

**RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL**

**New Scheme Based On AICTE Flexible Curricula**

**B. Tech. First Year**

**Branch- Common to All Disciplines**

<b>BT205</b>	<b>Basic Computer Engineering</b>	<b>3L-0T-2P</b>	<b>4 Credits</b>
--------------	-----------------------------------	-----------------	------------------

**Course Contents:**

**UNIT I**

**Computer:** Definition, Classification, Organization i.e. CPU, register, Bus architecture, Instruction set, Memory & Storage Systems, I/O Devices, and System & Application Software. Computer Application in e-Business, Bio-Informatics, health Care, Remote Sensing & GIS, Meteorology and Climatology, Computer Gaming, Multimedia and Animation etc.

**Operating System:** Definition, Function, Types, Management of File, Process & Memory. Introduction to MS word, MS powerpoint, MS Excel

**UNIT II**

Introduction to Algorithms, Complexities and Flowchart, Introduction to Programming, Categories of Programming Languages, Program Design, Programming Paradigms, Characteristics or Concepts of OOP, Procedure Oriented Programming VS object oriented Programming. Introduction to C++: Character Set, Tokens, Precedence and Associativity, Program Structure, Data Types, Variables, Operators, Expressions, Statements and control structures, I/O operations, Array, Functions,

**UNIT III**

Object & Classes, Scope Resolution Operator, Constructors & Destructors, Friend Functions, Inheritance, Polymorphism, Overloading Functions & Operators, Types of Inheritance, Virtual functions. Introduction to Data Structures.

**UNIT IV**

**Computer Networking:** Introduction, Goals, ISO-OSI Model, Functions of Different Layers. Internetworking Concepts, Devices, TCP/IP Model. Introduction to Internet, World Wide Web, E-commerce

**Computer Security Basics:** Introduction to viruses, worms, malware, Trojans, Spyware and Anti-Spyware Software, Different types of attacks like Money Laundering, Information Theft, Cyber Pornography, Email spoofing, Denial of Service (DoS), Cyber Stalking, Logic bombs, Hacking Spamming, Cyber Defamation , pharming Security measures Firewall, Computer Ethics & Good Practices, Introduction of Cyber Laws about Internet Fraud, Good Computer Security Habits,

**UNIT V**

**Data base Management System:** Introduction, File oriented approach and Database approach, Data Models, Architecture of Database System, Data independence, Data dictionary, DBA, Primary Key, Data definition language and Manipulation Languages.

**Cloud computing:** definition, cloud infrastructure, cloud segments or service delivery models (IaaS, PaaS and SaaS), cloud deployment models/ types of cloud (public, private, community and hybrid clouds), Pros and Cons of cloud computing

**List of Experiment**

01. Study and practice of Internal & External DOS commands.
02. Study and practice of Basic linux Commands – ls, cp, mv, rm, chmod, kill, ps etc.
03. Study and Practice of MS windows – Folder related operations, My-Computer, window explorer, Control Panel,
04. Creation and editing of Text files using MS- word.
05. Creation and operating of spreadsheet using MS-Excel.
06. Creation and editing power-point slides using MS- power point
07. Creation and manipulation of database table using SQL in MS-Access.
08. WAP to illustrate Arithmetic expressions
09. WAP to illustrate Arrays.
10. WAP to illustrate functions.
11. WAP to illustrate constructor & Destructor
12. WAP to illustrate Object and classes.

13. WAP to illustrate Operator overloading
14. WAP to illustrate Function overloading
15. WAP to illustrate Derived classes & Inheritance
16. WAP to insert and delete and element from the Stack
17. WAP to insert and delete and element from the Queue
18. WAP to insert and delete and element from the Linked List

**Recommended Text Books:**

1. Fundamentals of Computers : E Balagurusamy, TMH
2. Basic Computer Engineering: Silakari and Shukla, Wiley India
3. Fundamentals of Computers : V Rajaraman, PHI
4. Information Technology Principles and Application: Ajoy Kumar Ray & Tinku Acharya PHI.

**Recommended Reference Books:**

1. Introduction of Computers : Peter Norton, TMH
2. Object Oriented Programming with C++ :E.Balagurusamy, TMH
3. Object Oriented Programming in C++: Rajesh K.Shukla, Wiley India
4. Concepts in Computing: Kenneth Hoganson, Jones & Bartlett.
5. Operating Systems – Silberschatz and Galvin - Wiley India
6. Computer Networks:Andrew Tananbaum, PHI
7. Data Base Management Systems, Korth, TMH
8. Cloud Computing, Kumar, Wiley India

## NOTES

## UNIT-1

**What is Computer**

The term computer has taken from the computing. A computer can be defined as an electronic device that takes data and instruction from the user as input perform computing and give information as output to the user. Fig.no 1.1.

The electronic device is known as hardware & the set of instruction is known as software.

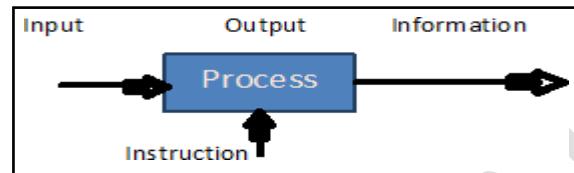


Fig.no 1.1

**Basic Computer Operation**

1. **INPUT:** It accepts data and instruction through the input device.
2. **PROCESS AND CONTROL:** it performs the action as per the instruction issued and process the given data.
3. **STORAGE:** it stores the data and instruction for future execution.
4. **OUTPUT:** it generates the desired output after executing the instruction and processing the input data.

**Advantages of Computer**

1. **ACCURACY** Computer performs complex and repetitive calculation with accurate results.
2. **MEMORY** Store large amount of data and information
3. **USER-FRIENDLY** Provide information to the user in many different forms.
4. **FAST** Perform fast execution or processing.
5. **LESS MAN POWER** Manual requirement is less.
6. **BEST RESULTS** Artificial intelligence, Decision-making best results.

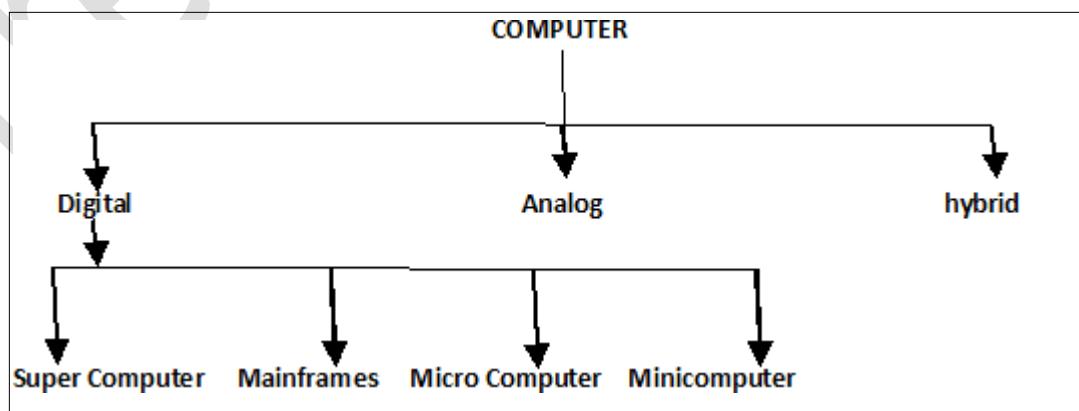
**CLASSIFICATION OF COMPUTER**

Fig.no 1.2

**Analog Computer:** As figure 1.2 an **analog computer** is a form of computer that uses the continuously changeable aspects of physical phenomena such as electrical, mechanical, or hydraulic quantities to model the problem being solved.



Fig.no 1.3



Fig.no 1.4

### Digital Computer:

The digital computer works on discontinuous data. They convert the data into digits (binary digits 0 and 1) and all operations are carried out on these digits at extremely fast rates.

A digital computer basically knows how to count the digits and add the digits. Digital computers are much faster than an analog computer and far more accurate.

**Supercomputer:** A supercomputer contains many CPUs which operate in parallel to make it faster. They are used for massive data processing and solving very sophisticated problems. They are used for weather forecasting, weapons research and development, rocketing, aerodynamics, seismology, atomic, nuclear etc.

**MAINFRAME COMPUTERS:** Mainframe computers are very powerful, large general-purpose computers. They are used where a large amount of data is to be processed or very complex calculations are to be made and these tasks are beyond the computing capacity of minicomputers. They are used in research organizations, large industries, large business, and government organizations, bank, and airline



Fig.No 1.5



Fig.no 1.6



Fig.no 1.7

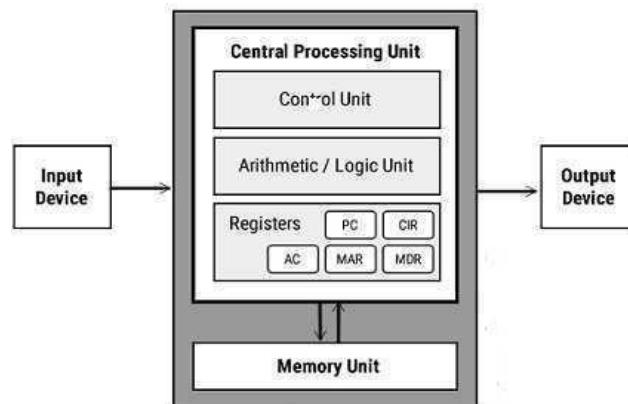
reservations where a large database is required as fig 1.7.

**Microcomputers:** A Microcomputer is a low-cost, small, digital computer. It contains the microprocessor as its CPU, a memory unit, an input device, and an output device. Microcomputers have a wide range of applications like general purpose calculations, industrial control, home application; Microcomputers are also called personal computers as fig 1.6.

**Minicomputers:** Minicomputers are faster and more powerful than microcomputers. Minicomputers are general purpose computers, smaller than mainframe and give computing power without adding the prohibitive expenses associated with large systems. They are used in accounting, word processing, database management, statistical packages for social sciences, CAD, and numerical analysis etc as Fig 1.7.

## Central Processing Unit

The CPU which is referred to as the brain of a computer is responsible for processing the data inside the computer system. It is also responsible for controlling all other components of computer system.



**Block Diagram of a Computer System Fig.no 1.8**

Central processing unit (CPU) is the central component of the PC. Sometimes it is called as a processor. It is the brain that runs the show inside the PC. All work that is done on a computer is performed directly or indirectly by the processor. Obviously, it is one of the most important components of the PC. It is also, scientifically, not only one of the most amazing parts of the PC but one of the most amazing devices in the world of technology. The processor plays a significant role in the following important aspects of your computer system;

**Performance:** The processor is probably the most important single determinant of system performance on the PC. While other components also play a key role in determining performance, the processor's capabilities dictate the maximum performance of a system. The other devices only allow the processor to reach its full potential.

**Software Support:** Newer, faster processors enable the use of the latest software. In addition, new processors such as the i3/i5/i7 with MMX Technology, enable the use of specialized software not usable on earlier machines.

**Reliability and Stability:** The quality of the processor is one factor that determines how reliably your system will run. While most processors are very dependable, some are not. This also depends to some extent on the age of the processor and how much energy it consumes. The CPU consists of Control Unit, Arithmetic and Logic Unit (ALU) and register set.

### Main Operation of the CPU Includes

1. **FETCH:** Fetching instruction from the memory issued by the user.
2. **DECODE:** Decoding the instruction to decide what operation to be performed.
3. **EXECUTE:** Execute the instruction.
4. **STORE:** Store the result in the memory.

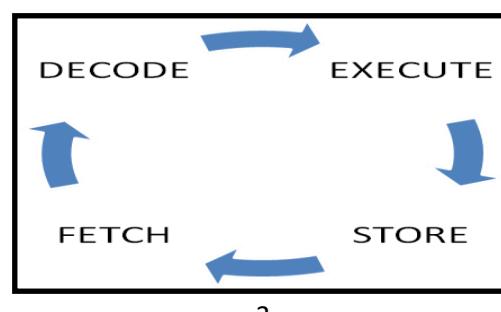


Fig.no 1.9

### The structure of CPU:

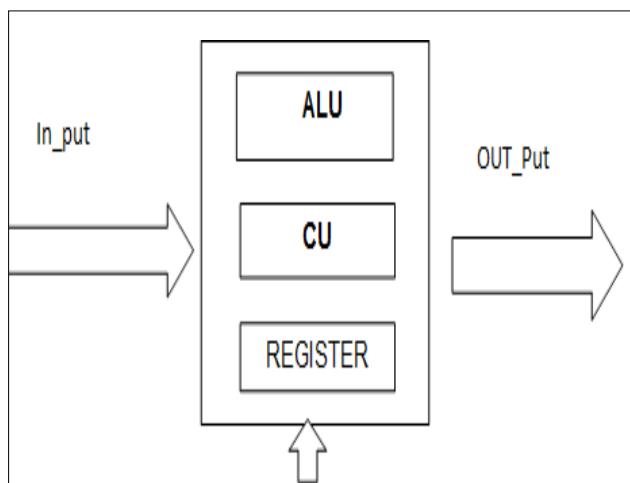


Fig.no 1.10

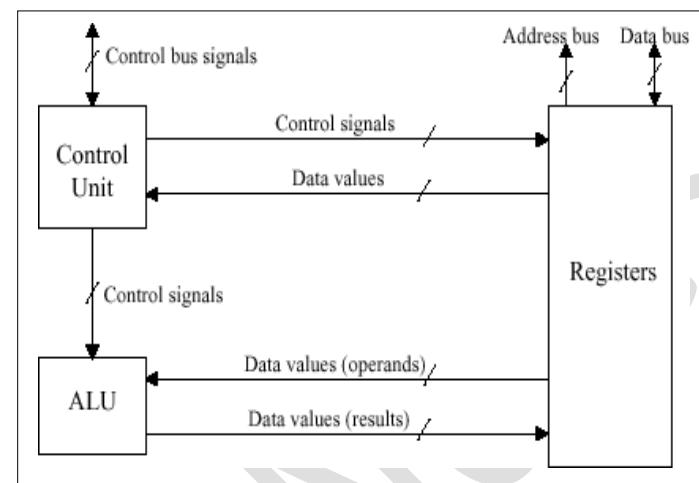


Fig.no 1.11

### Arithmetic Logical Unit (ALU)

ALU is the combination of arithmetic unit and logical unit is used to perform an arithmetic operation on the input data (+, -, \*, /). Logical unit is used to perform logical operation on input data (<, >, <=, >=, =, OR, NOT, AND)

### Control Unit (CU)

CU is an important component that controls the flow of data and information. It is important for the proper execution of the instruction.

### Cache Memory

Cache memory is a small fast and expensive memory that stores the copies of data that needs to be accessed frequently from main memory.

### Types of Cache memory

**Primary Cache:** - It is also known as L1 cache or internal cache it is located inside the CPU and provides quick access to the frequently provided data to the microprocessor.

**Secondary Cache:** - - It is also known as L2 cache or External cache it is located outside the CPU and positioned on the motherboard. It is slower as compared to L1.

### Register (Memory Unit)

**The register** is used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU, there are various types of Registers those are used for various purposes. Register are special purpose high-speed temporary storage area for holding data, address, and instruction during processing of the instruction. The register is always on the CPU.

**PC:** - The **program counter (PC)** just part of the instruction sequencer in some computers is a processor register. It keeps track of the **next memory address** of the instruction that is to be executed once the execution of the current instruction is completed. **In other words, it holds the address of the memory location of the next instruction when the current instruction is executed.**

**AC: -Accumulator:** This Register is used for storing the Results those are produced by the System. When the CPU will generate Some Results after the Processing then all the Results will be Stored into the **AC Register.**

**IR:** - Instruction Register: store the instruction currently being executed.

**MAR:** - (Memory address register) this register holds the memory addresses of data and instructions. This register is used to access data and instructions from memory during the execution phase of an instruction. Suppose CPU wants to store some data in the memory or to read the data from the memory. It places the address of the required memory location in the MAR.

**MBR:** - **Memory buffer register:** -MBR stands for Memory Buffer Register. **This register holds the contents of data or instruction read from, or written in the memory. It means that this register is used to store data/instruction coming from the memory or going to the memory.**

**MDR:** - **(Memory Data register)** MDR is the register of a computer's control unit that contains the **data to be stored in the computer storage** (e.g. RAM), or the **data after a fetch from the computer storage**. It acts as a **buffer** and holds anything that is copied from the memory ready for the processor to use it.

### Internal Communication

- Processor to memory communication
- Processor to I/O device communication

### Bus architecture: -

A bus is a set of wire that is used to connect the different internal components of a computer system for transferring data, address, and control.

**There may be several buses in a computer system broadly divided into categories.**

- Serial Bus
- Parallel Bus

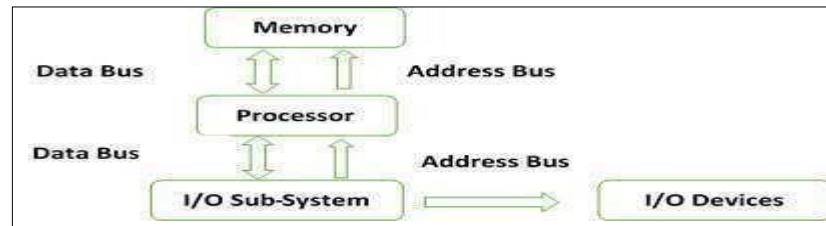
The speed of any type of bus is measured in terms of the number of bits transferred per second, between two components.

In Serial Bus only one bit of data is transferred at a time, amongst the various hardware components.

In Parallel Bus, several bits of data can be transferred at a time, amongst the various hardware components.

### Based on the Type of data it carries

**e Control Bus** manages the transfer of data and address among various components by transferring appropriate control signals.

**Fig.no 1.12**

### # DATA BUS: -

The Databus in a computer system is used to transfer data amongst the different internal components. The speed of the data bus also affects the overall processing power of a computer system. The modern computer system uses 32-bit data buses for data transfer. This means that these buses can transfer 32 bits of data at a time.

The data bus implemented between the main memory and the processor of a computer system.

The above figure shows that a bidirectional data bus is implemented between the main memory and processor of the computer system. The bidirectional data bus allows the transfer of data in both the directions. The data bus is generally bi-directional in nature in the most computer system.

**# ADDRESS BUS:** -The address bus is also known as the memory bus. It transfers the memory address for reading and writes memory operations. It contains many address lines that determine the range of memory addresses that can be referenced using the address bus.

For Example-a 32-bit address bus can be used to reference 2 Memory locations. Like data bus, the address bus can also be a serial or a parallel bus.

### Instruction Set

The instruction set, also called instruction set architecture (ISA), is part of a computer that pertains to programming, which is basically machine language. The instruction set provides commands to the processor, to tell it what it needs to do. The instruction set consists of addressing modes, instructions, native data types, registers, memory architecture, interrupt, and exception handling, and external I/O.

An example of an instruction set is the RISC, CISC, x86 instruction set, which is common to find on computers today. Different computer processors can use almost the same instruction set while still having very different internal design. Both the Intel Pentium and AMD Athlon processors use nearly the same x86 instruction set.

#### Examples of instruction set

- ADD - Add two numbers together.
- COMPARE - Compare numbers.
- IN - Input information from a device, e.g. a keyboard.
- JUMP - Jump to designated RAM address.
- JUMP IF - a Conditional statement that jumps to a designated RAM address.
- LOAD - Load information from RAM to the CPU.
- OUT - Output information to device, e.g. monitor.
- STORE - Store information to RAM.

### Types

## CISC Architecture

The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction. Computers based on the CISC architecture are designed to decrease the memory cost. Because, the large programs need more storage, thus increasing the memory cost and large memory becomes more expensive. To solve these problems, the number of instructions per program can be reduced by embedding the number of operations in a single instruction, thereby making the instructions more complex.

### Characteristics of CISC architecture

- Instruction-decoding logic will be Complex.
- One instruction is required to support multiple addressing modes.
- Less chip space is enough for general purpose registers for the instructions that are Operated directly on memory.
- Various CISC designs are set up two special registers for the stack pointer, handling interrupts, etc.
- MUL is referred to as a “complex instruction” and requires the programmer for storing functions.

## RISC Architecture

RISC (Reduced Instruction Set Computer) is used in portable devices due to its power efficiency. For Example, Apple iPod and Nintendo DS. RISC is a type of microprocessor architecture that uses a highly-optimized set of instructions. RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program. Pipelining is one of the unique features of RISC. It is performed by overlapping the execution of several instructions in a pipeline fashion. It has a high-performance advantage over CISC.

### RISC architecture characteristics

- Simple Instructions are used in RISC architecture.
- RISC helps and supports few simple data types and synthesize complex data types.
- RISC utilizes simple addressing modes and fixed length instructions for pipelining.
- RISC permits any register to use in any context.
- One Cycle Execution Time
- The amount of work that a computer can perform is reduced by separating “LOAD” and “STORE” instructions.
- RISC contains Large Number of Registers to prevent a various number of interactions with memory.
- In RISC, more RAM is required to store assembly level instructions.

## MEMORY AND STORAGE DEVICES: -

The memory unit of a computer is used to store data, instructions for processing data, intermediate results of processing and the final processed information. The memory unit of a computer is classified as primary and secondary memory.

## Memory

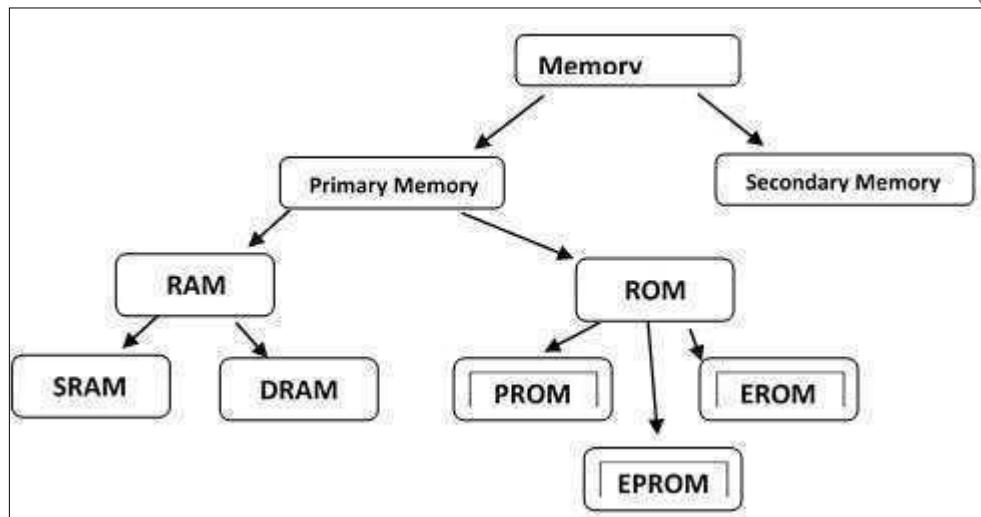


Fig.no 1.13

#### **PRIMARY MEMORY: -**

The primary memory is available on the computer as a built-in unit of the computer. The primary memory is represented as a set of location occupying 8 bits. Each bit in the memory is identified by a unique address. The data is stored in the machine understandable binary form in these memory locations.

#### **THE COMMONLY USED PRIMARY MEMORY ARE**

**Random Access Memory:** - This is the primary memory from where data & instructions can be received in a random manner (RAM). It is a volatile memory in which the contents are lost once the power is turned off. The kind of memory is used to store the data temporarily during the computer operations.

