Complex design:** The major drawback of the Hybrid topology is the design of the Hybrid network. It is very difficult to design the architecture of the Hybrid network.
- **Costly Hub:** The Hubs used in the Hybrid topology are very expensive as these hubs are different from usual Hubs used in other topologies.

Computing Environments

In the world of technology where every tasks are performed with help of computers, these computers have become one part of human life. Computing is nothing but process of completing a task by using this computer technology and it may involve computer hardware and/or software.

Types of Computing Environments :



Personal Computing Environment : In personal computing environment there is a stand-alone machine. Complete program resides on computer and executed there.

Time-Sharing Computing Environment : In Time Sharing Computing Environment multiple users share system simultaneously. Different users (different processes) are allotted different time slice and processor switches rapidly among users according to it.

Client Server Computing Environment : In client server computing environment two machines are involved i.e., client machine and server machine, sometime same machine also serve as client and server.

Distributed Computing Environment : In a distributed computing environment multiple nodes are connected together using network but physically they are separated. A single task is performed by different functional units of different nodes of distributed unit.

Grid Computing Environment : In grid computing environment, multiple computers from different locations works on single problem. In this system set of computer

nodes running in cluster jointly perform a given task by applying resources of multiple computers/nodes.

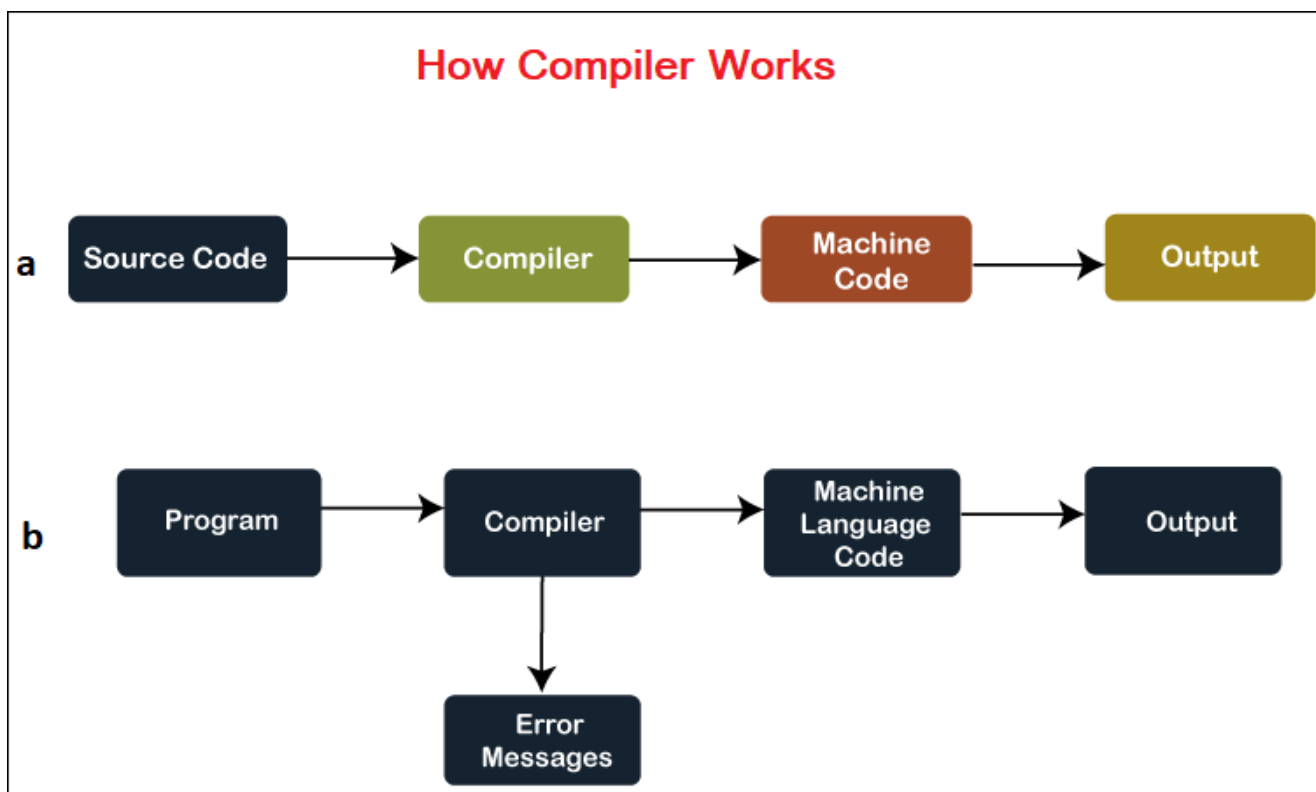
Cloud Computing Environment : In cloud computing environment on demand availability of computer system resources like processing and storage are availed. Here computing is not done in individual technology or computer rather it is computed in cloud of computers where all required resources are provided by cloud vendor.

