Computer Fundamentals and Office Tools
UNIT - II

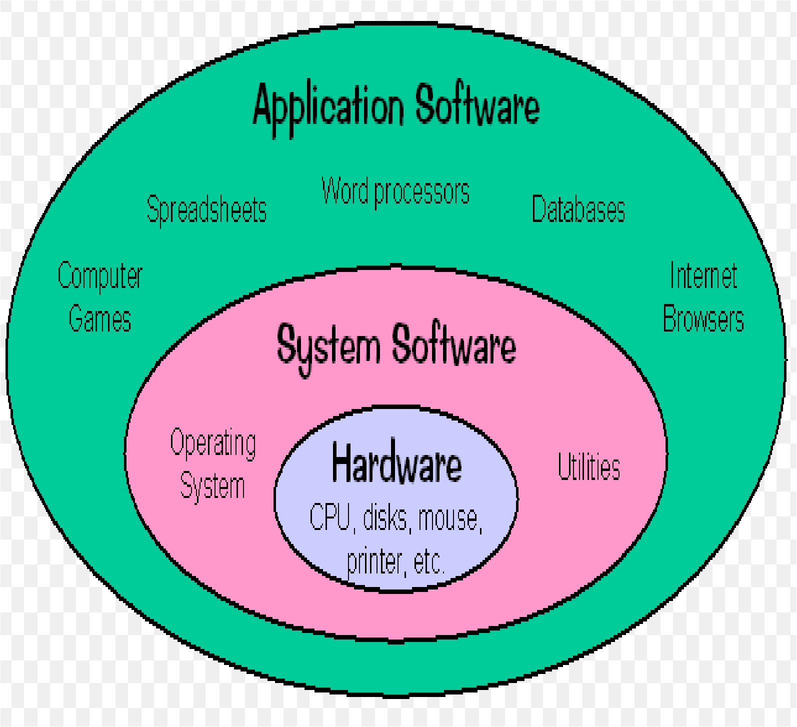
Computer Software : Types of Software, Logical systems Architecture, Acquiring Software, Software development Steps, Software Engineering.
Computer Languages : Machine language, Assembly Language, High Level Language, Some High Level Languages, Characteristics of good Programming Language.
Operating Systems : What is an Operating System, Process Management, Some Popular Operating Systems.

Computer Software

Definition : Software is a set of instructions, data or programs used to operate computers and execute specific task. Free software, open source, freeware, public domain and shareware are some of the most commonly used terms in the software industry.

The computer software is classified into two types.

1. **Systems software**
2. **Application software**



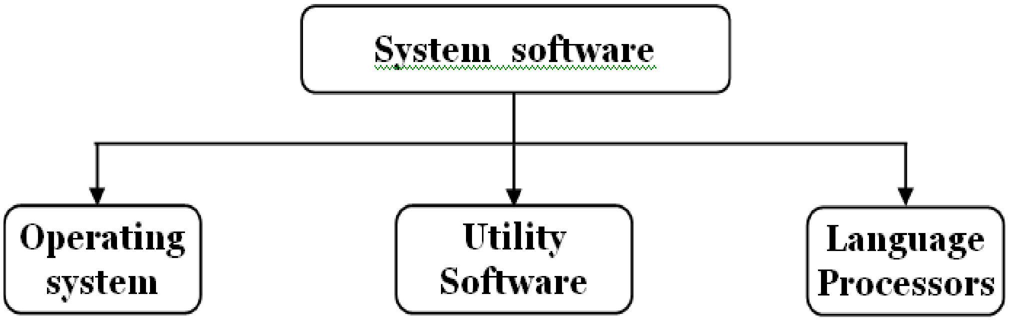
Application software :

A Specialized software dedicated to a specific purpose is called **Application software**.
✓ It is designed to perform tasks or activities for the benefit of the users.
Ex: word processor, a spreadsheet, an accounting application, a web browser, an email client, a media player, a file viewer, game or a photo editor.
✓ Application software resides above system software and includes applications such as

- Microsoft suite of products (Office, Excel, Word, PowerPoint, Outlook, etc.)
- Internet browsers like Firefox, Safari, and Chrome.
- database programs (MS Access, SQL Server etc.,)

System Software :

- ✓ System software is a collection of programs that enable the user to interact with the hardware components.
- ✓ The system software is the interface between the hardware and user applications. The operating system (OS) is the best-known example of system software. The OS manages all the other programs in a computer.



Operating system :

An operating system (OS) is system software that manages computer hardware and software resources.

- ✓ Every computer must have an operating system to run other programs and applications.
- ✓ It acts as an interface between the end user and computer hardware.
- ✓ Computer operating systems perform basic tasks, such as
  - o recognizing input from the keyboard,
  - o sending output to the display screen,
  - o keeping track of files and directories on the storage drives, and
  - o controlling peripheral devices such as printers, scanners etc.,

Utility Software :

Disk defragmenters : They are used to detect computer files whose contents are broken across several locations on the hard disk.
Disk checkers : They can be used to scan the contents of a hard disk to find files that are corrupted.
Disk space analyzers : They are used to visualizing disk space usage by obtaining the size of all folders and files in a folder or drive.

Language processors :

Compiler : A compiler is a special type of program that transforms source code written in a programming language into machine language contains two digits 1 and 0. The resultant code is known as object code, which is used to create an executable program.
Interpreter : Computer language processor that translates a program line-by-line (statement-by-statement) and executes the instructions in sequence.
Linker : A linker is a program that combines object modules to form an executable program.
Loader : A loader is a special type of program that copies programs from a storage device to the main memory where they can be executed.

Types of Applications Software

A Specialized software dedicated to a specific purpose is called **Application software**. The following are the different types of application software.

1. **Word processing software**
2. **Spreadsheet software**

3. **Presentation software**
4. **Database software**
5. **Graphics software**
6. **Education software**
7. **Multimedia software**

1. **Word processing software** :

- Word processing application software is used to format and manipulate the text.
- It is mainly used to create documents, memos, letters, faxes, advertisements, letterheads etc.,
- It allows you change font color, effect and style and
- features like thesaurus, tables, headers and footers and mail merging.

Ex : **Microsoft Word, Google Docs**

2. **Spreadsheet software** :

Spreadsheet application software is used to perform calculations. The data is stored in a table format as rows and columns. The intersection of tow and column is called cell. It allows the users to provide formulas and functions to perform calculations.

Ex : **Microsoft Excel, Google Sheets**

3. **Presentation Software** :

Presentation is a type of application software that enables you to represent your thoughts and ideas with ease and clarity by using visual information. It displays the information in the form of slides.

Ex: **Microsoft Powerpoint, Google Slides**

4. **Database Software** :

Database application software is used to create and manage a database. Also known as DBMS (Data Base Management System). You can fetch the data from the database, modify and store back in the database.

Ex : **Oracle, MySQL, Microsoft SQL Server, PostgreSQL, MongoDB and IBM DB2** etc.,

5. **Graphics Software** :

Graphics application software allows you to edit or make changes in visual data or images. It is a picture editor software.