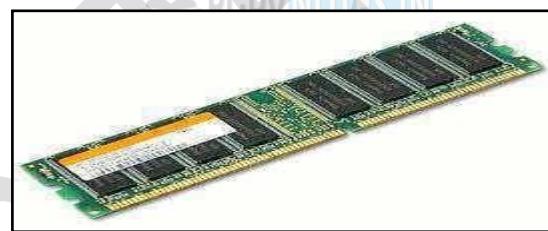


Fig.no 1.14

1. **Static random-access memory (SRAM):** It is a volatile memory based on traditional transistors using flip-flop gates to hold data if the power is on. The contents are lost once the power is turned off. It is very fast and that's why used in the cache memory. The SRAM takes more space and is expensive too, but it is easy to use. It does not need to be refreshed periodically and synchronizes itself with the timing of CPU.

2. **Dynamic random-access memory (DRAM):** It is a volatile memory based on the capacitors that hold data if the power is on. Due to discharging capacitors, the DRAM is refreshed periodically. This refreshing is done automatically, and due to the time consumed in refreshing. The DRAM is slow. It is inexpensive and takes less space; therefore, DRAM is used as the main memory.

**Read Only Memory:** ROM is a special type of memory which can only be read and contents of which are not lost even when the computer is switched off. ROM chips are used not only in computers but in most other electronic items such as washing machines, microwave ovens, calculators, laser printers, media players etc. It is not limited to electronic chips, CDROM and DVD ROM. It is categorized as follows:



Fig.no 1.15

1. **Programmable read-only memory (PROM):** This is a kind of ROM has not been pre-recorded by the manufacturer but is supplied empty. The user of this ROM can store programs on it using a special tool. Once the empty ROM is programmed, it behaves like any other ROM, that is, it cannot be rewritten.
2. **Erasable programmable read-only memory (EPROM):** This is a ROM which has not been pre-recorded by the manufacturer, but it is supplied empty. The user of this ROM can store programs on it using a special tool. Once the empty ROM is programmed, it can be rewritten repeatedly after erasing the previously written entire contents using the ultraviolet light of specific frequency. EPROM is more expensive than PROM.
3. **Electrically erasable programmable read-only memory (EEPROM):** This is a ROM which has not been pre-recorded by the manufacturer, but it supplied empty. The user of this ROM can store programs on it using a special tool. Once the empty ROM is programmed, it can be rewritten repeatedly after erasing the previously written contents using electric charge. This kind of ROM requires erasing the entire previously written content, but it allows erasing one byte at a time before writing the new content onto it. The EEPROM is more expensive than PROM, but it need not be removed from the computer for rewriting. This is the most flexible type of ROM, which is now commonly used for storing BIOS programs.

#### **SECONDARY MEMORY: -**

The secondary memory is the storage devices in which the data can be stored for a longer duration, and it is not lost even when the power is turned off. The hard disks, flash drive, floppy disks, CD-ROMs, and DVDs, etc., are examples of secondary memory. This memory has greater storage capacity than the primary memory. Also, it is inexpensive but slow.

The secondary storage device can be classified as:

**Magnetic Storage device:** - The magnetic storage devices store information that can be read, erased and rewritten many times. These include a floppy disk, hard disk, and magnetic tapes.



Fig.no 1.16

**Optical storage device:** - The optical storage devices are secondary storage devices that use laser beams to read the stored data. These include CD-ROM, rewritable compact disk (CD-RW). Digital video disks with read-only memory, etc.



Fig.no 1.17

**Magneto-optical storage device:** - The magneto-optical devices are generally used to store information, such as large programs, files and backup data. The end user can modify the information stored in magneto-optical devices multiple times. These devices provide higher storage capacity as they use laser beams and magnets for reading and writing data to the device.

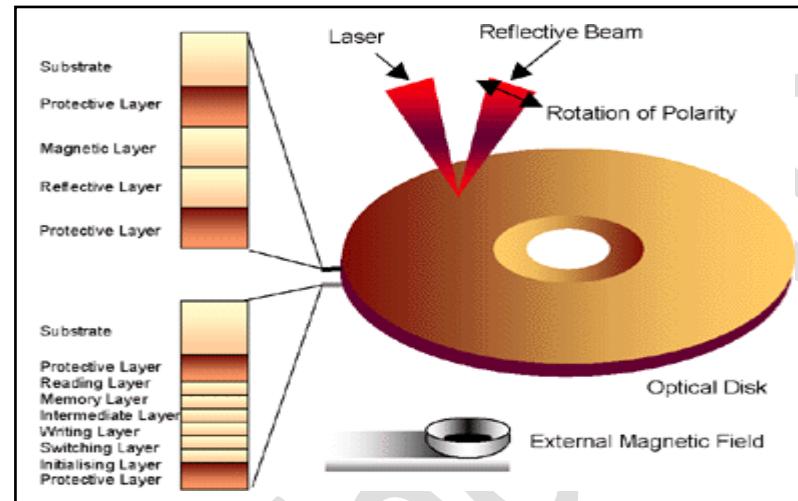


Fig.no 1.18

#### CACHE MEMORY: -

The cache memory is faster than the CPU registers and slower than the main or primary memory. It is a SRAM placed between the CPU and the main memory, and when CPU needs any data or instruction is found in the cache memory, it is retrieved by the CPU for processing. Otherwise, the main memory is searched for the same information. The most frequently used instructions and data are placed in the cache memory; therefore, the overall speed of the computer is increased.

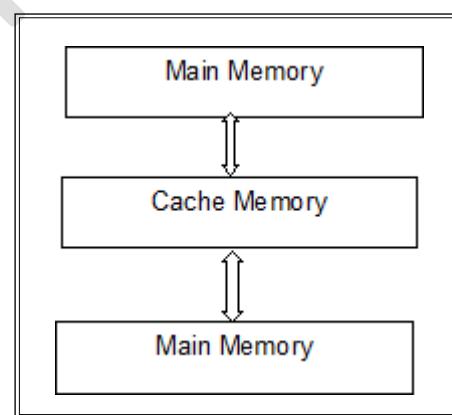


Fig.no 1.19

**MEMORY HIERARCHY:** -The computer requires different kinds of memory for its proper functioning. Since the fast memories are very expensive, therefore they cannot be used in excess. There is a hierarchy of

memories considering their speed and cost. The registers are the fastest storage devices; even faster than the cache memory (SRAM), which is faster than the main memory.

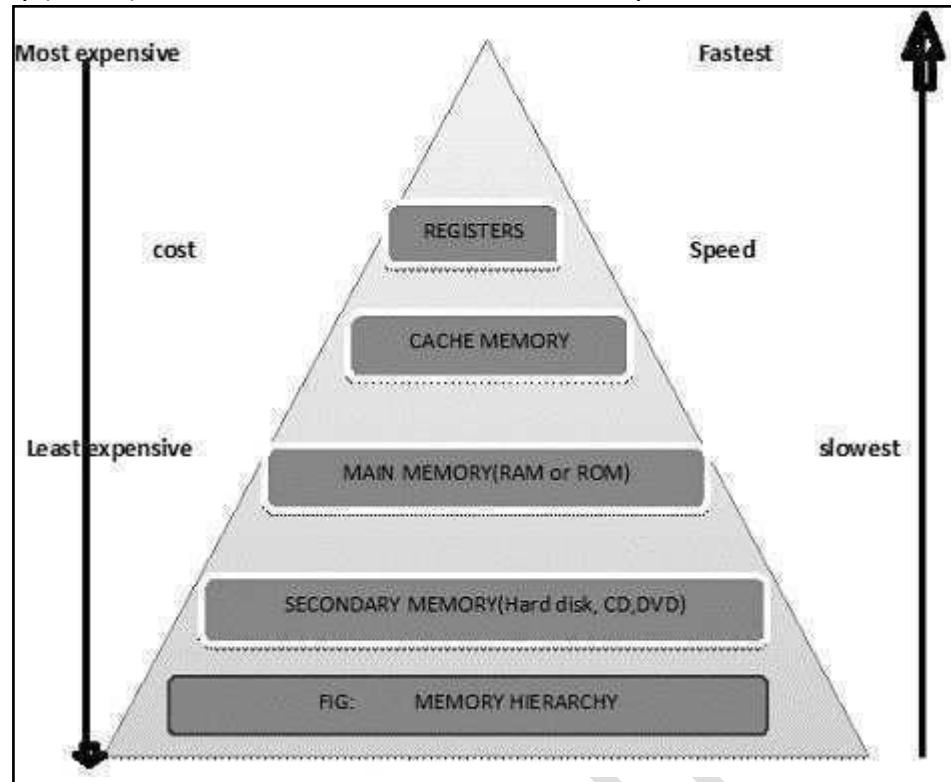


Fig.no 1.20

### Fundamental Components of Computer

**INPUT UNIT:** Input unit is formed by the input devices attached to the computer. Input devices take the raw data from the user to the computer for processing.

- **KEYBOARD:** -
- In computing, a **keyboard** is a typewriter-style device, which uses an arrangement of buttons or keys, to act as mechanical levers or electronic switches. Ref. Fig 1.21



Fig.no 1.21



Fig.no 1.22



Fig.no.1.23



Fig.no.1.24

- **MOUSE:** -
- In computing, a **mouse** is a pointing device that functions by detecting two-dimensional motion relative to its supporting surface. Physically, a mouse consists of an object held under one of the user's hands, with one or more buttons. Ref. Fig 1.22

- **LIGHT PEN:** -
- A **light pen** is a computer input device in the form of a light-sensitive wand used in conjunction with a computer's CRT display. Ref. Fig 1.23

- **TOUCH SCREEN:** -

A **touchscreen** is an electronic visual display that the user can control through simple or multi-touch gestures by touching the screen with one or more fingers. Some touchscreens can also detect objects such as a stylus or ordinary or specially coated gloves. Ref. Fig 1.24

➤ **JOYSTICK:** -

A **joystick** is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling. A joystick, also known as the **control column**, is the principal control device in the cockpit of many civilian and military aircraft, either as a center stick or side-stick. It often has supplementary switches to control various aspects of the aircraft's flight.



Fig.no.1.25



Fig.no.1.26



Fig.no.1.27



Fig.no.1.38

**MICROPHONE:** -

A microphone is an example of a transducer, a device that changes information from one form to another. Sound information exists as patterns of air pressure; the microphone changes this information into patterns of electric current. The recording engineer is interested in the accuracy of this transformation, a concept he thinks of as fidelity.

➤ **MAGNETIC INK CHARACTER READER:** -

**Magnetic ink character recognition**, or **MICR**, is a character recognition technology used primarily by the banking industry to facilitate the processing and clearance of cheques and other documents. The MICR encoding, called the MICR line, is located at the bottom of a cheque or other voucher and typically includes the document type indicator, bank code, bank account number, cheque number and the amount, plus some control indicator.

➤ **SCANNER:** -

In computing, an **image scanner**—often abbreviated to just **scanner**—is a device that optically scans images, printed text, handwriting, or an object, and converts it to a digital image.



Optical Mark Reader Fig.no.1.29



Barcode Reader Fig.no1.30



Smart C-R Fig.no.1.31



Web Cam Fig.no.1.32

➤ **OPTICAL MARK READER:** -

**Optical mark recognition** (also called **optical mark reading** and **OMR**) is the process of capturing human-marked data from document forms such as surveys and tests.

➤ **SMART CARD READER:** -

A **card reader** is a data input device that reads data from a card-shaped storage medium. The first were punched card readers, which read the paper or cardboard punched cards that were used during the first several decades of the computer industry to store information and programs for computer systems.

➤ **BARCODE READER**

A **barcode reader** (or **barcode scanner**) is an electronic device for reading printed barcodes. Like a flatbed scanner, it consists of a light source, a lens and a light sensor translating optical impulses into electrical ones.

➤ **WEBCAM:** -

A **webcam** is a video camera that feeds its image in real time to a computer or computer network.

**OUTPUT UNIT:** The output generated by the computer is sent to the output device.



Monitor Fig.no.1.33



Printer Fig.no1.34



Speaker Fig.no 1.35



Plotter Fig.1.36

➤ **MONITOR:** -

A **monitor** or a **display** is an electronic visual display for computers. The monitor comprises the display device, circuitry, and an enclosure. The display device in modern monitors is typically a thin film transistor liquid crystal display (TFT-LCD) thin panel, while older monitors use a cathode ray tube (CRT) about as deep as the screen size.

➤ **PRINTER:** -

In computing, a **printer** is a peripheral which produces a representation of an electronic document on physical media such as paper or transparency film. Many printers are local peripherals connected directly to a nearby personal computer. Individual printers are often designed to support both local and network connected users at the same time.

**Types of printer:**

- Inkjet Printers
- Laser Printers
- Dot Matrix

➤ **SPEAKER:** -

**Computer speakers**, or **multimedia speakers**, are speakers' external to a computer that disables the lower fidelity built-in speaker.

➤ **PLOTTER:** -

The **plotter** is an output device where a computer printer for printing vector graphics. In the past, plotters were used in applications such as computer-aided design, though they have generally been replaced with wide-format conventional printers.

**SOFTWARE: -**

The software is defined as a computer program or collection of programs which are designed to solve any task.

**TYPES OF SOFTWARE: -**

1. System software
2. Application software

**SYSTEM SOFTWARE:** - System software refers to a computer program that manages and controls hardware component of a computer system. The system software is responsible for handling the function of computer hardware. It is also responsible for the proper function of application software on a computer.

**EXAMPLE: -**

1. **OPERATING SYSTEM SOFTWARE:** -An operating system is a program which acts as an interface between user and hardware.
2. **LANGUAGE PROCESSOR:** -These are system software which is used to translate the instruction of any programmable language in the form that can be executed by the computer system,  
Example: - Computer, Interpreter, Assembler.

**APPLICATION SOFTWARE:** - Application software is a computer program that executes on system software it is designed and develops for performing a specific task for a specific application.

**It can be of two types**

1. **Special purpose Application Software**
2. **General purpose Application Software**

The application software may be used for painting, making presentations (MS PowerPoint), making documents and reports (MS Word), playing songs or videos(WinAmp), for managing the payroll, for managing the accounts(Tally), for image processing, for compressing the file (WinZip), for searching the location on the globe (Google Earth), for creating architectural design(CAD/CAM), etc. For application software to work smoothly, the system software is also required to be preloaded. The software may be a single program (MS Paint, Notepad) or a collection of programs (MS office) known as a software package. Application program doesn't necessarily come with the computer, rather the user must purchase them or freely download them to use.

**Computer Application in e-Business: -**

E-Business (e-Business), or Electronic Business, is the administration of conducting business via the internet. This would include the buying and selling of goods and services, along with providing technical or customer support through the Internet. E-Business is a term often used in conjunction with e-commerce but includes services in addition to the sale of goods.

**Electronic business** commonly referred to as "**E-Business**" or "**E-business**", or an internet business, may be defined as the application of information and communication technologies (ICT) in support of all the activities of the business. Commerce constitutes the exchange of products and services between businesses, groups, and individuals and is one of the essential activities of any business. Electronic

commerce focuses on the use of ICT to enable the external activities and relationships of the business with individuals, groups and other businesses.



**Fig.no 1.37**

#### **Bio-Informatics: -**

Bioinformatics is the field of science which applies computer-based tools and technologies on biological research and development. It primarily involves collection and storage of biological and genetic data on which statistical techniques are applied to arrive at the required solution.

Bioinformatics has become an important part of many areas of biology. In experimental molecular biology, bioinformatics techniques such as image and signal processing allow extraction of useful results from large amounts of raw data. In the field of genetics and genomics, it aids in sequencing and annotating genomes and their observed mutations. It plays a role in the textual mining of biological literature and the development of biological and gene ontologies to organize and query biological data. It plays a role in the analysis of gene and protein expression and regulation.

#### **Health Care: -**

Now a day, computers are being used to cater to several different aspects of healthcare. The use of a computer is evident right from the beginning when a patient approaches healthcare facility. The healthcare staffs log the patient's details in an organized manner in a computer system. The same system is used for finding and allocating a vacant bed to the patient if required.

The most significant use of computers within healthcare has been its amalgamation with medical equipment. Most of the medical equipment is now computer-based, thus enabling accurate capture of data in digital form. Further, a device like CT scanner helps the physicians to view a 3-D image of body organs.

Health care (or healthcare) is the diagnosis, treatment, and prevention of disease, illness, injury, and other physical and mental impairments in humans. Health care is delivered by practitioners in medicine, chiropractic, dentistry, nursing, pharmacy, allied health, and other care providers. It refers to the work done in providing primary care, secondary care, and tertiary care, as well as in public health.



**Fig.no 1.38**

#### **Remote Sensing: -**

Remote sensing is the technique of acquiring information about a subject (material or spatial) without coming in direct contact with it. Since there are no direct contact involved, wireless devices are used for performing remote sensing task. Such devices are typically real-time systems that continuously gather and store data related to the subject under observation. A **RADAR** system can be considered as a good example of remote sensing device that measures the time delay between sending and receiving of signals to detect information related to the objects.



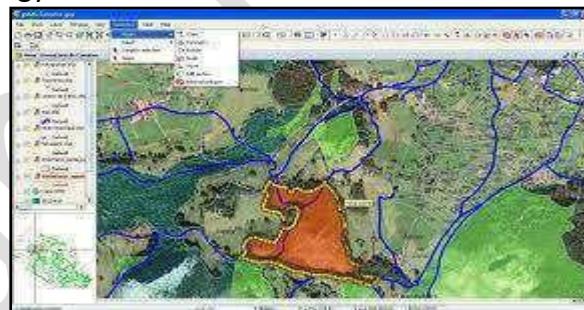
**Fig.no 1.39**

#### **GIS (Geographic information system): -**

Geographic information system (GIS) is a system that gathers location-specific data present in various meaningful forms. It is basically a computer-based information system that captures and stores location-specific data against different parameters.

A geographic information system (GIS) lets us visualize, question, analyse, interpret, and understand data to reveal relationships, patterns, and trends.

**Geographic information system (GIS)** is a system designed to capture, store, manipulate, analyse, manage, and present all types of geographical. The acronym **GIS** is sometimes used for **geographical information science** or **geospatial information studies** to refer to the academic discipline or career of working with geographic information systems. In the simplest terms, GIS is the merging of cartography, statistical analysis, and database technology.



**Fig.no 1.40**

#### **Meteorology and Climatology: -**

Meteorology is the study of the atmosphere and the related weather condition over short time intervals aims at making routine weather forecasts.

**Meteorology** is the interdisciplinary scientific study of the atmosphere. Studies in the field stretch back millennia, though significant progress in meteorology did not occur until the 18th century. The 19th century saw breakthroughs occur after observing networks developed across several countries. After the development of the computer in the latter half of the 20th century, breakthroughs in weather forecasting were achieved.



Fig.no 1.41

**Computer Gaming: -**

Computers are widely used for playing games that are like video or console-based games. A computer must process graphics and animations support for ensuring rich gaming experience to the user. The computer gaming industry has evolved tremendously over the past decade.

Computer and video games are a maturing medium and industry and have caught the attention of scholars across a variety of disciplines. By and large, computer and video games have been ignored by educators.



Fig.no 1.42

**Multimedia: -**

Multimedia is media and content that uses a combination of different content forms. This contrasts with media that use only rudimentary computer displays such as text-only or traditional forms of printed or hand-produced material. Multimedia includes a combination of text, audio, still images, animation, video, or interactivity content forms.

Multimedia is usually recorded and played, displayed, or accessed by information content processing devices, such as computerized and electronic devices, but can also be part of a live performance. Multimedia devices are electronic media devices used to store and experience multimedia content.



Fig.no 1.43

**Animation:-**

**The animation** is the rapid display of a sequence of images to create an illusion of movement. The most common method of presenting animation is as a motion picture or video program, although there are other methods. This type of presentation is usually accomplished with a camera and a projector or a computer viewing screen which can rapidly cycle through images in a sequence. Animation can be made with either hand rendered art, computer-generated imagery, or three-dimensional objects, e.g. puppets or clay figures, or a combination of techniques. The position of each object in any image relates to the

position of that object in the previous and following images so that the objects each appear to fluidly move independently of one another. The viewing device displays these images in rapid succession, usually 24, 25, or 30 frames per second.



**Fig.no 1.44**

#### **Operating System: -**

Operating System is software that works as an interface between a user and the computer hardware. The primary objective of an operating system is to make computer system convenient to use and to utilize computer hardware in an efficient manner. The operating system performs the basic tasks such as receiving input from the keyboard, processing instructions and sending output to the screen.

Various types of operating systems are UNIX/LINUX/REDHAT/UBUNTU, MS-DOS, MS-Windows - 98/XP/Vista/windows7/windows8/windows10, Windows-NT/2000/2003/2005, OS/2 and Mac OS.

The operating system manages overall activities of a computer and the input/output devices attached to the computer. It is the first software you see when you turn on the computer, and the last software you see when the computer is turned off. It is the software that enables all the programs you use. At the simplest level, an operating system does two things:

The first, it manages the hardware and software resources of the computer system. These resources include the processor, memory, disk space, etc.

The second, it provides a stable, consistent way for applications to deal with the hardware without having to know all the details of the hardware.

#### **Functions of Operating System: -**

The various Functions those are Performed by the Operating System are as Explained below: -

#### **Operating System as a Resource Manager: -**

Operating system also known as the resource manager means operating system will manage all the resources those are attached to the system means all the resources like memory and processor and all the input-output devices those are attached to the system are known as the resources of the computer system and the operating system will manage all the resources of the system.

#### **Storage Management: -**

The operating system also controls the all the storage operations means how the data or files will be stored on the computers and how the files will be accessed by the users etc. All the operations those are responsible for storing and accessing the files is determined by the operating system.

**Process management:** the operating system also treats the process management means all the processes those are given by the user or the process those are system 's own process is handled by the operating system. The operating system will create the priorities for the user and start or stops the execution of the process and makes the child process after dividing the large processes into the small processes.

**Memory management:** - the operating system also manages the memory of the computer system means to provide the memory to the process and deallocate the memory from the process. And defines that if a process gets completed then this will deallocate the memory from the processes.

**Extended machine:** - The operating system also behaves like an extended machine means operating system also provides us sharing of files between multiple users, also provides some graphical environments and provides various languages for communications and provides many complex operations like using many hardware and software.

**Mastermind:** - The operating system also performs many functions and for those reasons, we can say that operating system is a mastermind. It provides booting without an operating system and provides facility to increase the logical memory of the computer system by using the physical memory of the computer system and provides various types of formats like NTFS and fat file systems.

### Types of Operating System

There are Many Operating Systems those have been developed for performing the operations those are requested by the user. There are many Operating Systems which have the capability to perform the requests those are received from the System. The Operating system can perform a single operation and multiple operations at a time. So, there are many types of Operating systems those are organized by using their Working Techniques.

**1) Serial Processing:** - The Serial Processing Operating Systems are those which perform all the instructions in a sequential manner or the Instructions those are given by the user will be executed by using the FIFO manner means First in First Out. All the Instructions those are entered first in the system will be executed first and the Instructions those are entered later will be executed later. For running the instructions, the program counter is used which is used for executing all the Instructions.

**2) Batch Processing:** - The Batch Processing is same as the Serial Processing Technique, but in the Batch Processing similar types of jobs are firstly prepared and they are stored on the Card, and that card will be submitted to the system for the processing. The System then Perform all the operations on the Instructions one by one, and a user can't be able to specify any input. Operating System will increment his Program Counter for executing the next Instruction.

The main problem is that the Jobs those are prepared for execution must be the same type and if a job requires for any type of input then this will not be possible for the user. Many times, a lot of time will be wasted for preparing the batch.

The speed of the processing the Job will depend on the Jobs and the results those are produced by the system in the difference of time which is used for giving or submit the Job and the time which is used for displaying the results on the screen.