Cluster Computing Environment : In cluster computing environment cluster performs task where cluster is a set of loosely or tightly connected computers that work together. It is viewed as single system and performs task parallelly that's why also it is similar to parallel computing environment.

Unit 7

Compiler

A compiler is a software program that follows the syntax rule of programming language to convert a source code to machine code. It cannot fix any error if present in a program; it generates an error message, and you have to correct it yourself in the program's syntax. If your written program is correct (contains no error), then the compiler will convert your entire source code into machine code. A compiler **converts complete source code** into machine code at once. And finally, your program get executes.



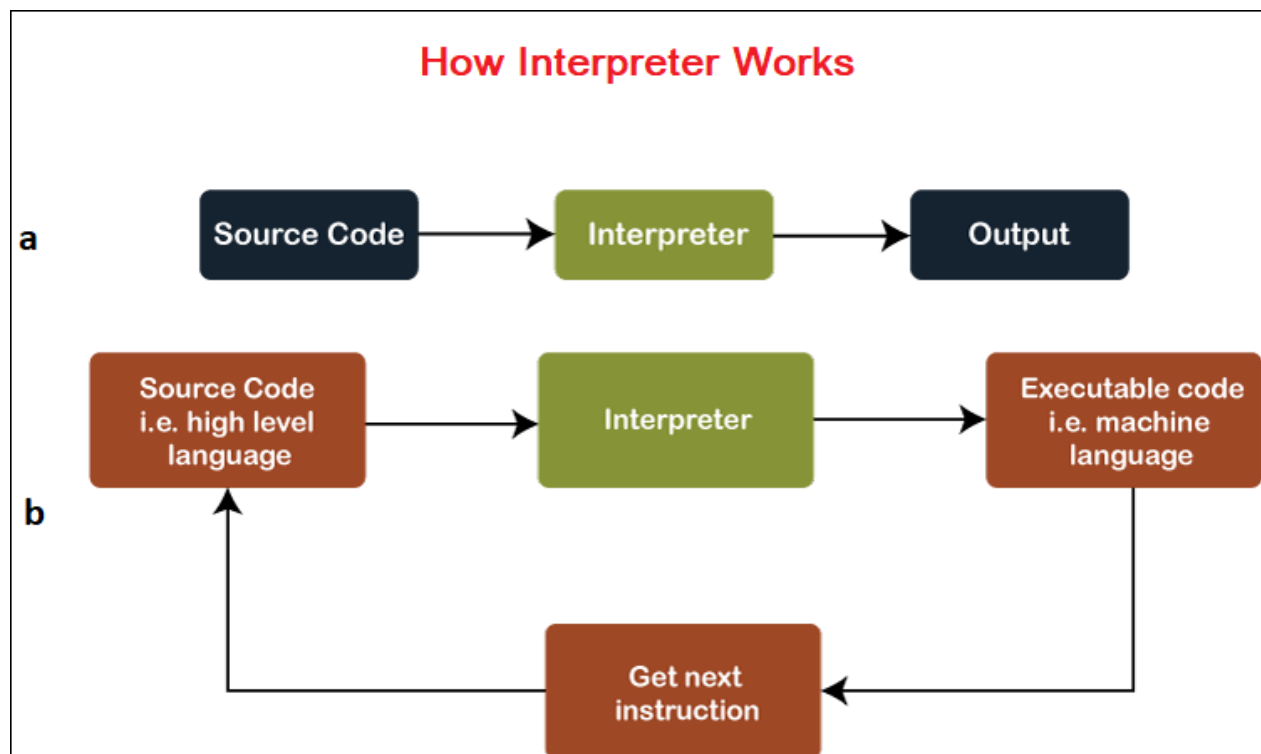
The entire compilation steps of source code are operated into two phases: **Analysis Phase** and **Synthesis Phase**.