Ex : **Adobe Photoshop, PaintShot** etc.,

6. **Education software** :

This application software is also termed as academic software. It is specifically designed to facilitate learning of a particular subject. Various kinds of tutorial software are included in this category.

Ex : Moodle, MindPlay etc.,

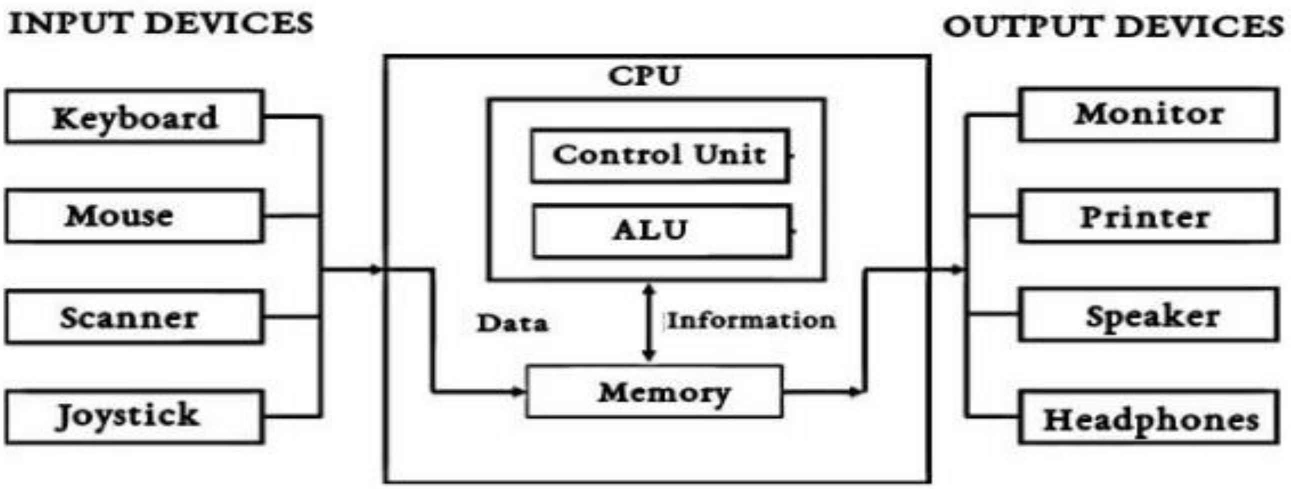
7. **Multimedia software** :

It allows easy creation of audio, video or pictorial files. They come with a wide variety of tools to facilitate the same.

Logical Systems Architecture

Modern-day computers follow the Von Neumann architecture, which was first published by John von Neumann in 1945.

Diagrammatic Representation of Logical Systems Architecture :



1. **Input Unit**
2. **Central Processing Unit**
    a. **Control Unit**
    b. **Arithmetic and Logic Unit**
    c. **Memory Unit**
3. **Output Unit**

1. **Input Unit** : They are the input devices that can be used to insert data and program into the main memory of the computer.
    - **Example** : Keyboard, Mouse, Trackball, Joystick, Scanners, Microphone, Webcam, Barcode readers, Touch Screen, Optical Mark Reader (OMR), Magnetic Ink Character Reader (MICR) etc.,
    - **Primary Input Device** : Keyboard

2. **Central Processing Unit** : It is the main processor that controls all data processing activities of a computer. It is known as the “Brain of the computer”. It stores data, intermediate results and instructions (programs). The CPU consists of three units.

    - **Control Unit** : It controls the flow of data and instructions in the processor and coordinates the activities of different units of the computer within the processor. It tells the computer’s memory, Arithmetic and Logic unit, I/O devices how to respond to program’s instructions.

    - **Arithmetic and Logic Unit** : It is the main building block of Central Processing Unit (CPU).
        - ✓ ALU performs arithmetic operations (addition, subtraction, multiplication and division)
        - ✓ Relational operations (such as less than, greater than, equal to , not equal to etc.,)
        - ✓ Logical operations (such as OR, AND, NOT, EX-OR etc.,)

        ALU is the place where the program instructions are actually executed.

        Data and instructions stored in the primary memory are transferred to ALU as and when required.

    - **Memory Unit** : It stores and supplies all the needed data, instructions, intermediate results and programs on a priority basis for the proper functioning of a computer.

3. **Output Unit** : These devices are used to output the final result processed by the computer to the user. The main function is to accept the binary data from the computer and converts them into human readable form and sends the results to the user.
   - **Example** : Monitor, Printer, Speaker, Plotter, Head phones, Projector etc.,
   - **Primary output device** : Monitor

Acquiring Software

Different kinds of software are made available for use to users in different ways.

The user may have to purchase the software, can download for free from the Internet, or can get it bundled along with the hardware.

1. **Open-Source Software**
2. **Demo software**
3. **Retail software**
4. **OEM Software**
5. **Shareware Software**
6. **Freeware Software**
7. **Public domain software**
8. **Cloud software**

1. **Open-Source Software** : It is software whose **source code** is available under a certain licence. The copy right holder grants users the rights to study, change and distribute the software.
   - ✓ You can download open source software on to your computer.
   - ✓ It can be **customized** and **altered** within the specified guidelines laid down by the creator.
   - ✓ It is the shared intellectual property of all developers and users.
   - ✓ It has restrictions on their use and modification, redistribution limitations, and copyrights.

   For example : Linux OS, Mozilla Firefox , Apache Server, MySQL and Ingres, Open Office etc.,

2. **Demo Software** : It is designed to demonstrate what a purchased version of the software is capable of doing and provides a restricted set of features. To use the software, the user must buy a fully- functional version.

3. **Retail Software** : It is off-the-shelf software sold in retail stores. It comes with printed manuals and installation instructions.
   For example : Microsoft Windows operating system.

4. **OEM Software** : OEM stands for **<u>O</u>riginal <u>E</u>quipment <u>M</u>anufacturer** software. It refers to software which is sold, and bundled with hardware. Microsoft sells its operating system as OEM software to hardware dealers. OEM software is sold at reduced price, without the manuals, packaging and installation instructions.
   For example : Dell computers are sold with the “Windows 7” OS pre-loaded on them.

5. **Shareware Software** :
   It is a program that the user is allowed to try for free, for a specified period of time, as defined in the license. It is downloadable from the Internet. When the trial period ends, the software must be purchased or uninstalled.

6. **Freeware Software**: A software that is distributed at no monetary cost to the end user is called freeware software. It is software that is free for personal use. It is downloadable from the Internet.
   - ✓ It was introduced by Richard Stallman in 1985.
   - ✓ Free Software Foundation (FSF), a non-profit organization that supports the development of free software.
   - ✓ Source code is not available to the public.
   - ✓ The copyright holder grant the user freedom to study how the software works, and modify it according to user needs.
   - ✓ The freedom to freely re-distribute the software at no cost.
   - ✓ The commercial use of this software may require a paid license.
   - ✓ The author of the freeware software is the owner of the software, though others may use it for free.
   - ✓ The users abide by the license terms, where the user cannot make changes to it, or sell it to someone else.