**3) Multi-Programming:** - As we know that in the batch processing system there are multiple jobs execute by the System. The system first prepares a batch and after that, he will execute all the jobs those are stored into the batch. But the main problem is that if a processor job requires an Input and Output operation, then it is not possible and second there will be the wastage of the Time when we are preparing the batch and the CPU will remain idle at that time.

**4) Real-Time System:** - There is also an Operating System which is known as Real Time processing system. In this response time is already fixed. Means time to display the results after possessing has fixed by the processor or CPU. A real-time system is used at those places in which we require higher and timely response. These types of systems are used in the reservation. So, when we specify the request, the CPU will perform at that time. There are two types of Real-time system

**I. Hard Real-Time System:** - In the Hard-real-time system, time is fixed and we can't change any moments of the time of processing. Means CPU will process the data as we enter the Data.

**II. Soft Real-Time System:** - In the Soft Real-Time System, some moments can be changed, means after giving the command to the CPU. CPU performs the operation after a Microsecond.

**5) Distributed Operating System:** - Distributed Means data is stored and processed on multiple locations. When a data is stored in the multiple computers, those are placed in different locations. Distributed means in the Network, with collections of computers connected with each other.

Then if we want to take some data from another computer, then we use the distributed Processing System. We can also Insert and Remove the data from our location to another location. In this data is shared between many users, and we can also access all the Input and Output Devices are also accessed by Multiple Users.

**6) Multiprocessing:** -Generally, a computer has a Single Processor means a computer have a just one CPU for processing the instructions. But if we are running multiple jobs, then this will decrease the speed of CPU. For increasing the speed of processing then we use the Multiprocessing. In the Multi-Processing there are two or more CPU in a single operating system if one CPU will fail, then another CPU is used for providing backup to the first CPU. With the help of multiprocessing, we can execute many jobs at a time. All the operations are divided into the Number of CPU's, if the first CPU completed his work before the second CPU, then the work of the second CPU will be divided into the first and second.

**7) Parallel operating systems:** - are used to interface multiple networked computers to complete tasks in parallel. The architecture of the software is often a UNIX-based platform, which allows it to coordinate distributed loads between multiple computers in a network. Parallel operating systems are able to use software to manage all of the different resources of the computers running in parallel, such as memory, caches, storage space, and processing power. Parallel operating systems also allow a user to directly interface with all the computers in the network.

#### **File System:** -

A file can be "free formed", indexed or a structured collection of related bytes having to mean only to the one who created it. Or in other words, an entry in a directory is the file. The file may have attributes like name, creator, date, type, permissions etc.

#### File Structure

A file has various kinds of structure. Some of them can be:

- **Simple Record Structure** with lines of fixed or variable lengths.
- **Complex Structures** like a formatted document or reloadable load files.
- **No Definite Structure** like a sequence of words and bytes etc.
- **Attributes of a File**
- Following are some of the attributes of a file:
  - **Name.** It is the only information which is in human-readable form.
  - **Identifier.** The file is identified by a unique tag(number) within the file system.
  - **Type.** It is needed for systems that support different types of files.
  - **Location.** Pointer to file location on device.
  - **Size.** The current size of the file.
  - **Protection.** This controls and assigns the power of reading, writing, executing.
  - **Time, date, and user identification.** This is the data for protection, security, and usage monitoring.

### **File Access Methods: -**

The way that files are accessed and read into memory is determined by Access methods. Usually, a single access method is supported by systems while there are OS's that support multiple access methods.

#### **Sequential Access: -**

- Data is accessed one record right after another in an order.
- Read command cause a pointer to be moved ahead by one.
- Write command allocate space for the record and move the pointer to the new End of File.
- Such a method is reasonable for tape.

#### **Direct Access: -**

- This method is useful for disks.
- The file is viewed as a numbered sequence of blocks or records.
- There are no restrictions on which blocks are read/written; it can be done in any order.
- The user now says, "read n" rather than "read next".
- "n" is a number relative to the beginning of the file, not relative to an absolute physical disk location.

#### **Indexed Sequential Access: -**

- It is built on top of Sequential access.
- It uses an Index to control the pointer while accessing files.

#### **What is a Directory: -**

Information about files is maintained by Directories. A directory can contain multiple files. It can even have directories inside of them. In Windows, we also call these directories as folders.

Following is the information maintained in a directory:

- Name: The name visible to the user.
- Type: Type of the directory.
- Location: Device and location on the device where the file header is located.
- Size: Number of bytes/words/blocks in the file.
- Position: Current next-read/next-write pointers.
- Protection: Access control on read/write/execute/delete.
- Usage: Time of creation, access, modification etc.
- Mounting: When the root of one file system is "grafted" into the existing tree of another file system it's called Mounting.

### **File Systems: -**

Provide a means to store data organized as files as well as a collection of functions that can be performed on files.

Maintain a set of attributes associated with the file.

Typical operations include:

- **Create**
- **Delete**
- **Open**

- Close
- Read
- Write

### **File Management System Objectives: -**

- Meet the data management needs of the user
- Guarantee that the data in the file are valid
- Optimize performance
- Provide I/O support for a variety of storage device types
- Minimize the potential for lost or destroyed data
- Provide a standardized set of I/O interface routines to user processes
- Provide I/O support for multiple users in the case of multiple user systems

### **Minimal User Requirements: -**

- Should be able to create, delete, read, write and modify files
- May have controlled access to other users' files
- May control what type of accesses on the files
- Should be able to restructure the files in a form appropriate to the problem
- Should be able to move data between files
- Should be able to back up and recover files in case of damage
- Should be able to access his or her files by name rather than by numeric identifier

### **Device Drivers: -**

- Lowest level
- Communicates directly with peripheral devices
- Responsible for starting I/O operations on a device
- Processes the completion of an I/O request
- Considered to be part of the operating system

### **Introduction to MS word, MS power point, MS Excel: -**

#### **Word processing: -**

Word processing software is used to create and maintain electronic documents. Alterations can easily be made to stored documents (instead of retyping them), and multiple copies can be printed. Professional looking results can be obtained by using different fonts, and by incorporating graphics into a document.

#### **The MS Word 2007 window: -**

- The old menu system has been replaced by the Ribbon and the Office button. The Office button contains a menu of file-related commands.
- The Quick Access toolbar provides a set of frequently used commands. The default options are to save a file, to undo the last action, and to repeat your most recent action.
- The Ribbon tabs provide you with a set of tools that are relevant to what you are currently doing.
- The Title bar displays the name of the program and the name of the current document. If you haven't named the document yet, then it will be called something like document.
- Window controls are used to change the size of a window or to close it.
- The Vertical scrollbar is used to scroll up and down the page. If your page is wider than the screen display, then you will also see a Horizontal scrollbar across the bottom of the window.

- The Status and information bar displays useful information about your documents, such as the page count and many words.

### **Options for viewing a document in Word**

Word offers five different views

- Print Layout
- Full-screen reading
- Web layout
- Outline view
- Draft view

Non-printing characters are characters that are used to format your document, but that isn't displayed as text on your screen. These symbols can be very useful when you are looking for formatting errors in your document.

### **Save or Save As Option: -**

If you want to keep the same file name and location, then the Save icon and the Save menu option will both save the file with no further comment. If you'd like to save an existing document under a new name, or in a different location, then you need to use the Save As command.

### **Closing Word: -**

- There are several methods you can use to end Word:
- Click the Office button, and then click the Exit Word button in the bottom right corner.
- Alternatively, close the window by clicking on the X at the far-right edge of the title bar.

### **Excel spreadsheets: -**

A spreadsheet is the computer equivalent of a paper ledger sheet. It consists of a grid made of columns and rows. It is an environment that can make number manipulation easy. If you change the month/day amount, you will have to start the math all over again. On the computer version, you can change the values and the result is automatically recalculated. Basics of a Spreadsheet are made up of columns

**rows:-**and their intersections are called cells. In each cell, there may be the following types of data

- text (labels)
- number data (constants)
- formulas (mathematical equations that do all the work).

Microsoft Excel is an electronic spreadsheet application. ^A spreadsheet is the computer equivalent of a paper ledger sheet. Microsoft Excel can be used to organize, calculate, and analyze the data.

- format your data.
- organize your data by sorting it.
- name ranges of data and use the range names in formulas and navigation for automatic updating.
- use cell references rather than values in formulas allowing you to adjust formulas as you copy and move them across the spreadsheet.
- generate charts and graphs illustrating your data.
- automate and customize procedures by using macros.
- How will you Perform Mathematical Calculations in excel?

In Microsoft Excel, we can enter numbers and mathematical formulas into cells. Whether we enter a number or a formula, can reference the cell when we perform mathematical calculations such as addition, subtraction, multiplication, or division. When entering a mathematical formula, precede the formula with an equal sign. Use the following to indicate the type of calculation wishes to perform:

- +Addition
- -Subtraction
- \*Multiplication
- /Division
- ^ Exponential

**Auto fills feature in Excel** A very handy feature in Excel is AutoFill, which allows us to automatically fill cells with pre-set data. If we need to add the months of the year or the days of the week to our spreadsheets, we can do so using AutoFill.

It is also possible to customize the lists of data that work with Auto Fill so that easily add data that is used frequently. If regularly adds the same department names or part numbers to spreadsheets one can add these names to the AutoFill feature making it easier to enter them when needed.

#### **Workbook: -**

A workbook is a spreadsheet file. By default, each workbook in Excel contains three pages or worksheets. The term spreadsheet is often used to refer to a workbook, when in actual fact; spreadsheet refers to the computer program, such as Excel. So, when we open the Excel spreadsheet program it loads an empty workbook file consisting of three blank worksheets for you to use.

#### **Features of Excel: -**

- Hyperlink: -We can link one file to another file or page with the use of Excel.
- Clipart: - In this, we can add images and audio, video clips can be added here.
- Charts: - With charts, we can clearly show products evaluation to the clients.
- Tables: - Tables are created with different fields eg -name, age, address, roll no so we add a table to fill these values.
- Functions: - MATHEMATICAL: Add, subtract, div, multiply. LOGICAL: average, sum, mod, product.
- Images and Backgrounds: - In this, we add images and backgrounds to the sheet.

**Macros:** - Macros are used for recording events for further use.

**Database:** - We can add database from other sources with data feature.

**Sorting and Filter:** - In sorting, we can sort our data and filter our data so that repetitions will be removed.

**Data Validations:** - In data tools, there are data validations consolidate etc are used.

**Grouping:** - In this, we can use group, ungroup subtotal etc.

**Page layout:** - In this theme, colors, sheets, margins, size, backgrounds, breaks, print, titles, sheets height, width, scaling, goodness, headings, views, bring to the front of font or back alignment etc will be used.

#### **Excel Headers and Footers: -**

In Microsoft Excel, headers and footers are lines of text that print at the top (header) and bottom (footer) of each page in the spreadsheet. They contain descriptive text such as titles, dates, and/or page numbers. They are used to add information to a spreadsheet that is being printed. Excel has several pre-set headers and footers that you can use. These standard options include the sheet name, date, time, page number,

and file name. There is also an option for creating your own custom headers or footers text, graphics, or other spreadsheet data.

### **The difference between a worksheet and a workbook in Excel: -**

A worksheet is a single spreadsheet page and a workbook is a collection of all the worksheets in a single file. A workbook contains worksheets, in the same way, that a book contains pages. A workbook consists of one or more worksheets.

For example, if you had one sheet that was a table with information and another sheet with a pie chart, you would have a workbook. On the other hand, a worksheet is one spreadsheet in Excel that you are working on.

### **PowerPoint – for presentations: -**

Microsoft PowerPoint is a presentation program currently developed by Microsoft, for use on both Microsoft Windows and Apple Macintosh operating systems. PowerPoint, initially named "Presenter", was created by Forethought Inc. Microsoft's version of PowerPoint was officially launched on May 22, 1990, as a part of the Microsoft Office suite. PowerPoint is useful for helping develop the slide-based presentation format and is currently one of the most commonly used slide-based presentation programs available. Microsoft has also released the PowerPoint mobile application for use on Apple and Android mobile operating systems. The following activities will be carried out with the help of powerpoint presentation: -

#### **Create a new presentation: -**

using the wizard, design-template, or blank presentation

Using blank-presentation is the most flexible for you

Understand the tools inside of power point

Moving from slide to slide – Presenting your slide Creating a new presentation

"AutoContent Wizard" walks you through a series of questions about your presentation, letting you choose from a variety of predetermined content themes, visual styles, and formatting options. You can answer all the questions or skip some and click "Finish" at any time.

"Design Template" allows you to choose from various background designs and slide styles to use throughout your presentation. Clicking the various presentation options shows thumbnail views of their designs. Click "OK" to choose one.

"Blank Presentation" starts you out with blank slides

VIEWS – PowerPoint has several 'views' that allows you to edit and see presentation content in various useful ways. You can switch among the views as you work by selecting one from the View menu or clicking one of the five 'view' buttons in the lower left-hand corner of the PowerPoint window. - In the Normal view, the screen is split into three sections showing the presentation outline on the left, the slide in the main window, and notes at the bottom.

## UNIT II

### Introduction to Algorithms,

An algorithm is a method for solving a computational problem. A formula or set of steps for solving a problem. To be an algorithm, a set of rules must be unambiguous and have a clear stopping point. Algorithms can be expressed in any language, from natural languages like English or French to programming languages like FORTRAN.

We use algorithms every day. For example, a recipe for baking a cake is an algorithm. Most programs, except for some artificial intelligence applications, consist of algorithms. Inventing elegant algorithms that are simple and require the fewest steps possible is one of the principal challenges in programming.

For example, we might be able to say that our algorithm indeed correctly solves the problem in question and runs in time at most  $f(n)$  on any input of size  $n$ .

**Definition.** An algorithm is a finite set of instructions for performing a computation or solving a problem.

**Types of Algorithms Considered.** In this course, we will concentrate on several different types of relatively simple algorithms, namely:

**Selection** -- Finding a value in a list, counting numbers;

**Sorting** -- Arranging numbers in order of increasing or decreasing value; and

**Comparison** -- Matching a test pattern with patterns in a database.

**Properties of Algorithms.** It is useful for an algorithm to have the following properties:

**Input** -- Values that are accepted by the algorithm is called input or arguments.

**Output** -- The result produced by the algorithm is the solution to the problem that the algorithm is designed to address.

**Definiteness** -- The steps in each algorithm must be well defined.

**Correctness** -- The algorithm must perform the task it is designed to perform, for all input combinations.

**Finiteness** -- Output is produced by the algorithm after a finite number of computational steps.

**Effectiveness** -- Each step of the algorithm must be performed exactly, infinite time.

**Generality** -- The procedure inherent in a specific algorithm should be applicable to all algorithms of the same general form, with minor modifications permitted.

### Complexities of Algorithm:

Algorithmic complexity is concerned about how fast or slow algorithm performs.

We define complexity as a numerical function  $T(n)$  - time versus the input size  $n$ . We want to define time taken by an algorithm without depending on the implementation details. But you agree that  $T(n)$  does depend on the implementation. A given algorithm will take different amounts of time on the same inputs depending on such factors as processor speed; instruction set, disk speed, a brand of compiler etc. The way around is to estimate the efficiency of each algorithm asymptotically. We will measure time  $T(n)$  as the number of elementary "steps" (defined in any way), provided each such step takes constant time.

Let us consider two classical examples: addition of two integers. We will add two integers digit by digit (or bit by bit), and this will define a "step" in our computational model. Therefore, we say that addition of two  $n$ -bit integers takes  $n$  steps. Consequently, the total computational time is  $T(n) = c * n$ , where  $c$  is time taken by addition of two bits. On different computers, the addition of two bits might take different time, say  $c_1$  and  $c_2$ , thus the addition of two  $n$ -bit integers takes  $T(n) = c_1 * n$  and  $T(n) = c_2 * n$  respectively. This shows that different machines result in different slopes, but time  $T(n)$  grows linearly as input size increases.

The process of abstracting away details and determining the rate of resource usage in terms of the input size is one of the fundamental ideas in computer science.

We use algorithms to introduce salient concepts and to concentrate on the analysis of algorithm performance, especially computational complexity.

**Concept.** To facilitate the design of efficient algorithms, it is necessary to estimate the bounds on the complexity of candidate algorithms.

**Representation.** Complexity is typically represented via the following measures, which are numbered in the order that they are typically computed by a system designer:

**Work  $W(n)$**  -- How many operations of each given type are required for an algorithm to produce a specified output given  $n$  inputs?

**Space  $S(n)$**  -- How much storage (memory or disk) is required for an algorithm to produce a specified output given  $n$  inputs?

**Time  $T(n)$**  -- How long does it take to compute the algorithm (with  $n$  inputs) on a given architecture?

**Cost  $C(n)$**  =  $T(n) \cdot S(n)$  -- Sometimes called the space-time bandwidth product, this measure tells a system designer what expenditure of aggregate computational resources is required to compute a given algorithm with  $n$  inputs.

**Procedure.** Analysis of algorithmic complexity generally proceeds as follows:

Step 1. Decompose the algorithm into steps and operations.

Step 2. For each step or operation, determine the desired complexity measures, typically using Big-Oh notation, or other types of complexity bounds discussed below.

Step 3. Compose the component measures obtained in Step 2 via theorems presented below, to obtain the complexity estimate(s) for the entire algorithm.

**Example.** Consider the following procedure for finding the maximum of a sequence of numbers. An assessment of the work requirement is given to the right of each statement.

```
Let { $a_n$ } = ( $a_1, a_2, \dots, a_n$ )
{ max =  $a_1$                                 # One I/O operation
  for i = 2 to n do:                      # Loop iterates  $n-1$  times
    { if  $a_i \geq max$  then                 # One comparison per iteration
      max :=  $a_i$  }                         # Maybe one I/O operation
  }
```

**Analysis:** -

(1) In the preceding algorithm, we note that there are  $n-1$  comparisons within the loop.

(2) In a randomly ordered sequence, half the values will be less than the mean, and  $a_1$  would be assumed to have the mean value (for purposes of analysis). Hence, there will be an *average* of  $n/2$  I/O operations to replace the value max with  $a_i$ . Thus, there are  $n/2 + 1$  I/O operations.

(3) This means that the preceding algorithm is  $O(n)$  in comparisons and I/O operations. More precisely, we assert that in the average case:

$W(n) = n-1$  comparisons +  $(n/2 - 1)$  I/O operations.

**Tractable** -- An algorithm belongs to class P, such that the algorithm is solvable in polynomial time (hence the use of P for *polynomial*). This means that complexity of the algorithm is  $O(n^d)$ , where d may be large (which typically implies slow execution).

**Intractable** -- The algorithm in question requires greater than polynomial work, time, space, or cost. Approximate solutions to such algorithms are often more efficient than exact solutions and are preferred in such cases.

**Solvable** -- An algorithm exists that generates a solution to the problem addressed by the algorithm, but the algorithm is not necessarily tractable.

**Unsolvable** -- No algorithm exists for solving the given problem. *Example:* Turing showed that it is impossible to decide whether a program will terminate on a given input.

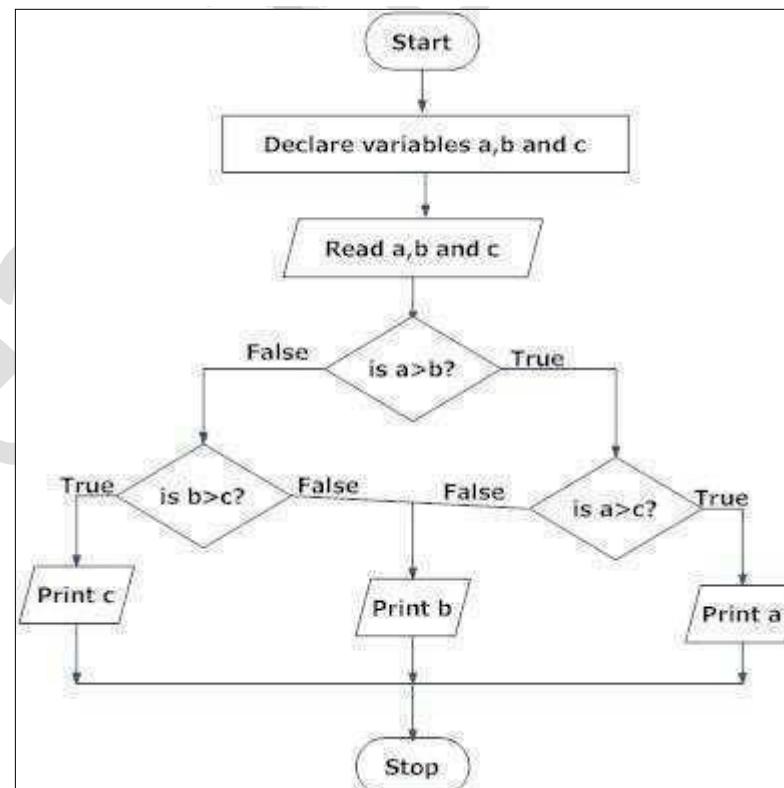
**Class NP** -- If a problem is in Class NP (non-polynomial), then no algorithm with polynomial worst-case complexity can solve this problem.

**Class NP-Complete** -- If any problem is in Class NP-Complete, then any polynomial-time algorithm that could be used to solve this problem could solve all NP-complete problems. Also, it is generally accepted, but not proven, that no NP-complete problem can be solved in polynomial time.

### FLOWCHART

1. The flowchart is a type of diagram (graphical or symbolic) that represents an algorithm or process.
2. Each step in the process is represented by a different symbol and contains a short description of the process step.
3. The flowchart symbols are linked together with arrows showing the process flow direction.
4. A flowchart typically shows the flow of data in a process.
5. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.
6. Flowcharts are generally drawn in the early stages of formulating computer solutions.
7. Flowcharts often facilitate communication between programmers and business people.

Example: - Draw a flowchart to find the largest among three different numbers entered by the user.



### Advantages of Using Flowcharts

1. Communication: a better way of logical communicating.

2. Effective analysis: the problem can be analyzed in more effective way.
3. Proper documentation.
4. Efficient Coding: a guide or blueprint during the systems analysis and program development phase.  
Proper Debugging: helps in debugging process.
5. Efficient Program Maintenance: easy with the help of flowchart.

### **Limitations of Using Flowcharts**

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

### **Flowchart Symbols**

1. Terminator: "Start" or "End"
2. Process
3. Decision: Yes/No question or True/False
4. Connector: jump in the process flow
5. Data: data input or output (I/O)
6. Delay
7. Arrow: flow of control in a process.

Symbol	Purpose	Description
	Flowline	Used to indicate the flow of logic by connecting symbols.
	Terminal(Stop/Start)	Used to represent start and end of the flowchart.
	Input/output	Used for input and output operation.
	Processing	Used for arithmetic operations and data-manipulations.
	Decision	Used to represent the operation in which there are two alternatives, true and false.
	On-page Connector	Used to join different flowline
	Off-page Connector	Used to connect flowchart portion on a different page.

Symbol	Purpose	Description
	Predefined Process/Function	Used to represent a group of statements performing one processing task.

## Introduction to Programming

An organized list of instructions that, when executed, causes the computer to behave in a predetermined manner. Without programs, computers are useless.

A program is like a recipe. It contains a list of ingredients (called *variables*) and a list of directions (called *statements*) that tell the computer what to do with the variables. The variables can represent numeric data, text, or graphical images.

There are many programming languages -- C, C++, Pascal, BASIC, FORTRAN, COBOL, and LISP are just a few. These are all high-level languages. One can also write programs in *low-level languages* called assembly languages, although this is more difficult. Low-level languages are closer to the language used by a computer, while high-level languages are closer to human languages.