- **Analysis Phase:** This compiler phase is also known as the front end phase in which a source code is divided into fundamental parts to check grammar, syntax, and semantic of code; after that, the *intermediate* code is generated. The analysis phase of the compilation process includes a lexical analyzer, semantic analyzer, and syntax analyzer.
- **Synthesis Phase:** The Synthesis phase is also known as the back end phase in which the *intermediate* code (which was generated in Analysis Phase) is optimized and

generated into *target* machine code. The synthesis phase of the compilation process includes code optimizer and code generator tasks.

Interpreter

An interpreter is also a software program that translates a source code into a machine language. However, an interpreter converts high-level programming language into machine language **line-by-line** while interpreting and running the program.



Difference between Compiler and Interpreter

Differences	Compiler	Interpreter

Programming Steps	○ Write a program in source code. ○ Compile will analyze your program statements and check their correctness. If an error is found in a program, it throws an error message.	○ Write a program in source code. ○ No linking of files happens, or no machine code will generate separately.	○
Translation type	A compiler translates complete high-level programming code into machine code at once.	An interpreter translates one statement of programming code at a time into machine code.	
Advantage	As the source code is already converted into machine code, the code execution time becomes short.	As the source code is interpreted line-by-line, error detection and correction become easy.	
Disadvantage	If you want to change your program for any reason, either by error or logical changes, you can do it only by going back to your source code.	Interpreted programs can run on only those computers which have the same interpreter.	
Machine code	It stores the converted machine code from your source code program on the disk.	It never stores the machine code at all on the disk.	
Running time	A compiler takes an enormous time to analyze source code. However, overall compiled programming code runs faster as compared to an interpreter.	An interpreter takes less time to analyze source code as compared to a compiler. However, overall interpreted programming code runs slower as compared to the compiler.	
Program generation	The compiler generates an output of a program (in the form of an exe file) that can run separately from the source code program.	The interpreter doesn't generate a separate machine code as an output program. So it checks the source code every time during the execution.	
Execution	The process of program execution takes place separately from its compilation process. Program execution only takes place after the complete program is compiled.	The process of program execution is a part of interpretation steps, so it is done line-by-line simultaneously.	

Memory requirement	A compiled program is generated into an intermediate object code, and it further required linking. So there is a requirement for more memory.	An interpreted program does not generate an intermediate code. So there is no requirement for extra memory.	
Best suited for	The compiled program is bounded to the specific target machine. It requires the same compiler on the machine to execute; C and C++ are the most popular programming language based on the compilation model.	In web environments, compiling takes place relatively more time to run even small code, which may not run multiple times. As load time is essential in the web environment, interpreters are better. JavaScript, Python, Ruby are based on the interpreter model.	
Error execution	The compiler shows the complete errors and warning messages at program compilation time. So it is not possible to run the program without fixing program errors. Doing debugging of the program is comparatively complex while working with a compiler.	An interpreter reads the program line-by-line; it shows the error if present at that specific line. You must have to correct the error first to interpret the next line of the program. Debugging is comparatively easy while working with an Interpreter.	

Advantage and disadvantage of compiler

While using a compiler to translate a source code into machine code, the program codes are translated into different object codes. Hence the time of code execution is significantly less. The drawback of using a compiler is that you can only make changes in the program by going back to your source code.

Advantage and disadvantage of an interpreter

The interpreter makes it easier to work with source code. So it is highly preferred, especially for beginners. Interpreted programs can run on only those computers which have the same interpreter.