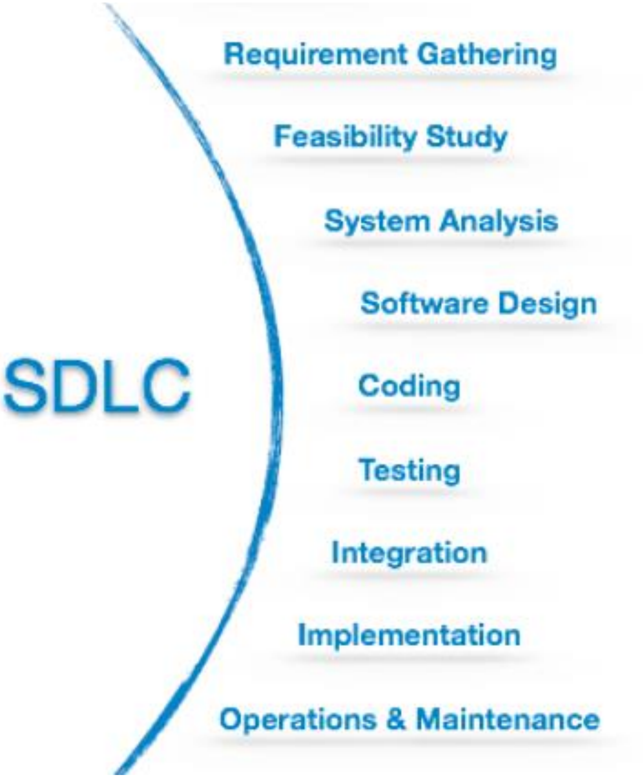
     Example : Skype, Adobe Acrobat Reader etc.,

7. **Public Domain Software** : It is free software. Unlike freeware, public domain software does not have a copyright owner or license restrictions. The source code is publicly available for anyone to use. Public domain software can be modified by the user.

8. **Cloud software** : Nowadays with the advent of Cloud computing, many application software are also available on the cloud for use through the Internet.
   Example : Google Docs, Google Sheets, Google Slides etc.,

Software Development Process (or)
Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. SDLC is the acronym of Software Development Life Cycle. It is also called as Software Development Process. It defines a methodology for improving the quality of software and the overall development process.



SDLC provides a series of steps to be followed to design and develop a software product efficiently.

The **SDLC framework** includes the following **steps** :

1. **Requirements gathering** : The requirements are segregated into user requirements, system requirements and functional requirements.
   The requirements are collected in the following ways.
   - ✓ Studying the existing system and software,
   - ✓ Conducting interviews with users and developers,
   - ✓ referring to the database or
   - ✓ collecting answers from the questionnaires.

2. **Feasibility Study** : After requirement gathering, the team prepares the rough plan for the software process. The team analyzes if the project is financially, technologically feasible for the organization.

3. **System Analysis** : After analyzing the requirements of the user, a requirement statement known as **Software Requirement Specification (SRS)** is developed. After analysis planning for the project begins. It includes developing plans that describe the activities to be performed during the project, such as software configuration management plans, project and scheduling and quality assurance plans. The resources required during the project are also determined.

4. **Software Design** : Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a **DDS - Design Document Specification**. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

5. **Coding** : The coding phase can be defined as a process of translating the software requirements into a programming language using the available tools. Writing efficient software code requires a thorough knowledge of programming language and its tools. It is important to choose the programming language according to the user’s requirements. The program code will be efficient if it makes optimal use of resources and contains minimum errors.

6. **Testing** : Software testing is performed to ensure that the software is free from errors. Software testing is performed to ensure that the software produces the correct outputs. Efficient testing improves the quality of the software. The outputs produced should be according to the user requirements, rules and guide lines specified in the SRS. In this stage, defects are reported, tracked, fixed and retested until the product reaches the quality standards defined in the SRS.

7. **Integration** : Software may need to be integrated with the libraries, databases and other programs. This stage of SDLC is involved in the integration of software with outer world entities.

8. **Implementation** : This means installing the software on user machines. Software is tested for portability, adaptability and integration related issues are solved during implementation.

9. **Deployment and Maintenance** : In this stage the users are trained, or aided with the documentation on how to operate the software and how to keep the software operational. Software maintenance starts after the software product is delivered to the user. The objective of software maintenance is to make the software operational as per user requirements. Software maintenance focuses on fixing errors, recovering from failures such as hardware and software failures.

Software Engineering

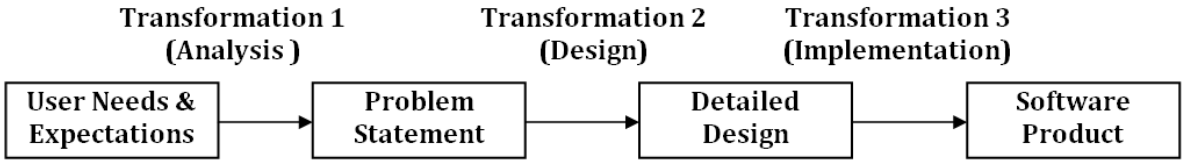
Definition :

- According to **IEEE**, software Engineering is defined as "The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software".
- Software engineering can be defined as a systematic approach to develop software within specified time and budget.
- Software Engineering is an engineering branch associated with the development of **software product** using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.

Software Product : A program is an executable code, which performs a particular task. Software is considered to be collection of executable programming code, associated libraries and documentations. Software designed for a specific requirement is called **Software Product**.

Ex : Telecommunication, Military, Medical sciences, Online shopping, Online Banking, etc.,

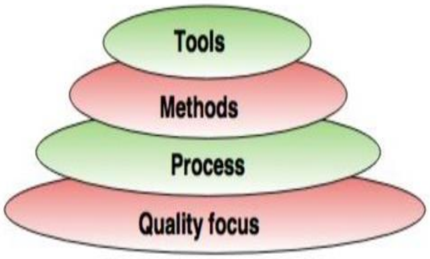
The software is the key element in all **computer based systems** and **products**. The **main objective** of software engineering is to give a framework for building software product with best quality.



Software Development Process Transformations

Software Engineering - Layered Technology :

Software Engineering is a fully layered technology. The main objective of software engineering layers is to help software developers obtain high-quality software. There are four types of layers in Software Engineering. To develop software, we need to go from one layer to another. All these layers are related to each other. It is the application of principles that deals with the design, development, testing, deployment and management of systems.



1. **Quality Focus** : It includes the characteristics of good quality software such as
   - Maintainability of the software.
   - Providing integrity
   - Checking the correctness of the functions (test input and test output) by the software.
   - Providing security so that the unauthorized user cannot access information or data.
   - Usability of the software.

2. **Process** : It is the **base layer** or **foundation layer** for the software engineering. It covers all activities, actions and tasks required to be carried out for software development. It defines a framework that includes different activities and tasks.

3. **Methods** : Methods encompass a broad array of tasks that include communication, requirements analysis, design modelling, program construction, testing and support.