## Categories of Programming Language

Programming languages fall into two fundamental categories low and high-level languages. Low-level languages are machine-dependent; that is, they are designed to be run on a particular computer. In contrast, high-level languages (for example, COBOL and BASIC) are machine-independent and can be run on a variety of computers.

The hierarchy of programming languages contain various types of programming languages. Through the first four decades of computing, programming languages evolved in generations. The first two generations were low-level and the next two high-level generations of programming languages.

The higher-level languages do not provide us greater programming capabilities, but they do provide a more sophisticated programmer/computer interaction. In short, the higher the level of the language, the easier it is to understand and use. For example, in a fourth-generation language, you need only instruct the computer system what to do, not necessarily how to do it.

## Characteristics of Programming language

The following are the characteristics of a programming language

1. **Readability:** A good high-level language will allow programs to be written in some ways that resemble a quite-English description of the underlying algorithms. If care is taken, the coding may be done in a way that is essentially self-documenting.
2. **Portability:** High-level languages, being essentially machine independent, should be able to develop portable software.
3. **Generality:** Most high-level languages allow the writing of a wide variety of programs, thus relieving the programmer of the need to become an expert in many diverse languages.

**4. Brevity:** Language should have the ability to implement the algorithm with less amount of code.

Programs expressed in high-level languages are often considerably shorter than their low-level equivalents.

**5. Error checking:** Being human, a programmer is likely to make many mistakes in the development of a computer program. Many high-level languages enforce a great deal of error checking both at compile-time and at run-time.

**6. Cost:** The ultimate cost of a programming language is a function of many of its characteristics.

**7. Familiar notation:** A language should have a familiar notation, so it can be understood by most of the programmers.

**8. Quick translation:** It should admit quick translation.

**9. Efficiency:** It should permit the generation of efficient object code.

**10. Modularity:** It is desirable that programs can be developed in the language as a collection of separately compiled modules, with appropriate mechanisms for ensuring self-consistency between these modules.

**11. Widely available:** Language should be widely available and it should be possible to provide translators for all the major machines and for all the major operating systems.

## Generation of Languages

**1GL** or first-generation language was (and still is) machine language or the level of instructions and data that the processor is given to work on (which in conventional computers is a string of 0s and 1s).

**2GL** or second-generation language is assembler (sometimes called "assembly") language. A typical 2GL instruction looks like this:

```
ADD 12,8
```

An assembler converts the assembler language statements into machine language.

**3GL** or third-generation language is a "high-level" programming language, such as PL/I, C, or Java. Java language statements look like this:

```
public boolean handleEvent (Event evt) {  
  
    switch (evt.id) {  
  
        case Event.ACTION_EVENT: {  
  
            if ("Try me" .equald(evt.arg)) {
```

A compiler converts the statements of a specific high-level programming language into machine language. (In the case of Java, the output is called bytecode, which is converted into appropriate machine language by a Java virtual machine that runs as part of an operating system platform.) A 3GL language requires a considerable amount of programming knowledge.

**4GL** or fourth-generation language is designed to be closer to natural language than a 3GL language.

Languages for accessing databases are often described as 4GLs. A 4GL language statement might look like this:

```
EXTRACT ALL CUSTOMERS WHERE "PREVIOUS PURCHASES" TOTAL MORE THAN $1000
```

**5GL** or fifth-generation language is programming that uses a visual or graphical development interface to create source language that is usually compiled with a 3GL or 4GL language compiler. Microsoft, Borland, IBM, and other companies make 5GL visual programming products for developing applications in Java, for example. Visual programming allows you to easily envision object-oriented programming class hierarchies and drags icons to assemble program components.

## Programming paradigms

Programming paradigms are a way to classify programming languages based on their features. Languages can be classified into multiple paradigms.

Some paradigms are concerned mainly with implications for the execution model of the language, such as allowing side effects, or whether the sequence of operations is defined by the execution model. Other paradigms are concerned mainly with the way that code is organized, such as grouping a code into units along with the state that is modified by the code. Yet others are concerned mainly with the style of syntax and grammar.

### Common programming paradigms include:

- Imperative which allows side effects,
- Functional which disallows side effects,
- Declarative which does not state the order in which operations execute,
- Object-oriented which groups code together with the state the code modifies,
- Procedural which groups code into functions,
- Logic which has a style of execution model coupled to a style of syntax and grammar, and
- Symbolic programming which has a style of syntax and grammar.

### Procedural oriented programming (pop):-

A program in a procedural language is a list of instruction where each statement tells the computer to do something. It focuses on procedure (function) & algorithm is needed to perform the derived computation.

When the program becomes larger, it is divided into function & each function has clearly defined purpose. Dividing the program into functions & module is one of the cornerstones of structured programming.

E.g.: c, basic, FORTRAN.

### Characteristics of Procedural oriented programming: -

- It focuses on process rather than data.
- It takes a problem as a sequence of things to be done such as reading, calculating and printing. Hence, many functions are written to solve a problem.
- A program is divided into many functions and each function has clearly defined purpose.
- Most of the functions share global data.
- Data moves openly around the system from function to function.

### Drawback of Procedural oriented programming (structured programming):-

- It's emphasis on doing things. Data is given a second-class status even though data is the reason for the existence of the program.
- Since every function has complete access to the global variables, the new programmer can corrupt the data accidentally by creating function. Similarly, if new data is to be added, all the function needed to be modified to access the data.
- It is often difficult to design because the components function and data structure do not model the real world.
- For example, in designing graphical user interface, we think what functions, what data structures are needed rather than which menu, menu item and soon.
- It is difficult to create new data types. The ability to create the new data type of its own is called extensibility. Structured programming languages are not extensible.

### Difference between Procedure Oriented Programming (POP) & Object-Oriented Programming (OOP)

Object-Oriented Programming	Procedure Oriented Programming	Points
In OOP, the program is divided into parts called objects.	In POP, the program is divided into small parts called functions.	Divided Into
In OOP, Importance is given to the data rather than procedures or functions because it works as a real world.	In POP, Importance is not given to data but to functions as well as the sequence of actions to be done.	Importance
OOP follows Bottom Up approach.	POP follows Top-Down approach.	Approach
OOP has access specifiers named Public, Private, Protected, etc.	POP does not have any access specifier.	Access Specifiers
In OOP, objects can move and communicate with each other through member functions.	In POP, Data can move freely from function to function in the system.	Data Moving
OOP provides an easy way to add new data and function.	To add new data and function in POP is not so easy.	Expansion
In OOP, data cannot move easily from function to function, it can be kept public or private so we can control the access of data.	In POP, the most function uses Global data for sharing that can be accessed freely from function to function in the system.	Data Access
OOP provides Data Hiding so provides more security.	POP does not have any proper way for hiding data so it is less secure.	Data Hiding
In OOP, overloading is possible in the form of Function Overloading and Operator Overloading.	In POP, Overloading is not possible.	Overloading

Example of OOP are: C++, JAVA, VB.NET, C#.NET.	Example of POP are: C, VB, FORTRAN, Pascal.	Examples
--	---	----------

The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods. While designing C++ modules, we try to see the whole world in the form of objects. For example, a car is an object which has certain properties such as color, the number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.

### **There are a few principle concepts that form the foundation of object-oriented programming:**

#### **Object**

This is the basic unit of object-oriented programming. That is both data and functions that operate on data are bundled as a unit called as an object.

#### **Class**

When you define a class, you define a blueprint for an object. This doesn't define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

#### **Abstraction**

Data abstraction refers to, providing only essential information to the outside world and hiding their background details, i.e., to represent the needed information in program without presenting the details.

For example, a database system hides certain details of how data is stored and created and maintained.

#### **Encapsulation**

Encapsulation is placing the data and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which functions work on which variables but object-oriented programming provides you a framework to place the data and the relevant functions together in the same object.

#### **Inheritance**

One of the most useful aspects of object-oriented programming is code reusability. As the name suggests Inheritance is the process of forming a new class from an existing class that is from the existing class called as a base class, a new class is formed called as the derived class.

This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

#### **Polymorphism**

The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism. Poly refers to many. That is a single function or an operator functioning in many ways different upon the usage is called polymorphism.

#### **Overloading**

The concept of overloading is also a branch of polymorphism. When the exiting operator or function is made to operate on new data type, it is said to be overloaded.

#### **Advantage of OOP**

- Simplicity: software objects model real-world objects, so the complexity is reduced and the program structure is very clear;
- Modularity: each object forms a separate entity whose internal workings are decoupled from other parts of the system;
- Modifiability: it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program since the only public interface that the external world must a class is using methods;
- Extensibility: adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones;
- Maintainability: objects can be maintained separately, making locating and fixing problems easier;

- e-usability: objects can be reused in different programs.

## Introduction to C++

### History

The C++ programming language is an extension of C that was developed by Bjarne Stroustrup in the early 1980s at Bell Laboratories. C++ provides many features that "spruce up" the C language, but more importantly, it provides capabilities for object-oriented programming. A computer cannot understand our language that we use in our day to day conversations, and likewise, we cannot understand the binary language that the computer uses to do its tasks. It is, therefore, necessary for us to write instructions in some specially defined language like C++ which is like natural language and after conversing with the help of compiler the computer can understand it.

### Significant Language Features

Object-oriented programs are easier to understand, correct and modify. Many other object-oriented languages have been developed, including most notably, Smalltalk. The best features of C++ are:

- C++ is a hybrid language-it is possible to program in either a C-like style, an object-oriented style, or both.
- C++ programs consist of pieces called classes and functions. You can program each piece you may need to form a C++ program. The advantage of creating your own functions and classes is that you will know exactly how they work. You will be able to examine the C++ code.

### Areas of Application

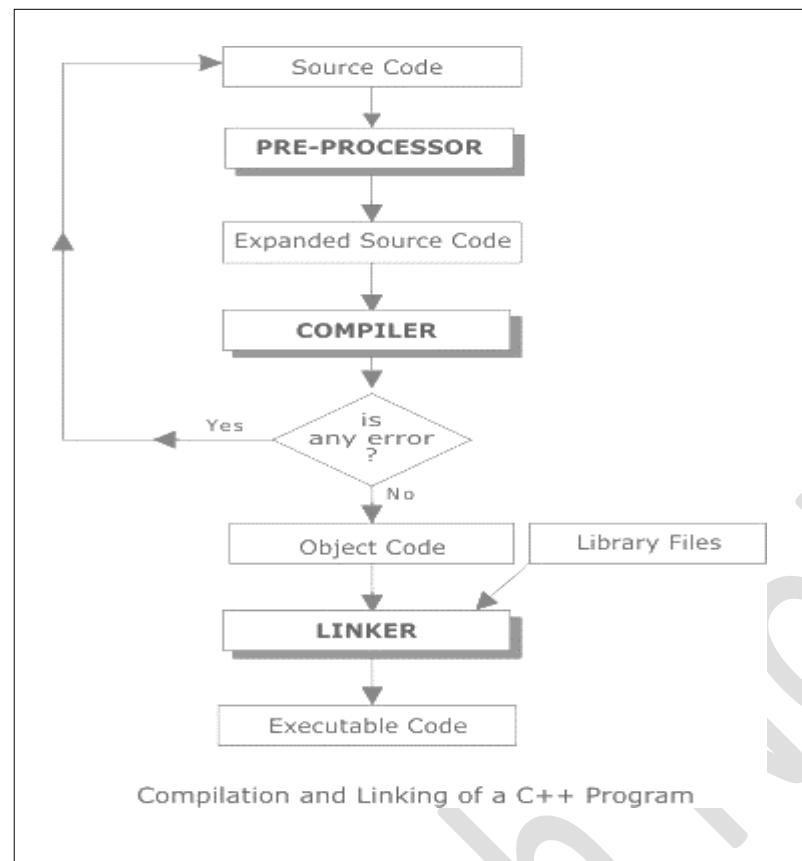
- For Develop Graphical related application like a computer and mobile games.
- To evaluate any kind of mathematical equation use C++ language.
- C++ Language is also used for design OS. Like window xp.
- Google also use C++ for Indexing
- Few parts of Apple OS X are written in C++ programming language.
- Internet browser Firefox is written in C++ programming language
- All major applications of Adobe systems are developed in C++ programming language. Like Photoshop, Image Ready, Illustrator, and Adobe Premier.
- Some of the Google applications are also written in C++, including Google file system and Google Chromium.
- C++ is used for design database like MySQL.

### C++ COMPILER

A C++ compiler is itself a computer program which's the only job is to convert the C++ program from our form to a form the computer can read and execute. The original C++ program is called the "**source code**", and the resulting compiled code produced by the compiler is usually called an "**object file**".

Before compilation, the **preprocessor** performs preliminary operations on C++ source files. A preprocessed form of the source code is sent to the compiler.

After compilation stage object files are combined with predefined libraries by a **linker**, sometimes called a binder, to produce the final complete file that can be executed by the computer. A library is a collection of pre-compiled "object code" that provides operations that are done repeatedly by many computer programs.



### Using Turbo C++ Com

If you are still willing to set up your environment for C++, you need following two software available on your computer.

#### Text Editor:

This will be used to type your program. Examples of few editors include Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.

Name and version of text editor can vary on different operating systems. For example, Notepad will be used on Windows and vim or vi can be used on windows as well as Linux, or UNIX.

The files you create with your editor are called source files and for C++ they typically are named with the extension .cpp, .cp, or .c.

#### C++ Compiler:

This is an actual C++ compiler, which will be used to compile your source code into the final executable program.

Most C++ compilers don't care what extension you give your source code, but if you don't specify otherwise, many will use .cpp by default

Most frequently used and the free available compiler is GNU C/C++ compiler, otherwise, you can have compilers either from HP or Solaris if you have respective Operating Systems.

### C++ BASICS

#### C++ CHARACTER SET

The character set is a set of valid characters that a language can recognize.

<b>Letters</b>	A-Z, a-z
<b>Digits</b>	0-9

<b>Special Characters</b>	Space + - * / ^ \ () [] {} = != <> ' " \$ , ; : % ! & ? _ # <= >= @
<b>Formatting characters</b>	backspace, horizontal tab, vertical tab, form feed, and carriage return

## TOKENS

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

*Keywords, Identifiers, Literals, Punctuators, Operators.*

### 1. Keywords

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

Asm	Auto	Break	case	catch
Char	Class	Const	continue	default
Delete	Do	Double	else	enum
Extern	inline	Int	float	for
friend	Goto	If	long	new
operator	private	Protected	public	register
return	Short	Signed	sizeof	static
struct	switch	Template	this	Try
typedef	union	Unsigned	virtual	void
volatile	While			

### 2. Identifiers

Symbolic names can be used in C++ for various data items used by a programmer in his program. A symbolic name is generally known as an identifier. The identifier is a sequence of characters taken from C++ character set. The rule for the formation of an identifier are:

- An identifier can consist of alphabets, digits and/or underscores.
- It must not start with a digit
- C++ is case sensitive that is upper case and lower-case letters are considered different from each other.
- It should not be a reserved word.

### 3. Literals

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

- Integer-Constants
- Character-constants
- Floating-constants
- Strings-constants

**Integer Constants:** Integer constants are the whole number without any fractional part. C++ allows three types of integer constants.

**Decimal integer constants:** It consists of a sequence of digits and should not begin with 0 (zero). For example, 124, - 179, +108.

**Octal integer constants:** It consists of a sequence of digits starting with 0 (zero). For example. 014, 012.

**Hexadecimal integer constant:** It consists of a sequence of digits preceded by ox or OX.

#### Character constants

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

Escape Sequence	Nongraphic Character
\a	Bell (beep)
\n	Newline
\r	Carriage Return
\t	Horizontal tab
\0	Null Character

#### Floating constants

They are also called real constants. They are numbers having fractional parts. They may be written in fractional form or exponent form. A real constant in fractional form consists of signed or unsigned digits including a decimal point between digits. For example 3.0, -17.0, -0.627 etc.

#### String Literals

A sequence of characters enclosed in double quotes is called a string literal. String literal is by default (automatically) added a special character '\0' which denotes the end of the string. Therefore, the size of

the string is increased by one character. For example "COMPUTER" will re-represented as "COMPUTER\0" in the memory and its size is 9 characters.

#### 4. Punctuators

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

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

**5. Operators:** Operators are special symbols used for specific purposes. C++ provides six types of operators. Arithmetical operators, Relational operators, Logical operators, Unary operators, Assignment operators, Conditional operators, Comma operator

#### DATA HANDLING

##### BASIC DATA TYPES

C++ supports many data types. The built-in or basic data types supported by C++ are an integer, floating point, and character. These are summarized in table along with description and memory requirement

Type	Byte	Range	Description
Int	2	-32768 to +32767	Small whole number
long int	4	-2147483648 to +2147483647	Large whole number
Float	4	3.4x10-38 to 3.4x10+38	Small real number
double	8	1.7x10-308 to 1.7x10+308	Large real number

long double	10	3.4x10-4932 to 3.4x10+4932	Very Large real number
char	1	0 to 255	A Single Character

## VARIABLES

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

To understand more clearly, we should study the following statements:

Total = 20.00; In this statement, a value 20.00 has been stored in a memory location Total.

### Declaration of a variable

Before a variable is used in a program, we must declare it. This activity enables the compiler to make available the appropriate type of location in the memory.

```
float Total;
```

You can declare more than one variable of the same type in a single statement

```
int x,y;
```

### Initialization of variable

When we declare a variable its default value is undetermined. We can declare a variable with some initial value.

```
int a = 20;
```

### Primitive Built-in Types

C++ offers the programmer a rich assortment of built-in as well as user-defined data types. Following table lists down seven basic C++ data types:

Keyword	Type
Bool	Boolean
Char	Character
Int	Integer
Float	Floating point
Double	Double floating point
Void	Valueless
wchar_t	Wide character

Following is the example, which will produce the correct size of various data types on your computer.

```
#include<iostream>

using namespace std;

int main(){
    cout<<"Size of char : "<<sizeof(char)<<endl;
    cout<<"Size of int : "<<sizeof(int)<<endl;
    cout<<"Size of short int : "<<sizeof(shortint)<<endl;
    cout<<"Size of long int : "<<sizeof(longint)<<endl;
    cout<<"Size of float : "<<sizeof(float)<<endl;
    cout<<"Size of double : "<<sizeof(double)<<endl;
    cout<<"Size of wchar_t : "<<sizeof(wchar_t)<<endl;
    return0;
}
```

This example uses endl, which inserts a new-line character after every line and << operator is being used to pass multiple values out to the screen. We are also using sizeof() operator to get sizeof various data types.

When the above code is compiled and executed, it produces the following result which can vary from machine to machine –

```
Size of char: 1
Size of int: 4
Size of short int: 2
Size of long int: 8
Size of float: 4
Size of double: 8
Size of wchar_t: 4
```

### Type Qualifiers in C++

The type qualifiers provide additional information about the variables they precede.

Meaning	Qualifier
Objects of type const cannot be changed by your program during execution	const
The modifier volatile tells the compiler that a variable's value may be changed in ways not explicitly specified by the program.	volatile
A pointer qualified by restricting is initially the only means by which the object it points to can be accessed. Only C99 adds a new type qualifier called restrict.	restrict

### STRUCTURE OF C++ PROGRAM

```
#include<header file>
main ()
{
.......
```

.....

.....

}

A C++ program starts with function called main ( ). The body of the function is enclosed in curly braces. The program statements are written within the braces. Each statement must end with a semicolon;(statement terminator). A C++ program may contain as many functions as required. However, when the program is loaded in the memory, the control is handed over to function main ( ) and it is the first function to be executed.

## Flow of control

### Statements

Statements are the instructions given to the computer to perform any kind of action. Action may be in the form of data movement, decision making etc. Statements form the smallest executable unit within a C++ program. Statements are always terminated by a semicolon.

### Compound Statement

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

### Null Statement

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

### Conditional Statements

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

- if statement
- if else statement
- nested if statement
- switch statement

### if statement

The syntax of the if statement

```
if (condition)
```

```
{
```

```
  statement(s);
```

```
}
```

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

### if else statement

syntax of the if - else statement

```
if (condition)
```

```
  statement1;
```

```
else
```

```
  statement2;
```

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

```

if example      if else example
if (x == 100)
    cout << "x is 100"; if (x == 100)
    cout << "x is 100";
else
    cout << "x is not 100";

```

### Nested if statement

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

The syntax of the nested if statement

```

if(condition 1)
{
    if(condition 2)
    {
        statement(s);
    }
}
if(condition 1)
    statement 1;
else if (condition 2)
    statement2;
else
    statement3;

```

### if-else-if example

```

if(percentage>=60)
    cout<<"Ist division";
else if(percentage>=50)
    cout<<"IIInd division";
else if(percentage>=40)
    cout<<"IIIrd division";
else
    cout<<"Fail" ;

```

### Switch statement

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

```

switch (var / expression)
{
    case constant1: statement 1;
    break;
    case constant2: statement2;
    break;
    .
    .
    default: statement3;
    break;
}

```

The execution of switch statement begins with the evaluation of an expression. If the value of expression matches with the constant then the statements following this statement execute sequentially till it

executes break. The break statement transfers control to the end of the switch statement. If the value of expression does not match with any constant, the statement with default is executed.

Some important points about switch statement

- The expression of a switch statement must be of type integer or character type.
- The default case need not be used at last case. It can be placed at any place.
- The case values need not be in specific order.

### **Flow of control**

#### **Looping statement**

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

- while loop
- do-while loop
- for loop

**While loop**

Syntax of while loop

while(condition)

{

    statement(s);

}

The flow diagram indicates that a condition is first evaluated. If the condition is true, the loop body is executed and the condition is re-evaluated. Hence, the loop body is executed repeatedly if the condition remains true. As soon as the condition becomes false, it comes out of the loop and goes to the state next to the 'while' loop.

**do-while loop**

Syntax of do-while loop

do

    {statements;

} while (condition);

Note: That the loop body is always executed at least once. One important difference between the while loop and the do-while loop the relative ordering of the conditional test and loop body execution. In the while loop, the loop repetition test is performed before each execution the loop body; the loop body is not executed at all if the initial test fails. In the do-while loop, the loop termination test is Performed after each execution of the loop body. Hence, the loop body is always executed least once.

### **for loop**

It is a count controlled loop in the sense that the program knows in advance how many times the loop is to be executed.

Syntax of for loop for (initialization; decision; increment/decrement)

{

    statement(s);

}

In the For loop Three operation takes place

- Initialization of loop control variable

- Testing of loop control variable
- Update the loop control variable either by incrementing or decrementing.

Operation (i) is used to initialize the value. On the other hand, operation (ii) is used to test whether the condition is true or false. If the condition is true, the program executes the body of the loop and then the value of loop control variable is updated. Again, it checks the condition and so on. If the condition is false, it gets out of the loop

#### Jump Statements

The jump statements unconditionally transfer program control within a function.

- goto statement
- break statement
- continue statement

#### The goto statement

goto allows making the jump to another point in the program. goto pqr;

pqr: pqr is known as a label. It is a user-defined identifier. After the execution of goto statement, the control transfers to the line after label pqr.

#### The break statements

The break statement, when executed in a switch structure, provides an immediate exit from the switch structure. Similarly, you can use the break statement in any of the loops. When the break statement executes in a loop, it immediately exits from the loop.

### ARRAY

An array is a collection of data elements of same data type. It is described by a single name and each element of an array is referenced by using array name and its subscript no.