Assembler

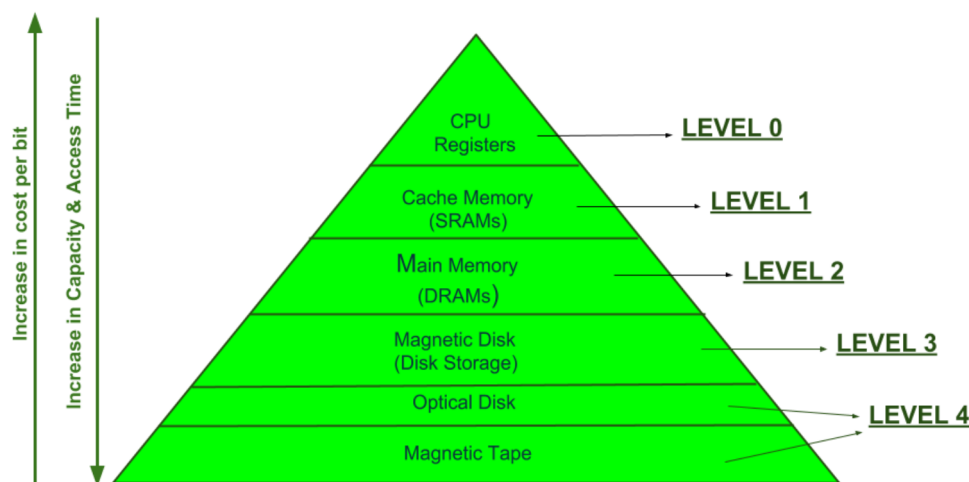
The Assembler is a Software that converts an assembly language code to machine code. It takes basic Computer commands and converts them into Binary Code that Computer's Processor can use to perform its Basic Operations. These instructions are assembler language or assembly language.

What Does Mail Merge Mean?

Mail merge is a feature within most data processing applications that enables users to send a similar letter or document to multiple recipients. It enables connecting a single form template with a data source that contains information about the recipient's name, address and other predefined and supported data.

Memory Hierarchy Design and its Characteristics

In the Computer System Design, Memory Hierarchy is an enhancement to organize the memory such that it can minimize the access time. The Memory Hierarchy was developed based on a program behavior known as locality of references. The figure below clearly demonstrates the different levels of memory hierarchy :



MEMORY HIERARCHY DESIGN

This Memory Hierarchy Design is divided into 2 main types:

1. External Memory or Secondary Memory —

Comprising of Magnetic Disk, Optical Disk, Magnetic Tape i.e. peripheral storage devices which are accessible by the processor via I/O Module.

2. Internal Memory or Primary Memory —

Comprising of Main Memory, Cache Memory & CPU registers. This is directly accessible by the processor.

We can infer the following characteristics of Memory Hierarchy Design from above figure:

1. Capacity:

It is the global volume of information the memory can store. As we move from top to bottom in the Hierarchy, the capacity increases.

2. Access Time:

It is the time interval between the read/write request and the availability of the data. As we move from top to bottom in the Hierarchy, the access time increases.

3. Performance:

Earlier when the computer system was designed without Memory Hierarchy design, the speed gap increases between the CPU registers and Main Memory due to large difference in access time

4. Cost per bit:

As we move from bottom to top in the Hierarchy, the cost per bit increases i.e. Internal Memory is costlier than External Memory.

What Does IO.SYS Mean?

IO.SYS is a hidden executable binary file or hidden system file which processes instructions when the operating system is booted. It was an essential part of MS-DOS and Windows 9x systems. The instructions tell the operating system how the computer is set up. Together with the MSDOS.SYS system file, they made up Microsoft's MS-DOS and were loaded into the memory of the computer.

What is MSDOS.SYS?

The MSDOS.SYS file is a hidden, system, read-only file created on the root of the boot drive. There are several configurations that can be changed using this file. To edit the file, first from C:\> Type **attrib -h -s -r MSDOS.SYS**

As always, make a backup copy in case you mistakenly cause an issue when trying to edit this file. To edit this file, type **edit msdos.sys**.

Most values in the MSDOS.SYS are either 0 or 1, which is off or on.

What Does COMMAND.COM Mean?

COMMAND.COM is the default shell for Microsoft operating systems, including MS-DOS and Windows versions up through Windows ME. Other versions of DOS not from Microsoft also have a command shell named COMMAND.COM, including DR DOS and FreeDOS. It allows users to execute commands and run scripts known as batch files. It has been superseded in Windows NT and by extension all modern versions of Windows by CMD.EXE and PowerShell.

What Does CONFIG.SYS Mean?

CONFIG.SYS is a configuration file on DOS systems. It is a text file that contains the settings and commands to load drives in a DOS system. This is a primary configuration file for OS/2 and DOS OS. This file was introduced in DOS and is replaced for 32-bit Windows versions with CONFIG.NT.