4. **Tools** : Software Engineering tools provide automated or semi-automated support for the process and methods. When tools are integrated, information created by one tool can be used by another tool, a system for the support of software development is called Computer-Aided Software Engineering. CASE tools are used by software project managers, analysts and engineers to develop software system.

i. **Documentation tools** : Documentation in a software project starts prior to the software process, goes throughout all phases of SDLC and after the completion of the project. They generate documents for technical users and end users. Technical users are mostly in-house professional of the development team who refer to system manual, reference manual, training manual, installation manuals etc.,

ii. **Project management tools** : These tools are used for project planning, cost and effort estimation, project scheduling and resource planning. Project management tools help in storing and sharing project information in real-time throughout the organization.

iii. **Process modelling tools** : They help the managers to choose a process model or modify it as per the requirement of software product.

iv. **Diagram tools :** These tools are used to represent the system structure in a graphical form. They are also used to represent system components, data and control flow among various software components.

Operating System

The set of system programs used to manage the various resources and overall functions of the computer system is called operating system. It is like backbone of a computer which manages both hardware and software. It is computer’s master control program.

1. Operating systems acts as a resource manager.
2. It acts as a communication channel between the user and the computer machine.
3. It acts as an interface between the computer and the hardware.
4. Operating system boots the computer.
5. It provides a platform for executing various application programs.
6. It secures the user programs.
7. It helps in performing input/output operations.

Functions of Operating System : Operating system performs the following functions.

1. **Resource Management**
2. **Process management**
3. **Memory management**
4. **File management**
5. **Device management**
6. **Security management**
7. **Command interpretation**
8. **Job scheduling**
9. **Data security and Integrity**
10. **Error handling**

1. **Resource Management** : Resource Management : Files, devices, peripherals, CPU, memory etc., are some of the major resources of a computer. A computer has various processes running simultaneously at any point of time. These processes require different resources at different times and for different durations.
   - To keep track of available resources i.e., to keep a note of which resource is free.
   - To allocate the resource to the process.
   - To deallocate the resource after the alloted time is over.
   - To decide which resource has to alloted for a particular process. For taking this decision, it takes help of algorithms like First Come First Serve (FCFS), Shortest Path First, Round Robin etc.,

2. **Process management** : A program in execution is called process.
   - Processor is the most important resource of a computer.
   - It needs certain resources like CPU time, memory, files and I/O device to perform a task.
   - The resources are given to the process when it is created at run time.
   - When many processes are running on a computer, the OS decides which process has to be allocated to the processor, when and for how long.
   - It allocates the CPU time to the process.
   - It deallocates the processor, after the alloted time slot.

3. **Memory management** : A program must be loaded into the main memory along with its data for execution. The functionality of OS that handles main memory is called **memory management**. Operating system allocates memory space to system programs, user programs and data. It is simple when there is a single program in memory. It is a difficult job in multi-programming system as number of programs reside in computer memory.

4. **File Management** : OS is responsible for storage of files on various storage devices and transfer of files from one storage device to another. It also keeps track of the location, access time, modified time, extension of file etc., It also keep track of files and directories saved on the computer. It performs the following tasks.
   a) Create , copy, delete, move and rename file and directory
   b) Mapping file on to secondary storage
   c) Back up files

5. **Input/output management** : It manages all the devices of the computer system such as printers, keyboards etc. It allocates and deallocates the input and output devices while one or more programs are being executed.

6. **Security management** : Computer systesm store a large amount of data, which can be confidential and very valuable to its users. It protects users data from unauthorized users and denies them access to data. It protects user data from virus programs. It also protects the system resources against unauthorized users.

7. **Job scheduling** : The data, programs and transactions that are to be processed are bundled together into jobs for execution. It determines and maintain the order in which jobs are to be executed in computer system.

8. **Command interpretation** : When the user issues a command OS translate into code that can be understood by the computer. It ensures that the results of your actions are displayed on screen, printed and so on.

9. **Data security and Integrity** : It keeps data and different programs in such a manner that they do not interfere with each other.

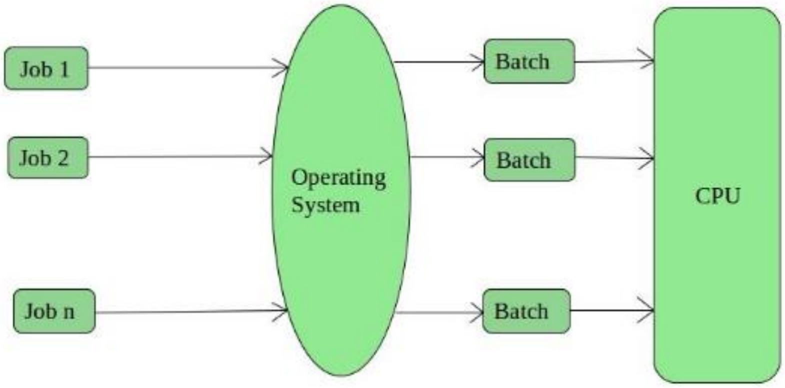
10. **Error handling** : Errors may occur either in CPU, memory or programs. It will describe those error messages and error protecting aids. It maintains the internal clock of the system

Types of Operating Systems

The set of system programs used to manage the various resources and overall functions of the computer system is called **operating system**. It acts as an interface between the hardware and the software.

Batch processing Operating system : The data and programs that are to be processed are bundled and collected as a **“batch”** and executed together. It performs very well when a large amount data has to be processed in similar nature. Batch processing is performed automatically without any user intervention.

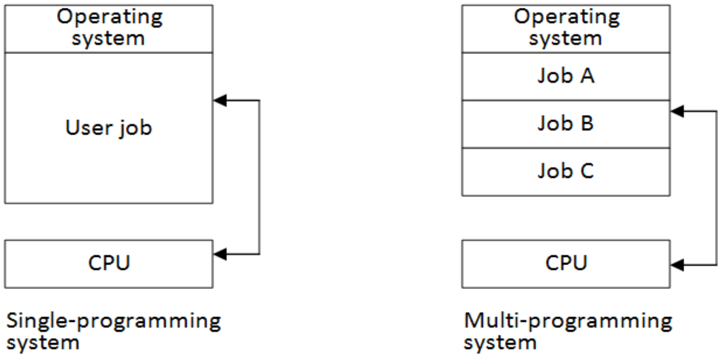
Ex: Payroll system, Bank statements etc.,



Single-user single-tasking operating system : The single-user single-tasking operating system allows only one program to be executed at a time. It was designed to manage the computer to enable a single user to do a single job effectively at any point of time.

Multiprogramming operating system : In batch processing system, only one job was stored in the memory and executed by the CPU. CPU remains idle when the job has an I/O operation. Thus, the entire main memory and CPU time were exclusively reserved for a single job.

In the multiprogramming operating system, more than one job resides in the main memory, so that when one job is busy with an I/O operation, it allocated the CPU to another job which is waiting for execution.