#### Declaration of Array

Type arrayname[numberofelements];

For example, int Age[5] ;

float cost[30];

#### Initialization of One-Dimensional Array

An array can be initialized along with declaration. For array initialization, it is required to place the elements separated by commas enclosed within braces. int A[5] = {11,2,23,4,15}; It is possible to leave the array size open. The compiler will count the array size. int B[] = {6,7,8,9,15,12};

#### Referring to Array Elements

In any point of a program in which an array is visible, we can access the value of any of its elements individually as if it was a normal variable, thus being able to both read and modify its value. The format is as simple as:

name[index]

Examples: cout<<age[4]; //print an array element

age[4]=55; // assign value to an array element

cin>>age[4]; //input element 4

Using Loop to input an Array from user

int age [10], i ;

for (i=0 ; i<10; i++)

```
{
    cin>>age[i];
}
```

### Arrays as Parameters

At some moment, we may need to pass an array to a function as a parameter. In C++, it is not possible to pass a complete block of memory by value as a parameter to a function, but we can pass its address.

For example, the following function: void print(int A[]) accepts a parameter of type "array of int" called A.

In order to pass to this function, an array declared as int arr[20]; we need to write a call like this:  
print(arr);

### Functions in C++

A function is a group of statements that together perform a specific task. Every C++ program has at least one function, which is main ().

The function is used for dividing a large code into a module, due to this we can easily debug and maintain the code. For example, if we write calculator programs at that time we can write every logic in a separate function (For addition sum (), for subtraction sub()). Any function can be called many times.

### Advantage of Function

- Code Re-usability
- Develop an application in module format.
- Easily to debug the program.
- Code optimization: No need to write a lot of code.

### Type of Function

There are two types of function in C++ Language. They are;

- Library function or pre-define function.
- User defined function.

### Library function

Library functions are those which are predefined in C++ compiler. The implementation part of pre-defined functions is available in library files that are .lib/.obj files. .lib or .obj files are contained pre-compiled code. printf(), scanf(), clrscr(), pow() etc. are pre-defined functions.

### Limitations of Library function

- All predefined function is contained limited task only that is for what purpose function is designed for the same purpose it should be used.
- As a programmer, we do not have any controls on predefined function implementation part is there in machine-readable format.
- In implementation whenever a predefined function is not supporting user requirement then go for user-defined function.

### User-defined function

These functions are created by a programmer according to their requirement, for example, suppose you

want to create a function for add two number then you create a function with name sum() this type of function is called a user-defined function.

### **Defining a function.**

Defining of the function is nothing but give the body of the function that means write logic inside the function body.

#### Syntax

```
return_type function_name(parameter)
{
    function body;
}
```

**Return type:** A function may return a value. The return type is the data type of the value the function returns. Return type parameters and returns statement are optional.

**Function name:** Function name is the name of the function it is decided by programmer or you.

**Parameters:** This is a value which is passed to function at the time of calling of function A parameter is a placeholder. It is optional.

**Function body:** Function body is the collection of statements.

### **Function Declarations**

A function declaration is a process of tells the compiler about a function name. The actual body of the function can be defined separately.

#### Syntax

```
return_type function_name(parameter);
```

Note: At the time of function declaration function must be terminated with ';'.

### **Calling a function.**

When we call any function, control goes to function body and execute entire code. For a call, any function just writes the name of the function and if any parameter is required then pass a parameter.

#### Syntax

```
function_name();
```

or

```
variable=function_name(argument);
```

Note: At the time of function calling function must be terminated with ';'.

### **Example of Function in C++**

```
#include<iostream.h>
Using namespace std;

void sum(); // declaring a function
int a=10,b=20, c;

void sum() // defining function
```

```
{
c=a+b;
cout<<"Sum: "<<c;
}
void main()
{
sum(); // calling function
}
Output
```

Sum: 30

### Function Arguments in C++

If a function takes any arguments, it must declare variables that accept the values as arguments. These variables are called the formal parameters of the function. There are two ways to pass value or data to function in C++ language which is given below;

- call by value
- call by reference

#### Call by value

In call by value, **original value cannot be changed** or modified. In call by value, when you passed a value to the function it is locally stored by the function parameter in stack memory location. If you change the value of function parameter, it is changed for the current function only but it does not change the value of a variable inside the caller function such as main().

Program Call by value in C++

```
#include<iostream.h>
#include<conio.h>

void swap(int a, int b)
{
int temp;
temp=a;
a=b;
b=temp;
}

void main()
{
int a=100, b=200;
clrscr();
swap(a, b); // passing value to function
cout<<"Value of a"<<a;
cout<<"Value of b"<<b;
getch();
}
Output
```

Value of a: 200  
 Value of b: 100

### Call by reference

In call by reference, the original value is changed or modified because we pass a reference (address). Here, the address of the value is passed to the function, so actual and formal arguments share the same address space. Hence, any value changed inside the function is reflected inside as well as outside the function.

Example Call by Reference in C++

```
#include<iostream.h>
#include<conio.h>

void swap(int *a, int *b)
{
    int temp;
    temp=*a;
    *a=*b;
    *b=temp;
}

void main()
{
    int a=100, b=200;
    clrscr();
    swap(&a, &b); // passing value to function
    cout<<"Value of a"<<a;
    cout<<"Value of b"<<b;
    getch();
}
Output
```

Value of a: 200  
 Value of b: 100

### Difference Between Call by Value and Call by Reference.

call by Value	call by Reference
This method copies the original value into the function as arguments.	This method copies address of arguments into the function as an argument.
Changes made to the parameter inside the function have no effect on the argument.	Changes made to the parameter affect the argument. Because the address is used to access the actual argument.
Actual and formal arguments will be created in different memory location	Actual and formal arguments will be created in same memory location

### Pointer in C++

A pointer is a variable whose value is the address of another variable. Like any variable or constant, you

must declare a pointer before you can work with it. The general form of a pointer variable declaration is

type \*var-name;

Here, the type is the pointer's base type; it must be a valid C++ type and var-name is the name of the pointer variable. The asterisk you used to declare a pointer is the same asterisk that you use for multiplication. However, in this statement, the asterisk is being used to designate a variable as a pointer.

Following are the valid pointer declaration –

```
int *ip; // pointer to an integer
double *dp; // pointer to a double
float *fp; // pointer to a float
char *ch // pointer to character
```

The actual data type of the value of all pointers, whether integer, float, character or otherwise, is the same, a long hexadecimal number that represents a memory address. The only difference between pointers of different data types is the data type of the variable or constant that the pointer points to.

```
#include <iostream>

using namespace std;

int main () {
    int var = 20; // actual variable declaration.
    int *ip; // pointer variable

    ip = &var; // store address of var in pointer variable

    cout << "Value of var variable: ";
    cout << var << endl;

    // print the address stored in ip pointer variable
    cout << "Address stored in ip variable: ";
    cout << ip << endl;

    // access the value at the address available in pointer
    cout << "Value of *ip variable: ";
    cout << *ip << endl;

    return 0;
}
```

## UNIT-III

### Class

The classes are the most important feature of C++ that leads to Object Oriented programming. The class is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.

The variables inside class definition are called as data members and the functions are called member functions.

For example Class of birds, all birds can fly and they all have wings and beaks. So here flying is a behavior and wings and beaks are part of their characteristics. And there are many different birds in this class with different names but they all possess this behavior and characteristics.

Similarly, the class is just a blueprint, which declares and defines characteristics and behavior, namely data members and member functions respectively. And all objects of this class will share these characteristics and behavior.

### Properties of a Class

The class name must start with an uppercase letter (Although this is not mandatory). If the class name is made of more than one word, then the first letter of each word must be in uppercase.

Example, class Study, class CDGI etc

- Classes contain, data members and member functions, and the access of these data members and variable depends on the access specifiers (discussed in next section).
- Class's member functions can be defined inside the class definition or outside the class definition.
- Class in C++ are like structures in C, the only difference being, class defaults to private access control, whereas structure defaults to public.
- All the features of OOPS, revolve around classes in C++. Inheritance, Encapsulation, Abstraction etc.
- Objects of the class hold separate copies of data members. We can create as many objects of a class as we need.
- Classes do possess more characteristics like we can create abstract classes, immutable classes, all this we will study later.

### Objects: -

The class is mere a blueprint or a template. No storage is assigned when we define a class. Objects are instances of the class, which holds the data variables declared in the class and the member functions work on these class objects.

```
#include <iostream>
using namespace std;

class Box {
public:
    double length; // Length of a box
    double breadth; // Breadth of a box
    double height; // Height of a box
};

int main() {
```

```

Box Box1; // Declare Box1 of type Box
Box Box2; // Declare Box2 of type Box
double volume = 0.0; // Store the volume of a box here

// box 1 specification
Box1.height = 5.0;
Box1.length = 6.0;
Box1.breadth = 7.0;

// box 2 specification
Box2.height = 10.0;
Box2.length = 12.0;
Box2.breadth = 13.0;

// volume of box 1
volume = Box1.height * Box1.length * Box1.breadth;
cout << "Volume of Box1 : " << volume << endl;

// volume of box 2
volume = Box2.height * Box2.length * Box2.breadth;
cout << "Volume of Box2 : " << volume << endl;
return 0;
}

```

Each object has different data variables. Objects are initialized using special class functions called **Constructors**.

```

class CDGI
{
int x;
void display(){} //empty function
};

int main()
{
CDGI obj; // Object of class CDGI created
}

```

### **Access Specifier in C++**

Access specifiers in C++ class define the access control rules.

1. public
2. private
3. protected

Access specifiers in the program, are followed by a colon. You can use either one, two or all 3 specifiers in the same class to set different boundaries for different class members.

## Public

Public means all the class members declared under public will be available to everyone. The data members and member functions declared public can be accessed by other classes too. Hence there are chances that they might change them. So, the key members must not be declared public.

```
class PublicAccess
{
public: // public access specifier
int x;      // Data Member Declaration
void display(); // Member Function declaration
}
```

## Private

Private keyword means that no one can access the class members declared private outside that class. If someone tries to access the private member, they will get a compile-time error. By default, class variables and member functions are private.

```
class PrivateAccess
{
private: // private access specifier
int x;      // Data Member Declaration
void display(); // Member Function declaration
}
```

## Protected

Protected, is the last access specifier, and it is like private, it makes class member inaccessible outside the class. But they can be accessed by any subclass of that class. (If class A is inherited by class B, then class B is a subclass of class A).

```
class ProtectedAccess
{
protected: // protected access specifier
int x;      // Data Member Declaration
void display(); // Member Function declaration
}
```

## SCOPE Resolution Operator: -

The :: (scope resolution) operator is used to qualify hidden names so that you can still use them. You can use the unary scope operator if a namespace scope or global scope name is hidden by an explicit declaration of the same name in a block or class.

## Usage of Scope Resolution Operator

### 1) To access a global variable when there is a local variable with same name:

```
// C++ program to show that we can access a global variable
// using scope resolution operator:: when there is a local
// variable with same name
#include<iostream>
```

```
using namespace std;

int x; // Global x

int main()
{
    int x = 10; // Local x
    cout << "Value of global x is " << ::x;
    cout << "\nValue of local x is " << x;
    return 0;
}
```

**2) To define a function outside a class.**

```
// C++ program to show that scope resolution operator:: is used
// to define a function outside a class
#include<iostream>
using namespace std;

class A
{
public:

    // Only declaration
    void fun();
};

// Definition outside class using ::
void A::fun()
{
    cout << "fun() called";
}

int main()
{
    A a;
    a.fun();
    return 0;
}
```

**3) To access a class's static variables.**

**4) In case of multiple Inheritance:**

### Constructor in C++

A class constructor is a special member function of a class that is executed whenever we create new objects of that class.

A constructor will have an exact same name as the class and it does not have any return type at all, not even void. Constructors can be very useful for setting initial values for certain member variables.

### **Special characteristics of Constructors:**

- They should be declared in the public section
- They do not have any return type, not even void
- They get automatically invoked when the objects are created
- They cannot be inherited though derived class can call the base class constructor
- Like other functions, they can have default arguments
- You cannot refer to their address
- Constructors cannot be virtual

```
#include <iostream>

using namespace std;

class Line {
public:
    void setLength( double len );
    double getLength( void );
    Line(); // This is the constructor
private:
    double length;
};

// Member functions definitions including constructor
Line::Line(void) {
    cout << "Object is being created" << endl;
}
void Line::setLength( double len ) {
    length = len;
}
double Line::getLength( void ) {
    return length;
}

// Main function for the program
int main() {
    Line line;

    // set line length
    line.setLength(6.0);
    cout << "Length of line : " << line.getLength() << endl;

    return 0;
}

class A
{
int x;
public:
A(); //Constructor
};
```

Constructors are special class functions which perform initialization of every object. The Compiler calls the Constructor whenever an object is created. Constructors initialize values to object members after storage is allocated to the object.

### Types of Constructors

Constructors are of three types:

- Default Constructor

The default constructor is the constructor which doesn't take any argument. It has no parameter.

- Parametrized Constructor

These are the constructors with a parameter. Using this Constructor, you can provide different values to data members of different objects, by passing the appropriate values as an argument.

- Copy Constructor

These are a special type of Constructors which takes an object as an argument and is used to copy values of data members of one object into another object. We will study copy constructors in detail later.

### Destructors

The destructor is a special class function which destroys the object as soon as the scope of an object ends. The destructor is called automatically by the compiler when the object goes out of scope.

The syntax for destructor is same as that for the constructor, the class name is used for the name of the destructor, with a tilde ~ sign as a prefix to it.

```
class A
```

```
{
```

```
public:
```

```
~A();
```

```
};
```

Destructors will never have any arguments.

### Friend Function: -

A friend function of a class is defined outside that class' scope but it has the right to access all private and protected members of the class. Even though the prototypes for friend functions appear in the class definition, friends are not member functions.

A friend can be a function, function template, or member function, or a class or class template, in which case the entire class and all its members are friends.

To declare a function as a friend of a class, precede the function prototype in the class definition with keyword friend as follows –

```
class Box {
```

```
    double width;
```

```

public:
double length;
friend void printWidth( Box box );
void setWidth( double wid );
};


```

To declare all member functions of class ClassTwo as friends of class ClassOne, place a following declaration in the definition of class ClassOne.

### Inheritance in C++

One of the most important concepts in object-oriented programming is that of inheritance. Inheritance allows us to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time.

When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the base class, and the new class is referred to as the derived class.

The idea of inheritance implements the is-a relationship. For example, mammal IS-A animal, dog IS-A mammal hence dog IS-A animal as well and so on.

### Base and Derived Classes

A class can be derived from more than one classes, which means it can inherit data and functions from multiple base classes. To define a derived class, we use a class derivation list to specify the base class(es). A class derivation list names one or more base classes and has the form –

#### **class derived-class: access-specifier base-class**

Where access-specifier is one of public, protected, or private, and base-class is the name of a previously defined class. If the access-specifier is not used, then it is private by default.

```

#include <iostream>
using namespace std;
// Base class
class Shape {
public:
    void setWidth(int w) {
        width = w;
    }
    void setHeight(int h) {
        height = h;
    }
protected:

```

```

int width;
int height;
};

// Derived class
class Rectangle: public Shape {
public:
    int getArea() {
        return (width * height);
    }
};

int main(void) {
    Rectangle Rect;

    Rect.setWidth(5);
    Rect.setHeight(7);

    // Print the area of the object.
    cout << "Total area: " << Rect.getArea() << endl;

    return 0;
}

```

Summary of Access type in Inheritance using different access specifiers.

Access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

- Constructors, destructors and copy constructors of the base class.
- Overloaded operators of the base class.
- The friend functions of the base class.

#### Type of Inheritance

When deriving a class from a base class, the base class may be inherited through public, protected or private inheritance.

We hardly use protected or private inheritance, but public inheritance is commonly used. While using different type of inheritance, following rules are applied –

**Public Inheritance** – When deriving a class from a public base class, public members of the base class become public members of the derived class and protected members of the base class become protected members of the derived class. A base class's private members are never accessible directly from a derived class but can be accessed through calls to the public and protected members of the base class.

**Protected Inheritance** – When deriving from a protected base class, public and protected members of the base class become protected members of the derived class.

**Private Inheritance** – When deriving from a private base class, public and protected members of the base class become private members of the derived class.

### Polymorphism in C++

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

Real life example of polymorphism, a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee. So, the same person possesses have different behavior in different situations. This is called polymorphism.

Polymorphism is considered as one of the important features of Object Oriented Programming.

In C++ polymorphism is mainly divided into two types:

- Compile time Polymorphism
- Runtime Polymorphism

**Compile time polymorphism:** This type of polymorphism is achieved by function overloading or operator overloading.

**Function Overloading:** When there are multiple functions with the same name but different parameters then these functions are said to be overloaded. Functions can be overloaded by change in number of arguments or/and change in type of arguments

Example

```
#include <iostream.h>
using namespace std;
class CDGI
{
public:
    // function with 1 int parameter
    void func(int x)
    {
        cout << "value of x is " << x << endl;
    }

    // function with same name but 1 double parameter
    void func(double x)
    {
        cout << "value of x is " << x << endl;
    }

    // function with same name and 2 int parameters
    void func(int x, int y)
    {
        cout << "value of x and y is " << x << ", " << y << endl;
    }
};
```

```

int main() {

    CDGI obj1;

    // Which function is called will depend on the parameters passed
    // The first 'func' is called
    obj1.func(7);

    // The second 'func' is called
    obj1.func(9.132);

    // The third 'func' is called
    obj1.func(85,64);
    return 0;
}

```

**Operator Overloading:** C++ also provide an option to overload operators. For example, we can make the operator ('+') for string class to concatenate two strings. We know that this is the addition operator whose task is to add to operands. So a single operator '+' when placed between integer operands, adds them and when placed between string operands, concatenates them.

Example:

```

// CPP program to illustrate
// Operator Overloading
#include<iostream>
using namespace std;

class Complex {
private:
    int real, imag;
public:
    Complex(int r = 0, int i = 0) {real = r; imag = i;}

    // This is automatically called when '+' is used with
    // between two Complex objects
    Complex operator + (Complex const &obj) {
        Complex res;
        res.real = real + obj.real;
        res.imag = imag + obj.imag;
        return res;
    }
    void print() { cout << real << " + " << imag << endl; }
};

int main()
{
    Complex c1(10, 5), c2(2, 4);
    Complex c3 = c1 + c2; // An example call to "operator+"
    c3.print();
}

```

## Virtual Functions

Virtual Function is a function in a base class, which is overridden in the derived class, and which tells the compiler to perform Late Binding on this function.

Virtual Keyword is used to make a member function of the base class Virtual.

- Virtual functions ensure that the correct function is called for an object, regardless of the type of reference (or pointer) used for the function call.
- They are mainly used to achieve Runtime polymorphism
- Functions are declared with a virtual keyword in the base class.
- The resolving of the function call is done at Run-time.

## Rules for Virtual Functions

- They Must be declared in public section of the class.
- Virtual functions cannot be static and cannot be a friend function of another class.
- Virtual functions should be accessed using pointer or reference of the base class type to achieve runtime polymorphism.
- The prototype of virtual functions should be same in the base as well as derived class.
- They are always defined in a base class and overridden in the derived class. It is not mandatory for a derived class to override (or re-define the virtual function), in that case, base class version of the function is used.
- A class may have virtual destructor but it cannot have a virtual constructor.

## Runtime Polymorphism

Runtime polymorphism is achieved only through a pointer (or reference) of base class type. Also, a base class pointer can point to the objects of the base class as well as to the objects of the derived class.

**Late binding(Runtime)** is done in accordance with the content of pointer (i.e. location pointed to by pointer) an Early binding (Compile time) is done according to the type of pointer, since print() function is declared with virtual keyword so it will be binded at run-time (output is print derived class as pointer is pointing to object of derived class ) and show() is non-virtual so it will be binded during compile time(output is show base class as pointer is of base type ).

## Introduction to Data Structure

Data Structure is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. For example, we have data player's name "Virat" and age 26. Here "Virat" is of String data type and 26 is of integer data type.

We can organize this data as a record like Player record. Now we can collect and store player's records in a file or database as a data structure. For example "Dhoni" 30, "Gambhir" 31, "Sehwag" 33

In simple language, Data Structures are structures programmed to store ordered data, so that various operations can be performed on it easily. It represents the knowledge of data to be organized in memory. It should be designed and implemented in such a way that it reduces the complexity and increases the efficiency.

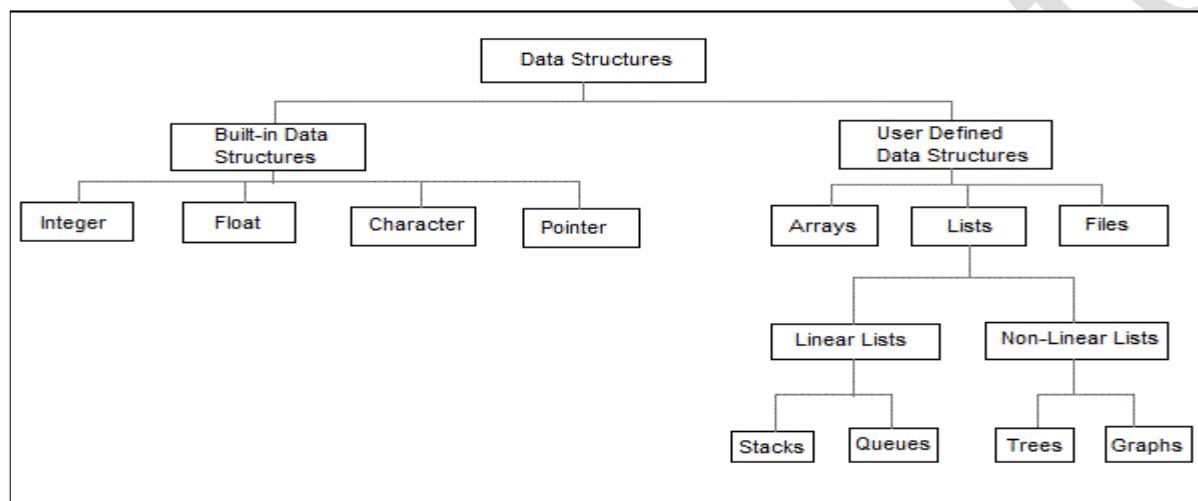
## Basic types of Data Structures

As we have discussed above, anything that can store data can be called as a data structure, hence Integer, Float, Boolean, Char etc, all are data structures. They are known as Primitive Data Structures.

Then we also have some complex Data Structures, which are used to store large and connected data. Some example of Abstract Data Structure is :

- Linked List
- Tree
- Graph
- Stack, Queue etc.

All these data structures allow us to perform different operations on data. We select these data structures based on which type of operation is required. We will consider these data structures in more details in our later lessons.



The data structures can also be classified based on the following characteristics:

Characteristic	Description
Linear	In Linear data structures, the data items are arranged in a linear sequence. Example: <b>Array</b>
Non-Linear	In Non-Linear data structures, the data items are not in sequence. Example: <b>Tree, Graph</b>
Homogeneous	Inhomogeneous data structures, all the elements are of the same type. Example: <b>Array</b>
Non-Homogeneous	In the Non-Homogeneous data structure, the elements may or may not be of the same type. Example: <b>Structures</b>
Static	Static data structures are those whose sizes and structures associated memory locations are fixed, at compile time. Example: <b>Array</b>
Dynamic	Dynamic structures are those which expands or shrinks depending on the program need and its execution. Also, their associated memory locations changes. Example: <b>Linked List created using pointers</b>

Stream TechNotes

## UNIT IV

### Computer Networking: Introduction

Today the world scenario is changing. Data Communication and network have changed the way business and other daily affair works. Now, they rely on computer networks and internetwork. A set of devices often mentioned as nodes connected by media link is called a Network. A node can be a device which is capable of sending or receiving data generated by other nodes on the network like a computer, printer etc. These links connecting the devices are called Communication channels.

Computer network is a telecommunication channel through which we can share our data. It is also called data network. The best example of computer network is Internet. Computer network does not mean a system with control unit and other systems as its slave. It is called a distributed system.

### GOALS OF NETWORKING

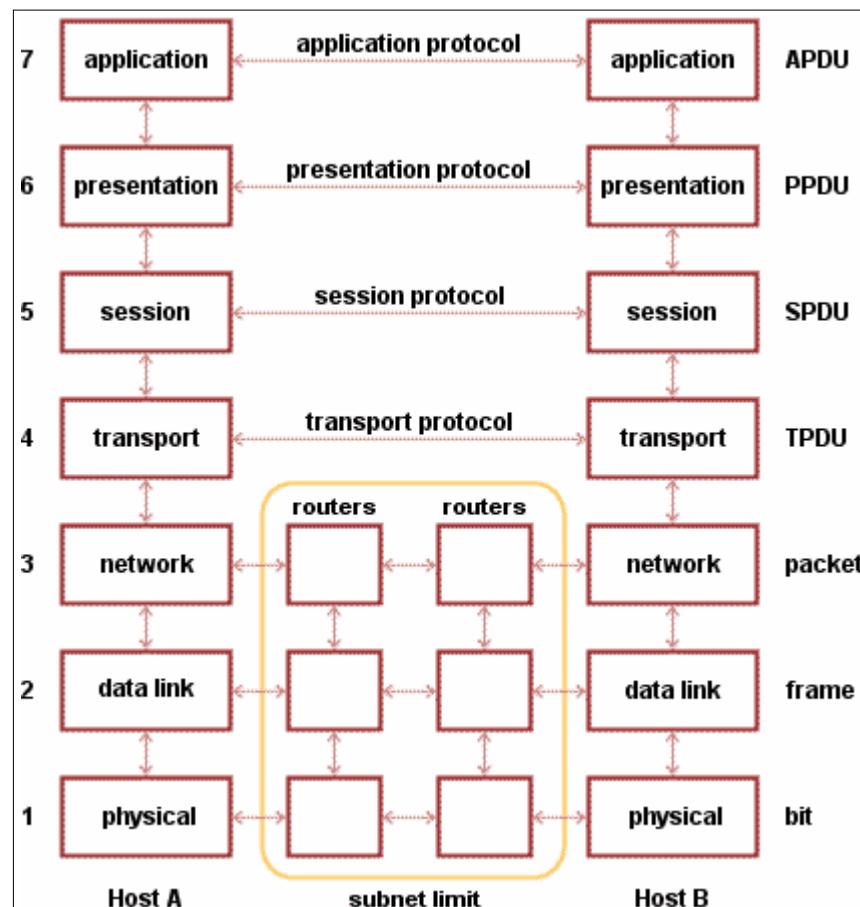
1. Resource and load sharing
2. Programs do not need to run on a single machine
3. Reduced cost
4. Several machines can share printers, tape drives, etc.
5. High reliability
6. If a machine goes down, another can take over
7. Mail and communication

### THE OSI REFERENCE MODEL

OSI (Open Systems Interconnection) is reference model for how applications can communicate over a network. A reference model is a conceptual framework for understanding relationships.

#### Feature of OSI Model:

- Big picture of communication over network is understandable through this OSI model.
- We see how hardware and software work together.
- We can understand new technologies as they are developed.
- Troubleshooting is easier by separate networks.
- Can be used to compare basic functional relationships on different networks.



### Functions of Different Layers:

#### Layer 1: The Physical Layer:

- It is the lowest layer of the OSI Model.
- It activates, maintains and deactivates the physical connection.
- It is responsible for transmission and reception of the unstructured raw data over network.
- Voltages and data rates needed for transmission is defined in the physical layer.
- It converts the digital/analog bits into electrical signal or optical signals.
- Data encoding is also done in this layer.

#### Layer 2: Data Link Layer:

- It translates logical network address into physical address. Concerned with circuit, message or packet switching.
- Routers and gateways operate in the network layer. Mechanism is provided by Network Layer for routing the packets to final destination.
- Connection services are provided including network layer flow control, network layer error control and packet sequence control.
- Breaks larger packets into small packets.

#### Layer 3: The Network Layer:

- It translates logical network address into physical address. Concerned with circuit, message or packet switching.

- Routers and gateways operate in the network layer. Mechanism is provided by Network Layer for routing the packets to final destination.
- Connection services are provided including network layer flow control, network layer error control and packet sequence control.
- Breaks larger packets into small packets.

#### **Layer 4: Transport Layer:**

- Service Point Addressing: Transport Layer header includes service point address which is port address. This layer gets the message to the correct process on the computer unlike Network Layer, which gets each packet to the correct computer.
- Segmentation and Reassembling: A message is divided into segments; each segment contains sequence number, which enables this layer in reassembling the message. Message is reassembled correctly upon arrival at the destination and replaces packets which were lost in transmission.
- Connection Control: It includes 2 types:  
 Connectionless Transport Layer: Each segment is considered as an independent packet and delivered to the transport layer at the destination machine.  
 Connection Oriented Transport Layer: Before delivering packets, connection is made with transport layer at the destination machine.
- Flow Control: In this layer, flow control is performed end to end.
- Error Control: Error Control is performed end to end in this layer to ensure that the complete message arrives at the receiving transport layer without any error. Error Correction is done through retransmission.

#### **Layer 5: The Session Layer:**

- Dialog Control: This layer allows two systems to start communication with each other in half-duplex or full-duplex.
- Synchronization: This layer allows a process to add checkpoints which are considered as synchronization points into stream of data. Example: If a system is sending a file of 800 pages, adding checkpoints after every 50 pages is recommended. This ensures that 50-page unit is successfully received and acknowledged. This is beneficial at the time of crash as if a crash happens at page number 110; there is no need to retransmit 1 to 100 pages.

#### **Layer 6: The Presentation Layer:**

- Translation: Before being transmitted, information in the form of characters and numbers should be changed to bit streams. The presentation layer is responsible for interoperability between encoding methods as different computers use different encoding methods. It translates data between the formats the network requires and the format the computer.
- Encryption: It carries out encryption at the transmitter and decryption at the receiver.
- Compression: It carries out data compression to reduce the bandwidth of the data to be transmitted. The primary role of Data compression is to reduce the number of bits to be transmitted. It is important in transmitting multimedia such as audio, video, text etc.

#### **Layer 7: Application Layer:**

- It is the topmost layer.
- Transferring of files disturbing the results to the user is also done in this layer. Mail services, directory services, network resource etc. are services provided by application layer.
- This layer mainly holds application programs to act upon the received and to be sent data.

#### **Merits of OSI reference model:**

- OSI model distinguishes well between the services, interfaces and protocols.
- Protocols of OSI model are very well hidden.
- Protocols can be replaced by new protocols as technology changes.
- Supports connection oriented services as well as connectionless service.

#### **Demerits of OSI reference model:**

- Model was devised before the invention of protocols.
- Fitting of protocols is tedious task.
- It is just used as a reference model.

### **Internetworking**

Internetworking started as a way to connect disparate types of computer networking technology. Computer network term is used to describe two or more computers that are linked to each other. When two or more computer networks or computer network segments are connected using devices such as a router then it is called as computer internetworking.

Internetworking is a term used by Cisco. Any interconnection among or between public, private, commercial, industrial, or governmental computer networks may also be defined as an internetwork or Internetworking.

#### **Internetworking in detail**

In modern practice, the interconnected computer networks or Internetworking use the Internet Protocol. Two architectural models are commonly used to describe the protocols and methods used in internetworking. The standard reference model for internetworking is Open Systems Interconnection (OSI). Internetworking is implemented in Layer 3 (Network Layer) of this model the most notable example of internetworking is the Internet (capitalized). There are three variants of internetwork or Internetworking, depending on who administers and who participates in them:

- Extranet
- Intranet
- Internet

**Extranet:** An extranet is a network of internetwork or Internetworking that is limited in scope to a single organization or entity but which also has limited connections to the networks of one or more other usually, but not necessarily, trusted organizations or entities. Technically, an extranet may also be categorized as a MAN, WAN, or other type of network, although, by definition, an extranet cannot consist of a single LAN; it must have at least one connection with an external network.

**Intranet:** An intranet is a set of interconnected networks or Internetworking, using the Internet Protocol and uses IP-based tools such as web browsers and ftp tools, that is under the control of a single administrative entity. That administrative entity closes the intranet to the rest of the world, and allows only specific users. Most commonly, an intranet is the internal network of a company or other enterprise. A large intranet will typically have its own web server to provide users with browsable information.

**Internet:** A specific Internetworking, consisting of a worldwide interconnection of governmental, academic, public, and private networks based upon the Advanced Research Projects Agency Network (ARPANET) developed by ARPA of the U.S. Department of Defense also home to the World Wide Web (WWW) and referred to as the 'Internet' with a capital 'I' to distinguish it from other generic internetworks. Participants in the Internet, or their service providers, use IP Addresses obtained from address registries

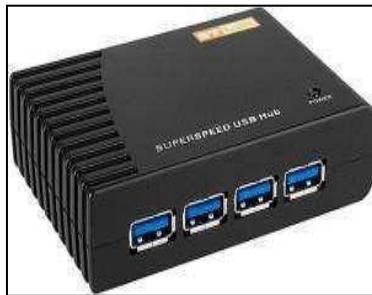
that control assignments.

### Different type of networking devices:

#### Network Hub:

Network Hub is a networking device which is used to connect multiple network hosts. A network hub is also used to do data transfer. The data is transferred in terms of packets on a computer network. So when a host sends a data packet to a network hub, the hub copies the data packet to all of its ports connected to. Like this, all the ports know about the data and the port for whom the packet is intended, claims the packet.

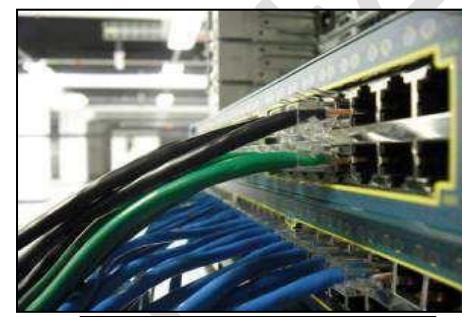
However, because of its working mechanism, a hub is not so secure and safe. Moreover, copying the data packets on all the interfaces or ports makes it slower and more congested which led to the use of network switch.



HUB



Modem



Switch

**Network Switch:** -Like a hub, a switch also works at the layer of LAN (Local Area Network) but you can say that a switch is more intelligent than a hub. While hub just does the work of data forwarding, a switch does 'filter and forwarding' which is a more intelligent way of dealing with the data packets.

So, when a packet is received at one of the interfaces of the switch, it filters the packet and sends only to the interface of the intended receiver. For this purpose, a switch also maintains a CAM (Content Addressable Memory) table and has its own system configuration and memory. CAM table is also called as forwarding table or forwarding information base (FIB).

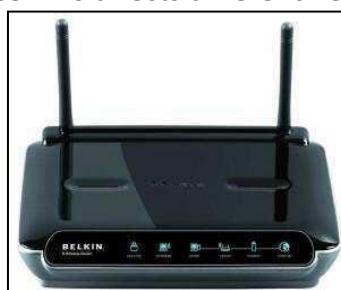
#### Modem:

A modem stands for (Modulator+Demodulator). That means it modulates and demodulates the signal between the digital data of a computer and the analogue signal of a telephone line.

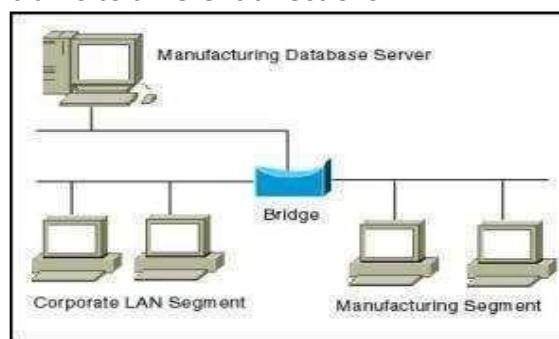
A Modem is somewhat a more interesting network device in our daily life. So, if you have noticed around, you get an internet connection through a wire (there are different types of wires) to your house. This wire is used to carry our internet data outside to the internet world. However, our computer generates binary data or digital data in forms of 1s and 0s and on the other hand, a wire carries an analog signal and that's where a modem comes in.

#### Network Router:

A router is a network device which is responsible for routing traffic from one to another network. These two networks could be a private company network to a public network. You can think of a router as a traffic police who directs different network traffic to different directions.



1WIFI Router



**Bridge:**

If a router connects two different types of networks, then a bridge connects two subnetworks as a part of the same network. You can think of two different labs or two different floors connected by a bridge.

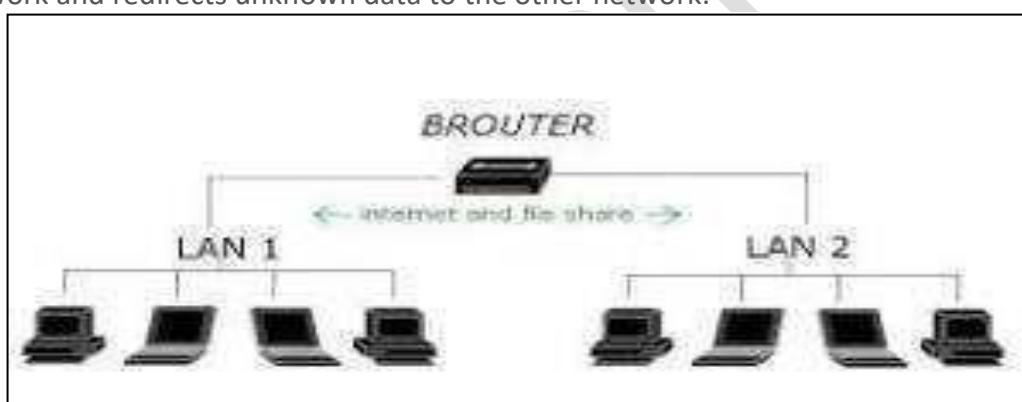
**Repeater:**

A repeater is an electronic device that amplifies the signal it receives. In other terms, you can think of repeater as a device which receives a signal and retransmits it at a higher level or higher power so that the signal can cover longer distances.

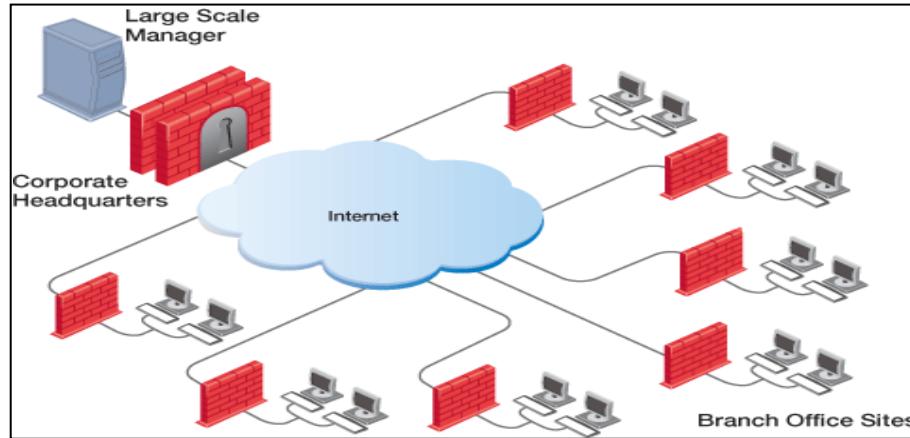
For example, inside a college campus, the hostels might be far away from the main college where the ISP line comes in. If the college authority wants to pull a wire in between the hostels and main campus, they will have to use repeaters if the distance is much because different types of cables have limitations in terms of the distances they can carry the data for.

**Brouters**

Brouters are the combination of both the bridge and routers. They take up the functionality of the both networking devices serving as a bridge when forwarding data between networks, and serving as a router when routing data to individual systems. Brouter functions as a filter that allows some data into the local network and redirects unknown data to the other network.

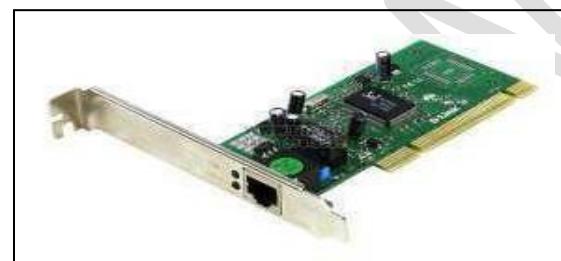
**Gateways**

Gateway is a device which is used to connect multiple networks and passes packets from one packet to the other network. Acting as the 'gateway' between different networking systems or computer programs, a gateway is a device which forms a link between them. It allows the computer programs, either on the same computer or on different computers to share information across the network through protocols. A router is also a gateway, since it interprets data from one network protocol to another.



### Network card

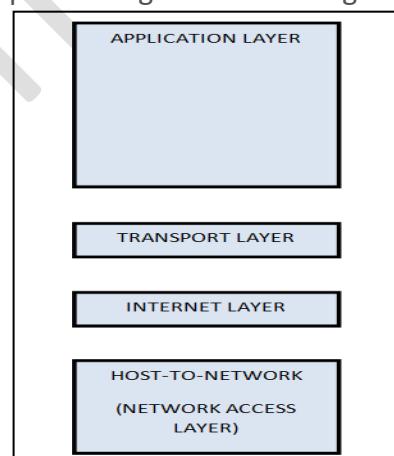
Network cards also known as Network Interface Cards (NICs) are hardware devices that connect a computer with the network. They are installed on the mother board. They are responsible for developing a physical connection between the network and the computer. Computer data is translated into electrical signals sent to the network via Network Interface Cards.



**Network Interface Card**

### The TCP/IP Reference Model

TCP/IP means Transmission Control Protocol and Internet Protocol. It is the network model used in the current Internet architecture as well. Protocols are set of rules which govern every possible communication over a network. These protocols describe the movement of data between the source and destination or the internet. These protocols offer simple naming and addressing schemes.



### Overview of TCP/IP reference model

TCP/IP that is Transmission Control Protocol and Internet Protocol was developed by Department of Defense's Project 'research Agency (ARPA, later DARPA) as a part of a research project of network interconnection to connect remote machines.

The features that stood out during the research, which led to making the TCP/IP reference model were:

- Support for a flexible architecture. Adding more machines to a network was easy.
- The network was robust, and connections remained intact until the source and destination machines were functioning.
- The overall idea was to allow one application on one computer to talk to (send data packets) another application running on different computer.

### Description of different TCP/IP protocols

#### Layer 1: Host-to-network Layer

- Lowest layer of the all.
- Protocol is used to connect to the host, so that the packets can be sent over it.
- Varies from host to host and network to network.

#### Layer 2: Internet layer

- Selection of a packet switching network which is based on a connectionless internetwork layer is called a internet layer.
- It is the layer which holds the whole architecture together.
- It helps the packet to travel independently to the destination.
- Order in which packets are received is different from the way they are sent.
- IP (Internet Protocol) is used in this layer.

#### Layer 3: Transport Layer

- It decides if data transmission should be on parallel path or single path.
- Functions such as multiplexing, segmenting or splitting on the data is done by transport layer.
- The applications can read and write to the transport layer.
- Transport layer adds header information to the data.
- Transport layer breaks the message (data) into small units so that they are handled more efficiently by the network layer.
- Transport layer also arrange the packets to be sent, in sequence.

#### Layer 4: Application Layer

- The TCP/IP specifications described a lot of applications that were at the top of the protocol stack. Some of them were TELNET, FTP, SMTP, DNS etc.
- TELNET is a two-way communication protocol which allows connecting to a remote machine and run applications on it.
- FTP(File Transfer Protocol) is a protocol, that allows File transfer amongst computer users connected over a network. It is reliable, simple and efficient.
- SMTP(Simple Mail Transport Protocol) is a protocol, which is used to transport electronic mail between a source and destination, directed via a route.
- DNS(Domain Name Server) resolves an IP address into a textual address for Hosts connected over a network.

#### Merits of TCP/IP model

- It operated independently.
- It is scalable.

- Client/server architecture.
- Supports a number of routing protocols.
- Can be used to establish a connection between two computers.

### Demerits of TCP/IP

- In this, the transport layer does not guarantee delivery of packets.
- The model cannot be used in any other application.
- Replacing protocol is not easy.
- It has not clearly separated its services, interfaces and protocols.

### Comparison of OSI Reference Model and TCP/IP Reference Model

Following are some major differences between OSI Reference Model and TCP/IP Reference Model.

<b>OSI(Open System Interconnection)</b>	<b>TCP/IP(Transmission Control Protocol / Internet Protocol)</b>
1. OSI is a generic, protocol independent standard, acting as a communication gateway between the network and end user.	1. TCP/IP model is based on standard protocols around which the Internet has developed. It is a communication protocol, which allows connection of hosts over a network.
2. In OSI model the transport layer guarantees the delivery of packets.	2. In TCP/IP model the transport layer does not guarantee delivery of packets. Still the TCP/IP model is more reliable.
3. Follows vertical approach.	3. Follows horizontal approach.
4. OSI model has a separate Presentation layer and Session layer.	4. TCP/IP does not have a separate Presentation layer or Session layer.
5. OSI is a reference model around which the networks are built. Generally, it is used as a guidance tool.	5. TCP/IP model is, in a way implementation of the OSI model.
6. Network layer of OSI model provides both connection oriented and connectionless service.	6. The Network layer in TCP/IP model provides connectionless service.
7. OSI model has a problem of fitting the protocols into the model.	7. TCP/IP model does not fit any protocol
8. Protocols are hidden in OSI model and are easily replaced as the technology changes.	8. In TCP/IP replacing protocol is not easy.
9. OSI model defines services, interfaces and protocols very clearly and makes clear distinction between them. It is protocol independent.	9. In TCP/IP, services, interfaces and protocols are not clearly separated. It is also protocol dependent.
10. It has 7 layers	10. It has 4 layers

## Internet: -

Internet is a network of computers linking many different types of computers all over the world. It is a network of networks sharing a common mechanism for addressing(identifying) computers, and a common set of communication protocols for communications between two computers on the network.

### Applications of Internet

#### 1. Communication

Computer users around the world extensively use the email service on internet to communicate with each other. Pictures, documents and other files are sent as email attachments. Emails can be cc-ed to multiple email addresses

#### 2. Job search

Nowadays, many people search for their jobs online as it is quicker and there is a larger variety of job vacancies present. People can publish resume online for prospective job

#### 3. Online Shopping

The internet has also facilitated the introduction of a new market concept consisting of virtual shops. They provide information about products or services for sale through www servers.

#### 4. Stock market updates

You can sell or buy shares while sitting on computer through internet. Several websites like ndtvprofit.com, moneypore.com, provide information regarding investment

#### 5. Travel

One can use internet to gather information about various tourist place. it can be used for booking Holiday tours, hotels, train, bus, flights and cabs.

#### 6. Research

Research papers are present online which helps in the researcher doing literature review.

#### 7. E-Commerce

E-commerce (electronic commerce or EC) is the buying and selling of goods and services, or the transmitting of funds or data, over an electronic network, primarily the Internet.

#### 8. Social networking

Social networking is the use of internet-based social media programs to make connections with friends, family, classmates, customers and clients.

### WWW (World wide Web)

The World Wide Web has been central to the development of the Information Age and is the primary tool billions of people use to interact on the Internet. Web pages are primarily text documents formatted and annotated with Hypertext Markup Language (HTML). In addition to formatted text, web pages may contain images, video, audio, and software components that are rendered in the user's web browser as coherent pages of multimedia content.