Single-user Multitasking operating system : It allows a single user to simultaneously perform several tasks. It is normally used in desktops and laptops. It enhances the productivity of the users.

Ex: When you are typing a document in MS Word while listening to a song and downloading a file from the internet.

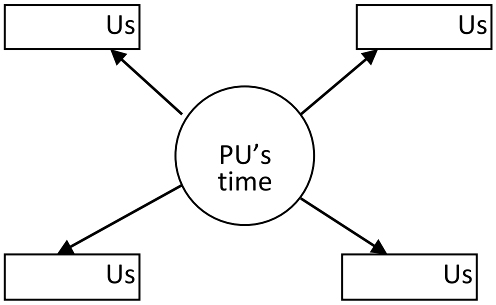
With the help of multi-tasking the user can perform several jobs at the same time.

Ex: Microsoft windows, Apple's Mac OS are single-user multi-tasking operating systems.

Multi-user Multi-tasking operating system : In this OS, each user accesses the same operating system at different machines. It enables multiple users to connect to the main computer (which has only one CPU and one OS) to perform more than one job at a time. It is quite complicated, as it must efficiently manage the jobs and resources required by different users connected to it.

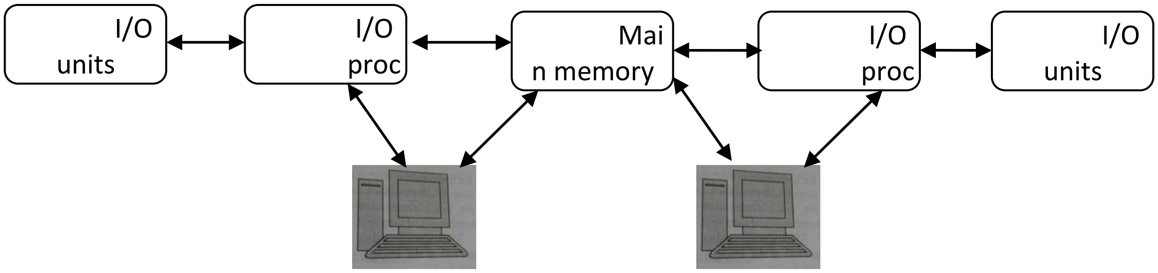
They are mainly used in mainframes.

Time sharing Operating system : An OS in which two or more users can share the processing power of the central computer through individual terminals is called **Time sharing Operating System**. The time given for running the program is called **time-slice** or **timeslot** (usually between **10 – 100 ms**). It allots timeslot for each job dynamically in round-robin fashion for program execution. When the timeslot allotted to a user's job expires, the operating system allocates CPU time to the next user on the system.



Multiprocessing operating system : Multiprocessing means using two or more processors (CPUs) within a single computer system. A complex program can be divided into smaller parts and then be executed concurrently by multiple processors.

Multiprocessing system has two processors for carrying out arithmetic and logical operations, two I/O processors for carrying out I/O operations and a shared main memory.



Users can share resources. If one processor fails, the job can be performed by other processors. There is reduction of load on one computer. There is improved response time.

Real time OS : An OS that supports a real time application is called a real time operating system. RTOS is used in embedded systems like cell phones, air conditioners, cars and so on.

Ex: RTOS can be used in car air bags. In case of severe accident, the air bag opens at right time and the person can be out of danger.

RTOS are widely used to control a device in a dedicated application

- in industry to control machinery,
- in scientific experiments
- weapon systems
- robots
- air traffic control systems.

Machine Language

Program : Program is a set of instructions that perform a particular task. Computers work according to the given instructions in the program.

In machine language, all instructions, memory locations, numbers and characters are represented in strings of 0s and 1s.

Machine language programs are typically displayed with the binary numbers represented in octal (base 8) or hexadecimal (base 16) number systems. These programs are not easy for humans to read, write or debug.

Machine language is the lowest level of programming language and is the only language that a computer understands. All the commands and data values are expressed using 0s and 1s, corresponding to the **off** and **on** electrical states in a computer. Machine language programs can be displayed with the binary numbers represented in octal or hexadecimal system.

Advantages :

✓ Programs written in machine language are **quite efficient** because it takes **shorter time for execution**.

✓ Machine language is **highly suitable** for **small computers** which have the **limited memory**.

- Execution is fast and efficient.
- Machine language code can be directly executed by the CPU.
- No need of translator (compiler or interpreter) as machine language programs are directly understood by the computer.

Disadvantages :

- Computer understands only one language called **Machine language** or **binary language**.
- Machine language consists of zeros and ones, therefore all instructions are written in the binary form.
- It is very **difficult to learn** and **difficult to remember** the address of data and instructions.
- Machine programs are **very long**.
- Debugging refers to the process of **finding** and **correcting errors**. It is very difficult to debug the machine language program.
- The programmer must know the **internal organization** of the computer such as
    - **processor**, **address of memory locations**, **address of instructions**,
    - **general purpose registers**, **special purpose registers**, **instruction set of CPU** etc.,
- Machine language is **hardware dependent** and **Machine dependent**.
- The modification in machine language program is a difficult task.
- If a program has to be modified by adding or deleting some instructions then addresses of all the subsequent instruction are to be changed.

The **Machine language program** to add two numbers.

Address	**Instruction code Binary**	**Instruction Code Hex**	**Comments**
0000	0100 1000 0000 0000 1111 1110	4900FE	Get first input from the user and store it @00FE
0003	0100 1000 0000 0000 1111 1111	4900FF	Get second input from the user and store it @00FF
0006	1001 0001 0000 0000 0010 0010	910022	Convert first input to decimal
0009	1111 0001 0000 0000 1111 1010	F100FA	Store first decimal
000C	1101 0001 0000 0000 1111 1110	D100FE	Convert second input to decimal
000F	1001 0001 0000 0000 0010 0010	910022	Store second decimal
0012	0111 0001 0000 0000 1111 1010	7100FA	Add two numbers
0015	1010 0001 0000 0000 0010 0000	A10020	Output the sum

Assembly Language

Definition : A low level language is designed for a specific family of processors that represents various instructions in symbolic code (mnemonics) and a more understandable form, called **Assembly language**.

Each personal computer has a microprocessor. Processor understands only machine language instructions which are strings of 1s and 0s.