Embedded hyperlinks permit users to navigate between web pages. Multiple web pages with a common theme, a common domain name, or both, make up a website. Website content can largely be provided by the publisher, or interactively where users contribute content or the content depends upon the users or their actions. Websites may be mostly informative, primarily for entertainment, or largely for commercial,

governmental, or non-governmental organizational purposes.

The World Wide Web is a system that makes exchange of data on the Internet easy and efficient. It consists of two basic components:

**The Web Server:** a computer and software ("server" can refer to either) that stores and distributes data to other computers throughout the Internet that request the information.

**The Web Browser:** software running on an individual's ("client") computer that request information from the Web server and displays it in a manner of directed in the data file itself.

### Introduction to E-Commerce

The term "**Electronic commerce**" (or *e-Commerce*) refers to the use of an electronic medium to carry out commercial transactions. Most of the time, it refers to the sale of products via Internet, but the term *eCommerce* also covers purchasing mechanisms via Internet (for B-To-B).

A client who purchases on the Internet is called a **cyber consumer**. *E-Commerce* is not only limited to online sales, but also covers:

- Preparation of estimates online
- Consulting of users
- Provision of an electronic catalog
- Access plan to point of sales
- Real-time management of product availability (stock)
- Online payment
- Delivery tracking
- After-sales service

In certain cases, electronic commerce makes it possible to highly customize products, in particular when the electronic commerce site is linked with the production system of the enterprise (e.g. business cards, customized items such as T-shirts, cups, caps, etc.)

### Computer Security Basics

Computer Security is the process of detecting and preventing any unauthorized use of your laptop/computer. It involves the process of safeguarding against trespassers from using your personal or office based computer resources with malicious intent or for their own gains, or even for gaining any access to them accidentally.

"Malware" is short for malicious software and used as a single term to refer to virus, spy ware, worm etc. Malware is designed to cause damage to a standalone computer or a networked pc. So, wherever a malware term is used it means a program which is designed to damage your computer it may be a virus, worm or Trojan.

#### Worms: -

Worms are malicious programs that make copies of themselves again and again on the local drive, network shares, etc. The only purpose of the worm is to reproduce itself again and again. It doesn't harm any data/file on the computer. Unlike a virus, it does not need to attach itself to an existing program. Worms spread by exploiting vulnerabilities in operating systems

Examples of worm are: - W32.SillyFDC.BBY

Packed.Generic.236

### W32.Troresba

Due to its replication nature it takes a lot of space in the hard drive and consumes more cpu uses which in turn makes the pc too slow also consumes more network bandwidth.

#### **Virus:** -

Virus is a program written to enter to your computer and damage/alter your files/data. A virus might corrupt or delete data on your computer. Viruses can also replicate themselves. A computer Virus is more dangerous than a computer worm as it makes changes or deletes your files while worms only replicates itself without making changes to your files/data.

Examples of virus are: - W32.Sfc!mod

ABAP.Rivpas.A

Accept.3773

Viruses can enter to your computer as an attachment of images, greeting, or audio / video files. Viruses also enters through downloads on the Internet. They can be hidden in a free/trial software's or other files that you download.

Virus is of different types which are as follows.

- 1) File viruses
- 2) Macro viruses
- 3) Master boot record viruses
- 4) Boot sector viruses
- 5) Multipartite viruses
- 6) Polymorphic viruses
- 7) Stealth viruses

**File Virus:** - This type of virus normally infects program files such as .exe, .com, .bat. Once this virus stays in memory it tries to infect all programs that load on to memory.

**Macro Virus:** - These types of virus infect word, excel, PowerPoint, access and other data files. Once infected repairing of these files is very much difficult.

**Master boot record files:** - MBR viruses are memory-resident viruses and copy itself to the first sector of a storage device which is used for partition tables or OS loading programs. A MBR virus will infect this particular area of Storage device instead of normal files. The easiest way to remove a MBR virus is to clean the MBR area,

**Boot sector virus:** - Boot sector virus infects the boot sector of a HDD or FDD. These are also memory resident in nature. As soon as the computer starts it gets infected from the boot sector.

Cleaning this type of virus is very difficult.

**Multipartite virus:** - A hybrid of Boot and Program/file viruses. They infect program files and when the infected program is executed, these viruses infect the boot record. When you boot the computer next time the virus from the boot record loads in memory and then start infecting other program files on disk

**Polymorphic viruses:** - A virus that can encrypt its code in different ways so that it appears differently in each infection. These viruses are more difficult to detect.

**Stealth viruses:** - These types of viruses use different kind of techniques to avoid detection. They either redirect the disk head to read another sector instead of the one in which they reside or they may alter the reading of the infected file's size shown in the directory listing. For example, the Whale virus adds 9216 bytes to an infected file; then the virus subtracts the same number of bytes (9216) from the size given in the directory.

**Trojans:** - A Trojan horse is not a virus. It is a destructive program that looks as a genuine application. Unlike viruses, Trojan horses do not replicate themselves but they can be just as destructive. Trojans also open a backdoor entry to your computer which gives malicious users/programs access to your system, allowing confidential and personal information to be theft.

Example: - JS.Debeski.Trojan

Trojan horses are broken down in classification based on how they infect the systems and the damage caused by them. The seven main types of Trojan horses are:

- 'emote Access Trojans
- Data Sending Trojans
- Destructive Trojans
- Proxy Trojans
- FTP Trojans
- security software disabler Trojans
- denial-of-service attack Trojans

**Adware:** - Generically adware is a software application in which advertising banners are displayed while any program is running. Adware can automatically get downloaded to your system while browsing any website and can be viewed through pop-up windows or through a bar that appears on a computer screen automatically. Adware's are used by companies for marketing purpose.

**Spywares:** - Spyware is a type of program that is installed with or without your permission on your personal computers to collect information about users, their computer or browsing habits tracks each and everything that you do without your knowledge and send it to remote user.

**Spam:** - Spamming is a method of flooding the Internet with copies of the same message. Most spams are commercial advertisements which are sent as an unwanted email to users. Spams are also known as Electronic junk mails or junk newsgroup postings. These spam mails are very annoying as it keeps coming every day and keeps your mailbox full.

**Tracking cookies:** - A cookie is a plain text file that is stored on your computer in a cookies folder and it stores data about your browsing session.

**Misleading applications:** - Misleading applications misguide you about the security status of your computer and shows you that your computer is infected by some malware and you have to download the tool to remove the threat.

**The common types of cybercrimes are: -**

**1. Hacking** – An unauthorized user who attempts to or gains access to an information system is known as hacker. Hacking is a cybercrime even if there is no visible damage to the system, because it is an invasion into the privacy of data.

There are 3 different classes of Hackers.

a) **White Hat Hackers** – They are those hackers who believe that information sharing is good, and that it is their duty to share their expertise by facilitating access to information. However, there are some white hat hackers who are just “joy riding” on computer systems.

b) **Black Hat Hackers** – Black hat hackers cause damage after intrusion. They may steal or modify data or insert viruses or worms which damage the system. They are also known as crackers.

c) **Grey Hat Hackers** – These types of hackers are typically ethical but occasionally they can violate the hacker ethics. They will hack into networks, stand-alone computers and software. Network hackers try to gain unauthorized access to private computer networks just for challenge, curiosity, and distribution of information.

**2. Cyber Stalking** – Cyber stalking involves use of internet to harass someone. The behavior includes false accusations, threats etc. Normally, majority of cyber stalkers are men and the majority of victims are women.

**3. Spamming** – Spamming is sending of unsolicited bulk and commercial messages over the internet. Although irritating to most email users, it is not illegal unless it causes damage such as overloading network and disrupting service to subscribers or creates negative impact on consumer attitudes towards Internet Service Provider.

**4. Cyber Pornography** – With the increasing approach of internet to the people, there is also an increase in the victimization of Women and children for sexual exploitation through internet

**5. Cyber Phishing** – It is a criminally fraudulent process in which cyber-criminal acquires sensitive information such as username, passwords and credit card details by disguising as a trustworthy entity in an electronic communication.

**6. Software Piracy** – It is an illegal reproduction and distribution of software for business or personal use. This is considered to be a type of infringement of copy right and a violation of a license agreement. Since the unauthorized user is not a party to the license agreement it is difficult to find out remedies. There are numerous cases of software piracy. Infact according to one report New Delhi's Nehru market is the Asia's largest market where one can easily find pirated software.

**7. Money Laundering** – Money laundering basically means the moving of illegally acquired cash through financial and other systems so that it appears to be legally acquired. This is possible prior to computer and internet technology and now times electronic transfers have made it easier and more successful.

**8. Password Sniffers** – These are programs that monitor and record the name and password of network users as they log in, jeopardizing security at a site. Whoever installs the sniffer can impersonate an authorized user and log in to access on restricted documents.

**9. Spoofing** – Spoofing is the act of disguising one computer to electronically “look” like another compute, in order to gain access to a system that would be normally is restricted.

**10. Credit Card Fraud** – In U.S.A. half a billion dollars have been lost annually by consumers who have credit cards and calling card numbers. These are stolen from on-line databases. In present world this cybercrime is emerged as a major threat as numerous cases had been filed in almost every major developed and developing country.

**13. Web Jacking** – The term refers to forceful taking of control of a web site by cracking the password.

**11. Cyber terrorism** – The use of computer resources to intimidate or coerce government, the civilian population or any segment thereof in furtherance of political or social objectives is called cyber terrorism. Individuals and groups quite often try to exploit anonymous character of the internet to threaten governments and terrorize the citizens of the country

**Cyber defamation** is not a specific criminal offense, misdemeanor or tort, but rather defamation or slander conducted via digital media, usually through the Internet.

Penalties for "cyber defamation" vary from country to country, but the fundamental rights covered in the UN Declaration of Human Rights and European Union Fundamental Human Rights.

Stopping or addressing defamation can be difficult. If the person has no serious grudge, then a cease and desist letter may stop the behavior and get the statements removed from the Internet. On the other hand, if the person is acting out of spite, it may be necessary to file a report with the police depending on local law.

**Pharming** is a cyber-attack intended to redirect a website's traffic to another, fake site. Pharming can be conducted either by changing the hosts file on a victim's computer or by exploitation of a vulnerability in DNS server software. DNS servers are computers responsible for resolving Internet names into their real IP addresses. Compromised DNS servers are sometimes referred to as "poisoned". Pharming requires unprotected access to target a computer, such as altering a customer's home computer, rather than a corporate business server.

### Firewall

A firewall is a network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules.

Firewalls have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet. A firewall can be hardware, software, or both.

**Packet filtering:** The system examines each packet entering or leaving the network and accepts or rejects it based on user-defined rules. Packet filtering is fairly effective and transparent to users, but it is difficult to configure. In addition, it is susceptible to IP spoofing.

**Circuit-level gateway implementation:** This process applies security mechanisms when a TCP or UDP connection is established. Once the connection has been made, packets can flow between the hosts without further checking.

**Acting as a proxy server:** A proxy server is a type of gateway that hides the true network address of the computer(s) connecting through it. A proxy server connects to the Internet, makes the requests for pages, connections to servers, etc., and receives the data on behalf of the computer(s) behind it.

**Web application firewall:** A web application firewall is a hardware appliance, server plug-in, or some other software filter that applies a set of rules to a HTTP conversation. Such rules are generally customized to the application so that many attacks can be identified and blocked.

### Computer Ethics & Good Practices: -

Ethics deals with placing a "value" on acts according to whether they are "good" or "bad". Every society has its rules about whether certain acts are ethical or not. These rules have been established because of consensus in society and are often written into laws.

The Ten Commandments of computer ethics have been defined by the Computer Ethics Institute. Here is our interpretation of them:

**1) Do not use a computer to harm other people:** If it is unethical to harm people by making a bomb, for example, it is equally bad to write a program that handles the timing of the bomb. Or, to put it more simply, if it is bad to steal and destroy other people's books and notebooks, it is equally bad to access and destroy their files.

**2) Do not interfere with other people's computer work:** Computer viruses are small programs that disrupt other people's computer work by destroying their files, taking huge amounts of computer time or memory, or by simply displaying annoying messages. Generating and consciously spreading computer viruses are unethical.

**3) Do not snoop around in other people's files:** Reading other people's e-mail messages are as bad as opening and reading their letters: This is invading their privacy. Obtaining other people's non-public files should be judged the same way as breaking into their rooms and stealing their documents. Text documents on the Internet may be protected by encryption.

**4) Do not use a computer to steal:** Using a computer to break into the accounts of a company or a bank and transferring money should be judged the same way as robbery. It is illegal and there are strict laws against it.

**5) Do not use a computer to bear false witness:** The Internet can spread untruth as fast as it can spread the truth. Putting out false "information" to the world is bad. For instance, spreading false rumors about a person or false propaganda about historical events is wrong.

**6) Do not use or copy software for which you have not paid:** Software is an intellectual product. In that way, it is like a book: Obtaining illegal copies of copyrighted software is as bad as photocopying a copyrighted book. There are laws against both. Information about the copyright owner can be embedded by a process called watermarking into pictures in the digital format.

### **Introduction to Cyber Law**

In a Simple way, we can say that cybercrime is unlawful acts wherein the computer is either a tool or a target or both. Cybercrimes can involve criminal activities that are traditional in nature, such as theft, fraud, forgery, defamation, and mischief, all of which are subject to the Indian Penal Code. The abuse of computers has also given birth to a gamut of new age crimes that are addressed by the Information Technology Act, 2000.

#### **We can categorize Cyber crimes in two ways**

- The Computer as a Target: -using a computer to attack other computers.

e.g. Hacking, Virus/Worm attacks, DOS attack etc.

- The computer as a weapon: -using a computer to commit real-world crimes.

e.g. Cyber Terrorism, IPR violations, Credit card frauds, Pornography etc.

Cyberlaw (also referred to as cyberlaw) is a term used to describe the legal issues related to using of communications technology, particularly "cyberspace", i.e. the Internet.

### **Cyberlaw in India**

When the Internet was developed, the founding fathers of Internet hardly had any inclination that Internet could transform itself into an all-pervading revolution which could be misused for criminal activities and which required regulation. Today, there are many disturbing things happening in cyberspace. Due to the anonymous nature of the Internet, it is possible to engage in a variety of criminal activities with impunity and people with intelligence, have been grossly misusing this aspect of the Internet to perpetuate criminal activities in cyberspace. Hence the need for Cyber laws in India.

### **Importance of Cyberlaw**

Cyberlaw is important because it touches almost all aspects of transactions and activities on and concerning the Internet, the World Wide Web, and Cyberspace. Initially, it may seem that Cyber laws is a very technical field and that it does not have any bearing on most activities in Cyberspace. But the actual truth is that nothing could be further than the truth. Whether we realize it or not, every action and every reaction in Cyberspace has some legal and Cyber legal perspectives.

### **Advantages of Cyber Laws**

The IT Act 2000 attempts to change outdated laws and provides ways to deal with cybercrimes. We need such laws so that people can perform purchase transactions over the Net through credit cards without fear

of misuse. The Act offers the much-needed legal framework so that information is not denied legal effect, validity or enforceability, solely because it is in the form of electronic records.

## Internet Frauds

Internet fraud is a type of fraud which makes use of the Internet. This type of fraud varies greatly and appears in many forms. It ranges from E-mail spam to online scams. Internet fraud can occur even if partly based on the use of internet services and is mostly or completely based on the use of the internet.

### The main types of internet fraud:

- Stolen credit cards

Credit Card fraud across the internet is one of the more common examples of this type of crime. Some people fall prey to this type of scam because they are careless whilst others are duped by clever phishing schemes.

- Emails

Used as intended, email is a great means of communication that can allow messages to be sent to huge numbers of people at virtually no cost. Unfortunately, this means that it is also an ideal medium for scam artists.

- Lotteries

Fake lottery scams will try to persuade you that you've won a huge amount of money in an online draw. People behind this fraud then try to trick you into revealing your personal information as you try to collect your winnings.

- Fake auctions

Buying and selling goods through internet auction sites is an extremely popular pastime for some, and a great means of doing business for others. Unfortunately, scam artists have seen the potential of infiltrating online auction sites. Internet auction fraud is one of the most common rip-offs on the net today.

- Untrustworthy Websites

A slightly newer form of internet fraud is the fake website. Cybercriminals have begun mimicking established websites and then tricking visitors into interacting with them as if they were the real deal.

## Good Computer Security Habits

### 1. Create strong passwords

Passwords are usually the first, and sometimes only, protection against unauthorized access. They are the keys to your online kingdom, so keep these guidelines in mind.

Do not use your name, common phrases or words or acronyms that can be found in the dictionary—including foreign languages.

### 2. Lock your computer screen

You never know who might use your computer when you're not around, so it's important to lock your screen to prevent unauthorized access. In the office, a co-worker, guest or a service provider might view or use your unattended computer. This is an easy way for private information to become public.

### 3. Secure mobile devices from loss

While mobile devices such as smartphones, tablets, and laptops are valued for their portability, this convenience can become a security risk.

It's easy to lose or misplace these devices, so be sure to:

- Make a list of phone numbers and email addresses to report stolen or lost devices. Use a hardware cable lock for your laptop, or store it in a locked drawer.
- Keep smartphones and tablets with you when in public

- Never put devices in your checked baggage when traveling

#### **4. Protect data on mobile devices and removable media**

Mobile devices and removable media, such as USB drives, enable us to easily share and transport information but can lead to the loss or misuse of data.

#### **5. Identify URLs before clicking**

Simply stated: think before you click. A malicious website that looks legitimate is a common method used by criminals. However, verifying the real destination is easy—just place your cursor over the displayed URL, and the true destination will reveal itself with a small pop-up. Don't click if it looks suspicious.

#### **6. Use public Wi-Fi safely**

Public Wi-Fi is riskier than corporate or home Wi-Fi because you can't determine its setup and security features. So, take extra precautions when using it.

- Do not access sensitive personal accounts, such as financial accounts
- Ensure websites use HTTPS and display a lock icon
- Watch out for "shoulder surfing" from people and security cameras
- Never use a public computer, such as one in a hotel lobby, to access personal information
- Use only for general web browsing, e.g., weather forecasts and restaurant reviews.

#### **7. Think before you post to social media**

Social media provides a convenient, fun way to stay in touch with friends and family. But be cautious about what you post. Understand both personal and business risks, and take the following precautions:

- Always comply with your company's rules for business conduct
- Ask friends and family to keep your personal information private, including relationships
- Be cautious about participating in games and surveys or clicking on links suggested by others
- Review and update your social media privacy and security settings often.

## UNIT V

### **Database-management system**

A database management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise.

The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. Database systems are designed to manage large bodies of information. Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access.

### **Purpose of Database Approach**

#### **File-Based Approach**

The traditional file-oriented approach to information processing has for each application a separate master file and its own set of personal files. COBOL language supported these file-oriented applications. It was used for developing applications such as of payroll, inventory, and financial accounting. However, in general an organization needs flow of information across these applications also and this requires sharing of data, which is very difficult to implement in the traditional file approach. In addition, a major limitation of file-based approach is that the programs are dependent on the files and the files are dependent upon the programs.

These file-based approaches, which came into being as the first commercial applications of computers, suffered from the following significant disadvantages:

#### **The disadvantages of file system processing**

##### **(i) Data Redundancy:**

- o Since files and applications are created by the different programmer of various departments over a long period of time, it might lead to several problems:
- o Inconsistency in data format
- o The same information may be kept in several different places (files).
- o Data inconsistency which means various copies of the same data are conflicting; waste storage space and duplication of effort

##### **(ii) Data Isolation**

- o It is difficult for a new application to retrieve the appropriate data which might be stored in various files.

##### **(iii) Integrity problems**

- o Data values must satisfy certain consistency constraints which are specified in the application programs.
- o It is difficult to add change the programs to enforce new constraint

##### **(iv) Security problems**

- o There are constraint regarding accessing privileges
- o Application is added to the system in an ad-hoc manner so it is difficult to enforce those constraints

##### **(v) Concurrent – access anomalies**

- o Data may be accessed by many applications that have not been coordinated previously so it is not easy to provide a strategy to support multiple users to update data simultaneously.

### **Database Approach**

#### **Fundamental Concepts**

The database is a shared collection of related data which will be used to support the activities of the organization. The database can be viewed as a repository of data that is defined once and then is accessed by various users.

### **A database has the following properties:**

- It is a representation of some aspect of the real world; or perhaps, a collection of data elements (facts) representing real-world information.
- The database is logically coherent and internally consistent.
- The database is designed, built, and populated with data for a specific purpose.

### **Characteristics of Database approach**

There are many characteristics that distinguish the database approach with the file-based approach. In this section, we describe in detail some of those important characteristics.

- **Self-Describing Nature of a Database System:** Database System contains not only the database itself but also the descriptions of data structure and constraints (meta-data). This information is used by the DBMS software or database users if needed.
- **Insulation between Program and Data:** In the file-based system, the structure of the data files is defined in the application programs so if the user wants to change the structure of a file, all the programs access to that file might need to be changed.
- **Support multiple views of data:** A view is a subset of the database which is defined and dedicated for users of the system. Multiple users in the system might have different views of the system. Each view might contain only the interested data of a user or a group of users.
- **Sharing of data and Multi-user system:** A multi-user database system must allow multiple users to access the database at the same time. As a result, the multi-user DBMS must have concurrency control strategies to ensure that several users tries to access the same data item at a time do so in the manner so that the data always be correct.

### **Benefits of Database Approach**

#### **-To Control Data Redundancy**

- o In the Database approach, ideally, each data item is stored in only one place in the database
- o However, in some cases redundancy still exists to improve system performance, but such redundancy is controlled and kept to minimum

#### **- Data Sharing**

- o The integration of the whole data in an organization leads to the ability to produce more information from a given amount of data

#### **- Enforcing Integrity Constraints**

- o DBMSs should provide capabilities to define and enforce certain constraints such as data type, data uniqueness.

#### **- Restricting Unauthorized Access**

- o Not all users of the system have the same access privileges.
- o DBMSs should provide a security subsystem to create and control the user accounts.

#### **- Data Independence**

- o The system data descriptions are separated from the application programs.
- o Changes to the data structure are handled by the DBMS and not embedded in the program.

#### **- Transaction Processing**

- o The DBMS must include concurrency control subsystem to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.

#### **- Providing multiple views of data**

- o A view may be a subset of the database. Various users may have different views of the database itself.
- o Users may not need to be aware of how and where the data they refer to is stored

#### **- Providing backup and recovery facilities**

- o If the computer system fails in the middle of a complex update program, the recovery subsystem is responsible for making sure that the database is restored to the stage it was in before the program started executing.

## Data Models in DBMS

A database model shows the logical structure of a database, including the relationships and constraints that determine how data can be stored and accessed. Individual database models are designed based on the rules and concepts of whichever broader data model the designers adopt. Most data models can be represented by an accompanying database diagram.

### Entity-Relationship Model

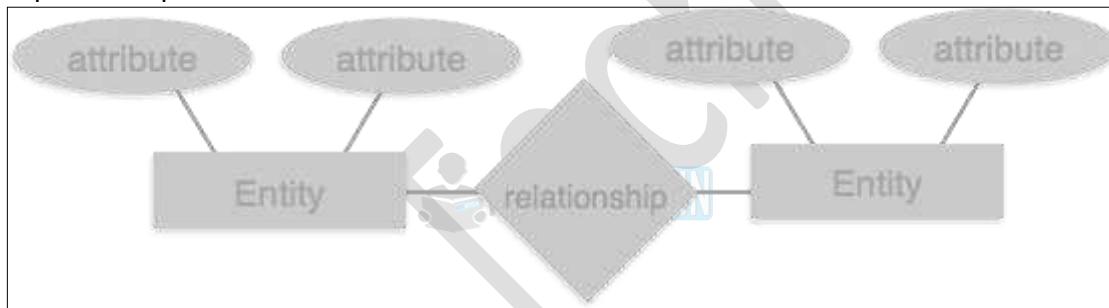
Entity-Relationship (ER) Model is based on the notion of real-world entities and relationships among them. While formulating real-world scenario into the database model, the ER Model creates entity set, relationship set, general attributes and constraints.

ER Model is best used for the conceptual design of a database.

ER Model is based on –

- **Entities** and their *attributes*.
- **Relationships** among entities.

These concepts are explained below.



**E-R Diagram**

- **Entity** – An entity in an ER Model is a real-world entity having properties called **attributes**. Every **attribute** is defined by its set of values called **domain**. For example, in a school database, a student is considered as an entity. Student has various attributes like name, age, class, etc.
- **Relationship** – The logical association among entities is called **relationship**. Relationships are mapped with entities in various ways. Mapping cardinalities define the number of association between two entities.

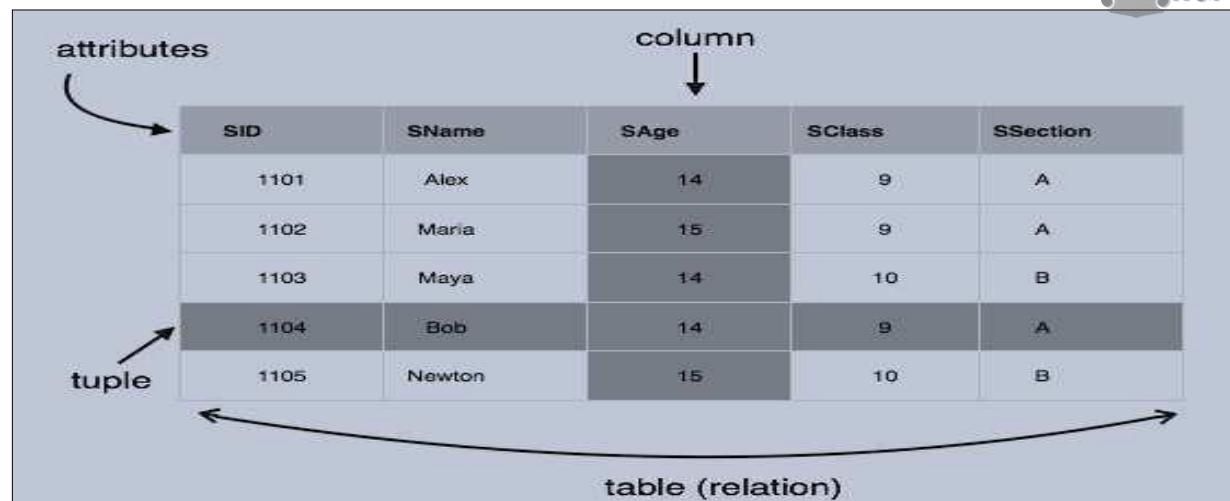
Mapping cardinalities –

- one to one
- one to many
- many to one
- many to many

### Relational Model

The most popular data model in DBMS is the Relational Model. It is more scientific a model than others.

This model is based on first-order predicate logic and defines a table as an **n-ary relation**.



### Relational Database Model

The main highlights of this model are –

- Data is stored in tables called **relations**.
- Relations can be normalized.
- In normalized relations, values saved are atomic values.
- Each row in a relation contains a unique value.
- Each column in a relation contains values from the same domain.
- 

### Hierarchical Model

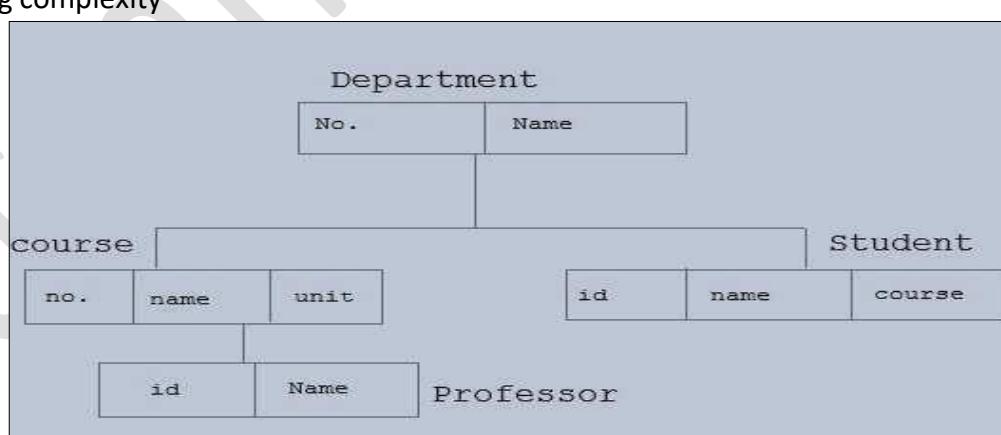
In this model, each entity has only one parent but can have several children. At the top of the hierarchy, there is only one entity which is called **Root**.

Advantages

- Simplicity
- Data Security and Data Integrity
- Efficiency

Disadvantages

- Implementation Complexity
- Lack of structural independence
- Programming complexity



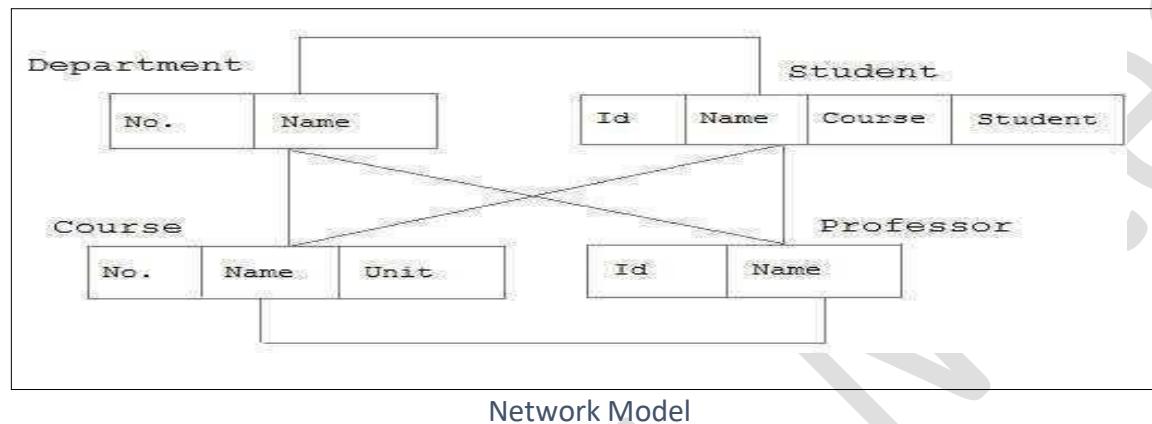
Hierarchical Model

### Network Model

In the network model, entities are organized in a graph, in which some entities can be accessed through several paths.

Advantages

- Conceptual Simplicity
  - Ease of data access
  - Data Integrity and capability to handle more relationship types
  - Data independence
  - Database standards
- Disadvantages
- System complexity
  - Absence of structural independence

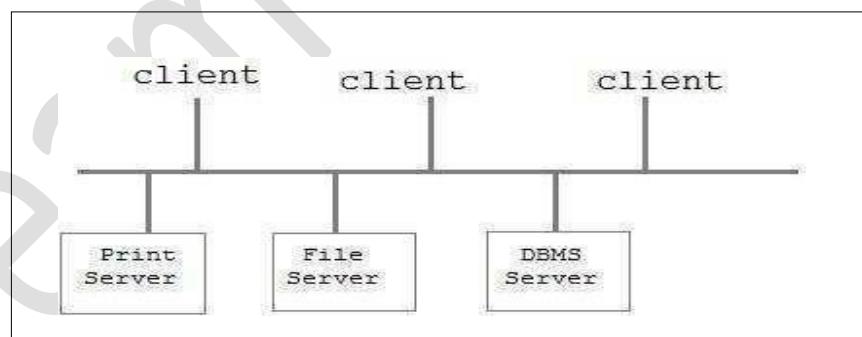


## Database Architecture

Database architecture is logically divided into two types.

1. Logical two-tier Client / Server architecture
2. Logical three-tier Client / Server architecture

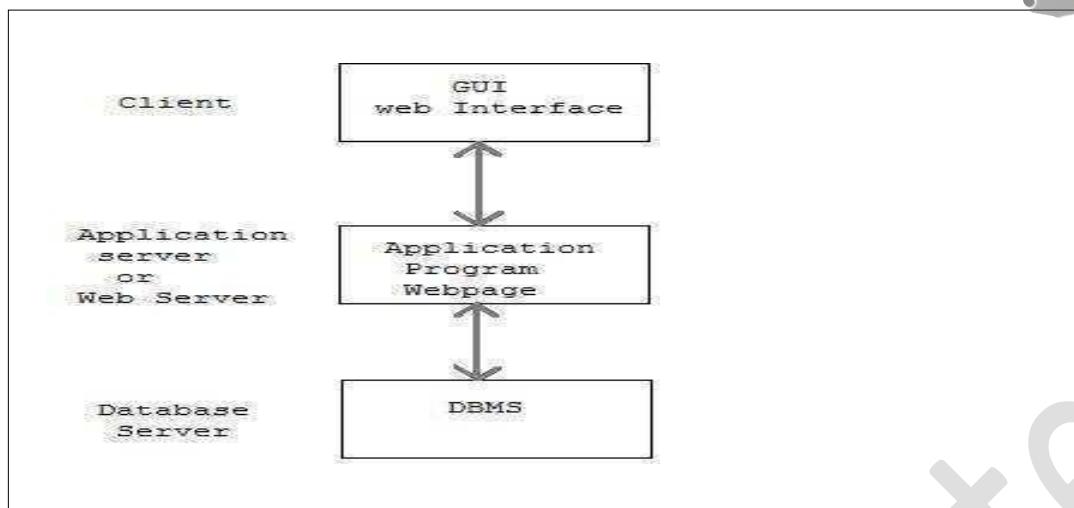
### Two-tier Client / Server Architecture



Two-Tier Architecture

Two-tier Client / Server architecture is used for User Interface program and Application Programs that run on the client side. An interface called ODBC(Open Database Connectivity) provides an API that allows client-side program to call the DBMS. Most DBMS vendors provide ODBC drivers. A client program may connect to several DBMS's. In this architecture some variation of the client is also possible for example in some DBMS's more functionality is transferred to the client including data dictionary, optimization etc. Such clients are called **Data server**.

### Three-tier Client / Server Architecture

**Three-tier Architecture**

### Three Tier Architecture

Three-tier Client / Server database architecture is

commonly used architecture for web applications. An intermediate layer called **Application server** or Web Server stores the web connectivity software and the business logic(constraints) part of application used to access the right amount of data from the database server. This layer acts like a medium for sending partially processed data between the database server and the client.

### Data Independence

A major objective for three-level architecture is to provide data independence, which means that upper levels are unaffected by changes in lower levels.

There are two kinds of data independence:

- Logical data independence
- Physical data independence

### Logical Data Independence

Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schemas. The change would be absorbed by the mapping between the external and conceptual levels. Logical data independence also insulates application programs from operations such as combining two records into one or splitting an existing record into two or more records. This would require a. change in the external/conceptual mapping to leave the external view unchanged.

### Physical Data Independence

Physical data independence indicates that the physical storage structures or devices could be changed without affecting conceptual schema. The change would be absorbed by the mapping between the conceptual and internal levels.

The Logical data independence is difficult to achieve than physical data independence as it requires the flexibility in the design of database and programmer should foresee the future requirements or modifications in the design.

### DBA (Data Base Administrator)

## Database Administrator

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA). The functions of a DBA include:

- Schema definition. The DBA creates the original database schema by executing a set of data definition statements in the DDL.
- Storage structure and access method definition.
- Schema and physical-organization modification. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.
- Granting of authorization for data access. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access.

Routine maintenance. Examples of the database administrator's routine maintenance activities are:

- Periodically backing up the database, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required.
- Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users.

## Data Dictionary

A metadata (also called the data dictionary) is the data about the data. It is the self-describing nature of the database that provides program-data independence. It is also called as the System Catalog. It holds the following information about each data element in the databases, it normally includes:

- Name
- Type
- Range of values
- Source
- Access authorization
- Indicates which application programs use the data so that, when a change in a data structure is contemplated, a list of the affected programs can be generated.

The data dictionary is used to control the database operation, data integrity, and accuracy. Metadata is used by developers to develop the programs, queries, controls, and procedures to manage and manipulate the data.

## Active and Passive Data Dictionaries

The data dictionary may be either active or passive. An active data dictionary (also called integrated data dictionary) is managed automatically by the database management software. Consistent with the current structure and definition of the database. Most of the relational database management systems contain active data dictionaries that can be derived from their system catalog.

The passive data dictionary (also called non-integrated data dictionary) is the one used only for documentation purposes. Data about fields, files, people and so on, in the data processing environment, are entered the dictionary and cross-referenced.

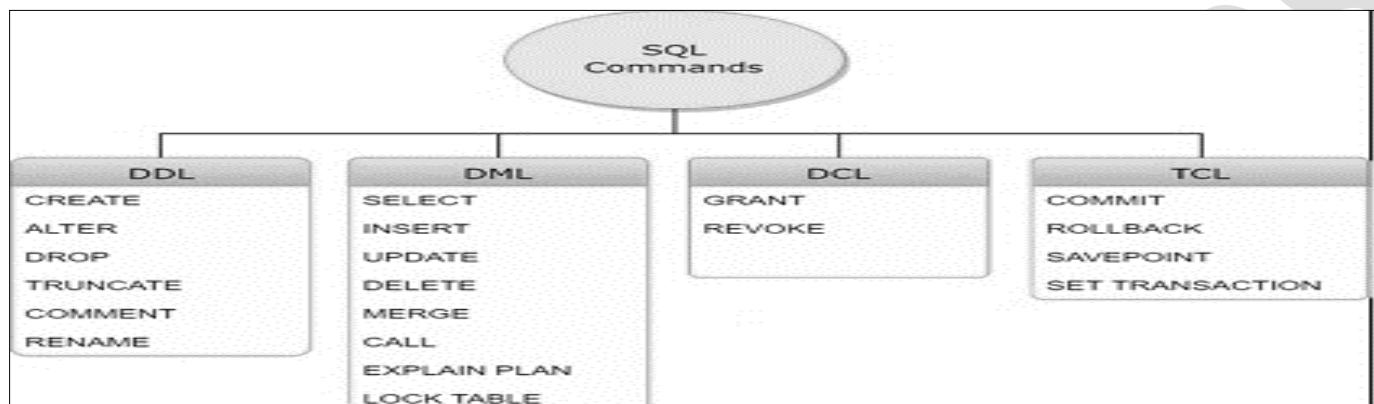
## Primary Key

A primary key is a special relational database table column (or combination of columns) designated to uniquely identify all table records.

A primary key's main features are:

- It must contain a unique value for each row of data.
- It cannot contain null values.
- A primary key is either an existing table column or a column that is specifically generated by the database according to a defined sequence.

## SQL Commands in DBMS



### DDL

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

**CREATE** – to create database and its objects like (table, index, views, stored procedure, function, and triggers)

**ALTER** – alters the structure of the existing database

**DROP** – delete objects from the database

**TRUNCATE** – remove all records from a table, including all spaces allocated for the records are removed

**COMMENT** – add comments to the data dictionary

**RENAME** – rename an object

### DML

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE etc, and it is used to store, modify, retrieve, delete and update data in the database.

**SELECT** – retrieve data from a database

**INSERT** – insert data into a table

**UPDATE** – updates existing data within a table

**DELETE** – Delete all records from a database table

**MERGE** – UPSERT operation (insert or update)

**CALL** – call a PL/SQL or Java subprogram

**EXPLAIN PLAN** – interpretation of the data access path

**LOCK TABLE** – concurrency Control

### Cloud Computing: -

A cloud is simply a centralized technology platform which provides specific IT services to a selected range of users, offering the ability to log in from anywhere, ideally from any device and over any connection, including the Internet.

Intercept IT believes that a true cloud computing service is one which removes the traditional barriers which exist between software applications, data, and devices. In other words, it is the nirvana of computing from a user's perspective, no need to worry about location, device, or type of connection, all the data and the software applications required by the user are fully available and the experience remains consistent. The highest standards of data protection must be a given, whereby users do not have to think about protecting the integrity of the data they use and store.

### **Characteristics of Cloud Computing as per NIST**

Cloud technology is in the news quite often these days, but it still seems to be mysterious and confusing to the non-techie crowd. Cloud options are enticing various industries across the board, which is why it's important to know its essential characteristics as a software offering. Here are the five main characteristics that cloud computing offers businesses today.

#### **1. On-demand capabilities:**

A business will secure cloud-hosting services through a cloud host provider which could be your usual software vendor. You have access to your services and you have the power to change cloud services through an online control panel or directly with the provider.

#### **2. Broad network access:**

Your team can access business management solutions using their smart phones, tablets, laptops, and office computers. They can use these devices wherever they are located with a simple online access point.

#### **3. Resource pooling:**

The cloud enables your employees to enter and use data within the business management software hosted in the cloud at the same time, from any location, and at any time.

#### **4. Rapid elasticity:**

If anything, the cloud is flexible and scalable to suit your immediate business needs. You can quickly and easily add or remove users, software features, and other resources.

#### **5. Measured service:**

Going back to the affordable nature of the cloud, you only pay for what you use. You and your cloud provider can measure storage levels, processing, bandwidth, and the number of user accounts and you are billed appropriately.

### **Cloud Computing Reference Model**

- The NIST Cloud Computing Reference Architecture consists of five major actors. Each actor plays a role and performs a set of activities and functions. The reference architecture is presented as successive diagrams in increasing level of detail.
- Among the five actors, cloud brokers are optional, as cloud consumers may obtain service directly from a cloud provider.

#### **1. Cloud Consumer:**

Person or organization that maintains a business relationship with, and uses service from, Cloud Providers.

#### **2. Cloud Provider:**

A person, organization or entity responsible for making a service available to Cloud Consumers.

### 3. Cloud Auditor:

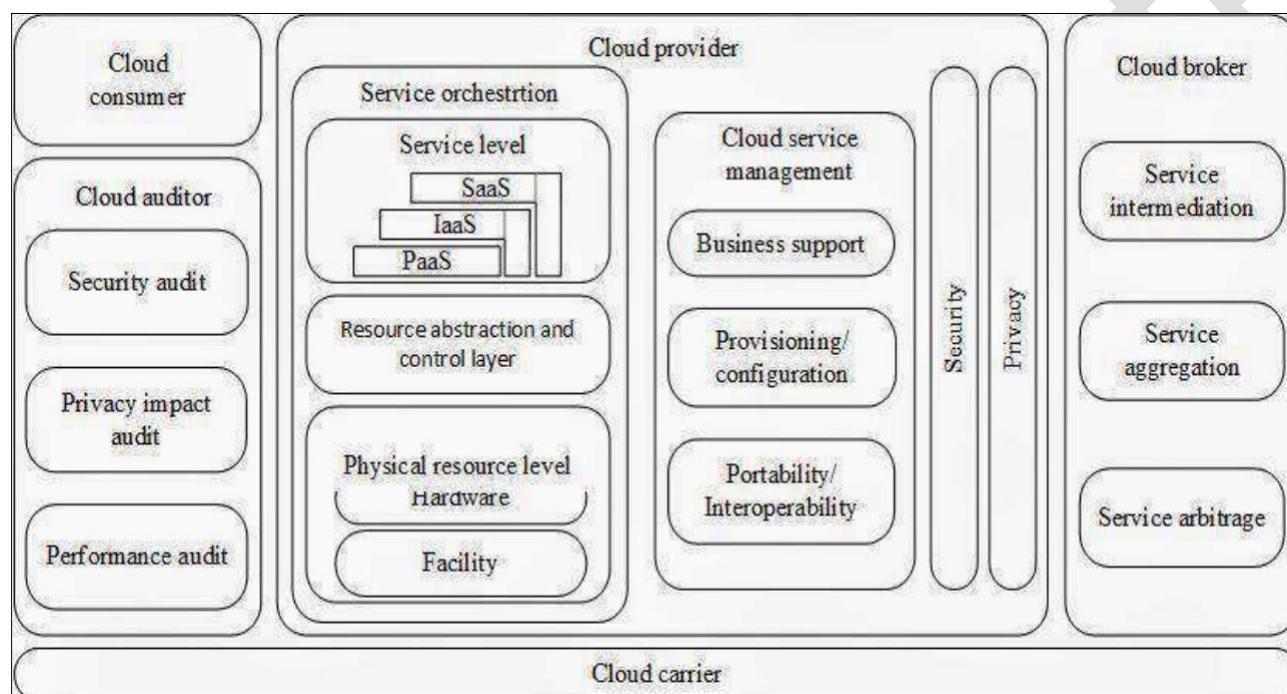
A party that can conduct an independent assessment of cloud services, information system operations, performance and security of the cloud implementation.

### 4. Cloud Broker:

An entity manages the use, performance, and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.

### 5. Cloud Carrier:

The intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.



## Cloud computing service & deployment models

According to National Institute of Standards and Technology (NIST), Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

**Software as a Service (SaaS).** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

**Platform as a Service (PaaS).** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

**Infrastructure as a Service (IaaS).** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer can deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and

deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

### Deployment Model: -

**Private cloud.** The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

**Community cloud.** The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

**Public cloud.** The cloud infrastructure is provisioned for open use by the public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

**Hybrid cloud.** The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

### Advantages of Cloud Computing

1. **Usability:** All cloud storage services reviewed in this topic have desktop folders for Mac's and PCs. This allows users to drag and drop files between the cloud storage and their local storage.
2. **Bandwidth:** You can avoid emailing files to individuals and instead send a web link to recipients through your email.
3. **Accessibility:** Stored files can be accessed from anywhere via an Internet connection.
4. **Disaster Recovery:** It is highly recommended that businesses have an emergency backup plan ready in the case of an emergency. Cloud storage can be used as a backup plan by businesses by providing a second copy of important files. These files are stored at a remote location and can be accessed through an internet connection.
5. **Cost Savings:** Businesses and organizations can often reduce annual operating costs by using cloud storage; cloud storage costs about 3 cents per gigabyte to store data internally. Users can see additional cost savings because it does not require internal power to store information remotely.

### Disadvantages of Cloud Computing

1. **Usability:** Be careful when using drag/drop to move a document into the cloud storage folder. This will permanently move your document from its original folder to the cloud storage location.
2. **Bandwidth:** Several cloud storage services have a specific bandwidth allowance. If an organization surpasses the given allowance, the additional charges could be significant. However, some providers allow unlimited bandwidth. This is a factor that companies should consider when looking at a cloud storage provider.
3. **Accessibility:** If you have no internet connection, you have no access to your data.
4. **Data Security:** There are concerns about the safety and privacy of important data stored remotely. The possibility of private data commingling with other organizations makes some businesses uneasy.
5. **Software:** If you want to be able to manipulate your files locally through multiple devices, you'll need to download the service on all devices.

References: -

1. <http://www.sitesbay.com/cpp/cpp-function>
2. [https://www.tutorialspoint.com/cplusplus/cpp\\_pointers.htm](https://www.tutorialspoint.com/cplusplus/cpp_pointers.htm)
3. <http://www.cplusplus.com/doc/tutorial/>
4. <http://www.studytonight.com/cpp/class-and-objects.php>

Stream Tech Notes