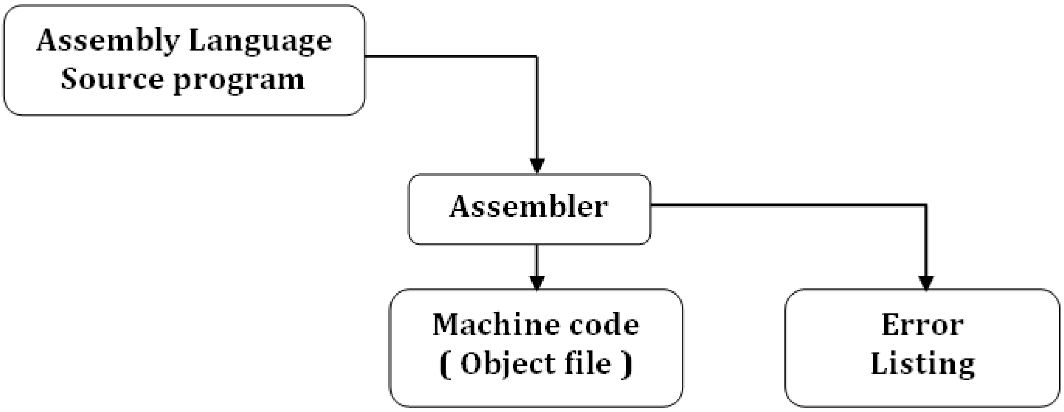
Assembly languages are symbolic programming languages that use symbolic notations to represent machine language instructions. These languages are closely connected to machine language and the internal architecture of the computer system.

Each processor has its own set of instructions for handling various operations like

- Getting input from keyboard,
- Displaying information on screen and
- Loading data and instructions from memory to registers,
- Performing arithmetic and logical operations etc.,

These set of instructions are called ‘machine language instructions’. They are complex and difficult to use.

Assembler : A special program (or language translator) required to convert the code written in assembly language into an equivalent code in machine language is called **assembler**.


Assembly Language Source program
Assembler
Machine code (Object file)
Error Listing

Syntax of Assembly language statement :

[label] mnemonic [operands] [; comments]

A basic instruction has two parts,

- ✓ the first one is the name of the instruction (or the mnemonic) which is to be executed, and
- ✓ the second are the operands or the parameters of the command.

The fields in the square brackets are optional.

Example :

```
INC  COUNT                ;  Increment the memory variable COUNT
MOV  TOTAL, 48            ;  Transfer the value 48 into the variable TOTAL
ADD  AH, BH               ;  Add the content of BH register with AH register
MUL  AL, BL               ;  Multiply the content of BL register with AL register
```

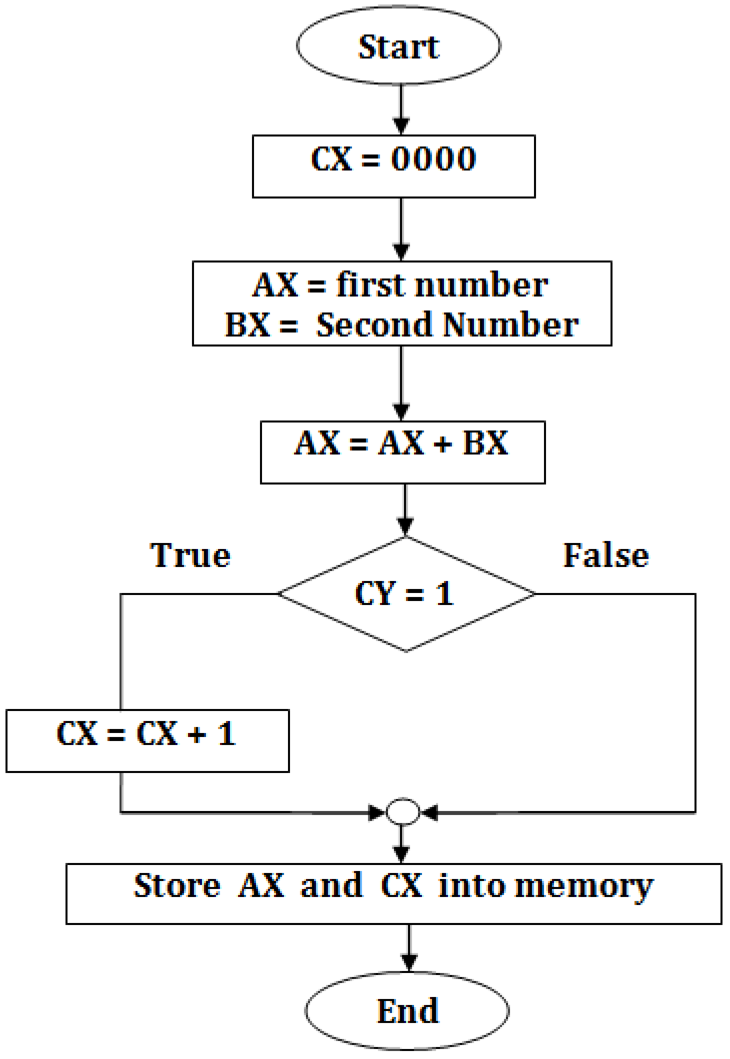
Assembly language program to add two numbers :

Description : We take the numbers from memory to AX and BX register. Then add them using ADD instruction. When the Carry is present store carry into memory, otherwise store the result AX into memory.

```
        MOV  CX, 0000          ; Initialize Count register with 0000H
        MOV AX, [3000]         ;  Load the first number into AX
        MOV BX, [3002]         ;  Load the second number into BX
        ADD  AX, BX            ;  Add AX and BX, and store to AX
        JNC  STORE             ;  If CY=0, jump to STORE
        INC   CX               ;  Increase the Count register by 1

STORE :     MOV [3004], AX            ; Store the AX register into memory
            MOV [3006], CX            ;  Store CX value into memory location
            HLT                       ; Terminate the program
```

Flow Diagram :



Advantages :

- ✓ It is easy to understand.
- ✓ It is easier to write programs in assembly langue than in machine language.
- ✓ It is easy to detect and correct errors.
- ✓ It is easy to modify.
- ✓ It is less prone to errors.

Disadvantages :

- ✓ Programmers must have a good knowledge of the hardware and internal architecture of the CPU.
- ✓ The code cannot be directly executed by the computer.
- ✓ Assembler, a translator is needed to convert into machine instructions.
- ✓ Code is machine dependent.
- ✓ Not portable.

High Level Languages

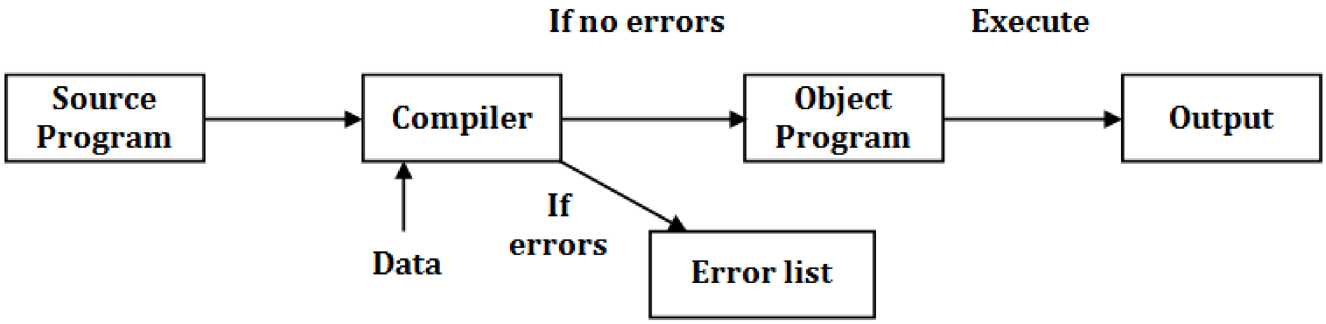
Programming languages are the primary tools for creating software. There are hundreds of programming languages in the market. High level languages are Third generation programming languages.
Examples : Pascal, FORTRAN, COBOL, C, C++, Java, Perl, Ruby, Python, VBScript, JavaScript, pHp etc.,

- ➢ Programmer friendly. Machine Independent.
- ➢ Easy to learn and understand. They are easy to use.
- ➢ They increase the readability of the program as it uses English like words.
- ➢ Debugging is easy.
- ➢ Each HLL contains a set of keywords.
- ➢ They deal with the variables, constants,
  - ✓ complex arithmetic or Boolean expression,
  - ✓ rich set of operators, arrays, objects,
  - ✓ functions, branching and looping, threads, locks, etc.,
- ➢ It is similar to human language, and the machine is not able to understand this language.

Language Processors : Compilers, interpreters are translators that converts the programs written in high-level languages into machine code that a computer understands.

Compiler : A compiler is a special type of program that transforms the source code written in a high level programming language into machine language, which uses only two digits - 0 and 1. The resultant code in 0s and 1s is known as the object code. The object code is used to create an executable program.
Each high level language has a separate compiler.
Ex : C, C++, Java etc.,

Diagrammatic Representation of Compiler :



The work of compiler is only to translate the human-readable source code into a computer-executable machine code.

Syntax Errors : Errors that limit the compiler in understanding a program are called syntax errors.
Examples : spelling mistakes, typing mistakes, illegal characters and use of undefined variables.

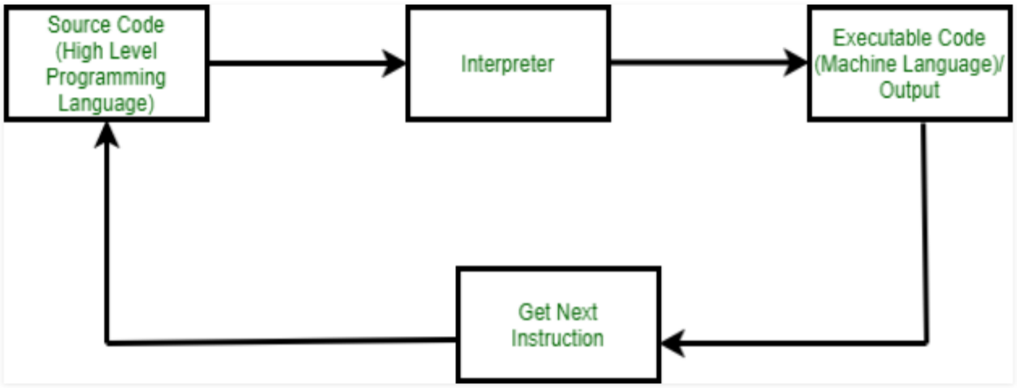
Logical Errors : Logical error occurs when the program does not function accurately.

- o Logical errors are much harder to locate and correct than syntax errors.
- o When errors are detected in the source code,
  - o the compiler generates a list of error messages indicating
    - ▪ the type of error and the line in which the error has occurred.
- o The programmer makes use of this error list to correct the source code.

Unless the **syntactical error is rectified**, the source code cannot be converted into the object code.

Interpreter : The translation of a single statement of the source program into machine code is done by a language processor and executes immediately before moving on to the next line is called an **interpreter**.

Example: Perl, Python, VBScript etc.,


Source Code (High Level Programming Language)
Interpreter
Executable Code (Machine Language)/ Output
Get Next Instruction

Compiler	Interpreter
A compiler is a program that converts the entire source code of a programming language into executable machine code for a CPU.	The language processor that translates each single statement into machine code executes immediately before moving on to the next line is called an **interpreter**.
Takes a large amount of time to analyze the entire source code	Takes less amount of time to analyze the source code
Overall execution time of the program is faster.	Overall execution time of the program is slower.
Debugging is hard, as errors are generated after scanning the entire program.	Debugging is easier.
It requires a lot of memory for generating object code.	It requires less memory for generating object code.
Examples : C, C++, Java	Examples : Python, Perl, JavaScript, Ruby

Some High Level Languages

Programming languages are the primary tools for creating software. There are hundreds of programming languages in the market. High level languages are Third generation programming languages. They are programmer friendly.

Ex : BASIC, FORTRAN, COBOL, Pascal, C, C++, Java, Perl and Ada

BASIC : BASIC (Beginner’s All-purpose Symbolic Instruction Code) is a general-purpose, high-level programming language developed by John G. Kemeny and Thomas E. Kurtz in 1964.

- BASIC was used by small business owners, professionals and a wide range of applications.
- BASIC was specifically designed to enable the students to use computers.
- BASIC was also used in teaching the introductory concepts of programming.
- Microsoft’s Visual Basic was developed by adding object-oriented programming features and a graphical user interface (GUI) to the standard BASIC.

FORTRAN :

FORTRAN stands for FORmula TRANslation. It was developed in 1957 at IBM by John Backus. Before the development of FORTRAN, programs were written either in machine language or in assembly language.

- Designed for easy translation of math formulas into code.
- First high-level programming language which is used for the scientific purpose.
- It is Machine independent.
- It is suitable for developing numeric, scientific, statistical and engineering applications.
- Used in programming
  - Video games, weather prediction, air traffic control systems, military applications,
  - Design of bridges and airplane structures,
  - Computational physics, computational chemistry and computational fluid dynamics etc.,

COBOL : COBOL stands for **CO**mmon **B**usiness **O**riented **L**anguage.

- It is developed in 1959 and used for the business and administrative purpose.
- Used by banks and major companies that depends on the accuracy of programs.

The COBOL language is divided into four parts:

- Identification Division.
- Environment Division.
- Data Division.
- Procedure Division.

Procedural / Structured programming languages :

Pascal : PASCAL was invented by Niklaus Wirth in 1970. The PASCAL is the first programming language used for the teaching tool.

- It is a procedural programming language.
- This language supports structured programming and data structure.
- Pascal allows the programmers to define complex structured data types and build dynamic and recursive data structures.

C language :

The 'C' programming language was developed by Dennis Ritchie at AT&T Bell laboratories of USA, in 1972.

- It is an outgrowth of earlier languages
  - BCPL (Basic Combined Programming language) developed by Martin Richards in 1967 and
  - B language developed by Ken Thompson in 1970.
- General purpose, structured programming language.
- Contains rich set of operators used to build complex programs.
- Best suited for writing system software as well as application software.
- Unix operating system was developed in C language.
- The compilers, libraries and interpreters of other programming languages are implemented in 'C'.

Object-oriented programming languages :

SIMULA(OOPL) :

The SIMULA was the first object-oriented programming language, developed in the 1967. The main purpose of this language is to create computer simulations. It is machine independent.

C++ : C++ is an object oriented programming language. C++ was developed by Bjourne Stroustrup at AT&T Bell Laboratories in Murray Hill, New Jersey, USA in 1980.

- C++ is an extension of C Language with major addition of **class construct** feature of **Simula67**.
- It was initially called as **"C with Classes"**.
- In 1983, it was renamed as "C++" by taking increment operator of C++.
- C++ is superset of C Language.
- Extensively used in developing system software and embedded software.

Java : It is an object-oriented programming language developed by Sun Microsystems..
It was developed by James Goosling (headed by Green Team) in early 1990s.

- It is simple , efficient , general-purpose, portable language.
- Both compiled and interpreted.
- Platform independent.
- Mainly used for developing internet-based applications.

- Java technology is used to develop the applications such as:
  - Mobile App Development, Desktop GUI Applications,
  - Web-based Applications, Gaming Applications,
  - Big Data Technologies, Distributed Applications,
  - Cloud-based Applications, IoT Applications.

LISP : LISP is an acronym for List Processing. It was developed by John McCarthy in 1959.

- General purpose and functional programming language.
- All computations are accomplished by applying functions to arguments.

For example :

A **function f** taking **three arguments** can be written as **(f arg1, arg2, arg3)**.

High Level Language	**Low Level Language**
Programmer friendly	Machine friendly
Easy to understand for programmers	Tough to understand for programmers
Simple to debug	Complex to debug comparatively
Less memory efficient	Highly memory efficient
Simple to maintain	Complex to maintain
Portable	Non-portable
Can run on any platform	Machine-dependent
Needs compiler or interpreter for translation	Needs assembler for translation

Characteristics of Good Programming Language

Programming language is a collection of instructions that the CPU (Central Processing Unit) assembles to complete a certain task in a computer. Each programming language has its own collection of keywords and syntax for constructing a set of instruction. There are thousands of computer programming languages.

Most popular programming languages :

C-language, C++, Python, Java, C#, R language, JavaScript, VBScript, etc.,

A good computer language should have following characteristics :

1. **Simple**
2. **Portability**
3. **Readability**
4. **Efficiency**
5. **Structural**
6. **Flexibility**
7. **Generality**
8. **Documentation**

1. **Simple** : A good programming language must be simple and easy to learn. It must be easy to use.

2. **Portability** : Portability refers to the ability of an application to run on different platforms (operating systems) with or without minimal changes. Due to rapid development in the hardware and software, now-a-days platform change is a common phenomenon.

3. **Readability** : If a program is developed structurally, it becomes more readable. It helps the programmers to understand the program in a better way. The program should be written in such a way that it makes other programmers to follow the logic of the program without much effort.

4. **Efficiency** : Programs written in a good programming language are translated into machine code efficiently. Every program requires certain processing time and memory to process the instructions and data. A program should utilize the least amount of memory and processing time as they the most precious resources of a computer.

5. **Structural** : A programming language must be well structured so that it is suitable for application development. To develop a program, the task must be broken down into a number of subtasks. These subtasks are developed independently and each subtask is able to perform the assigned job without the help of any other subtask. The testing and documentation process also becomes very easy.

6. **Flexibility** : Most of the programs are developed for a certain period and they require modifications from time to time. For example, in case of payroll management, as the time progresses, some employees may leave the company while some other may join the company. Hence, the payroll application must be flexible to incorporate all the changes without having to reconstruct the entire application.

7. **Generality** : The program should be general. Generality means that if a program is developed for a particular task, then it should also be used for all similar tasks of the same domain.
   For example, if a program is developed for a particular organization, then it should suit all the other similar organizations.

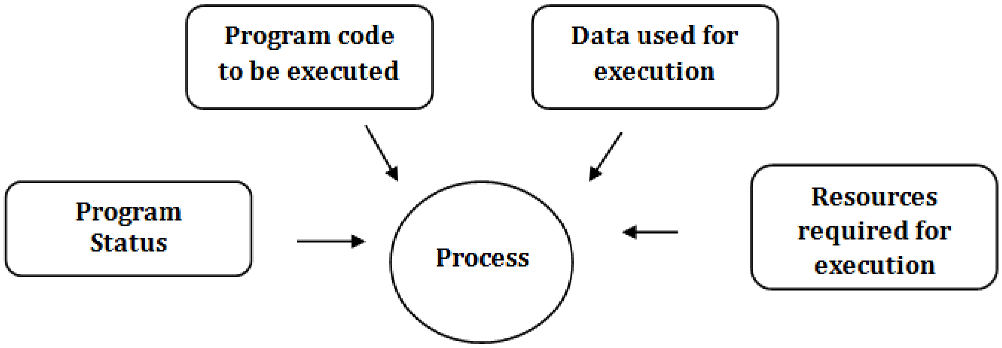
8. **Documentation** : It is one of the most important components of an application development. A well-documented application is also useful for other programmers because even in the absence of the author, they can understand it.

Process Management

Definition : A process is a program in execution. Process is an important part of modern-day operating systems. The operating system must allocate the resources to the process for its execution. It is the job of OS to manage all the running processes of the system. It handles operations by performing tasks like process scheduling and resource allocation.

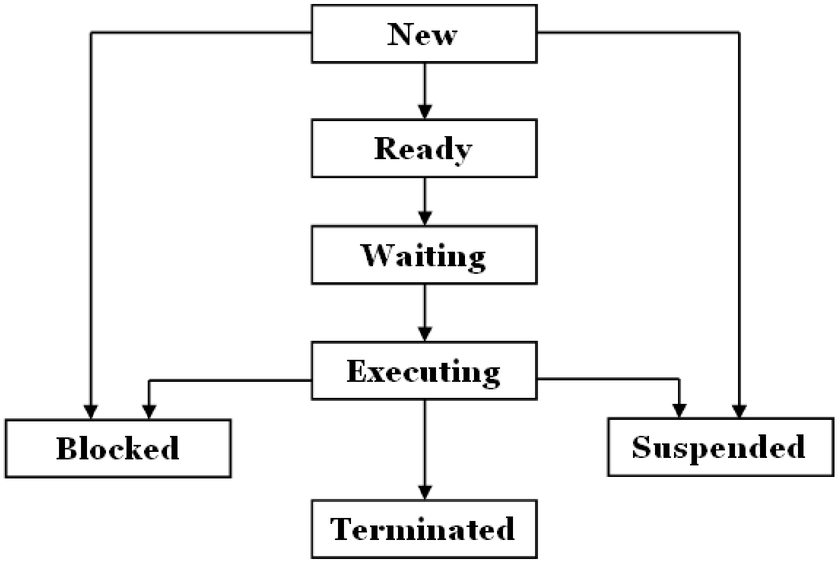
Example : When we write a program in C or C++ and compile it, the compiler creates binary code. The original code and binary code are both programs. When CPU executes the binary code instructions, it becomes a process.

Components of a process : Process components include code, data, resources and status of the process.



Process Management : Process management involves various tasks like creation, scheduling, termination of processes.

Process state : A process state is a condition of the process at a specific instant of time. A process can be new, ready, running, waiting etc.,



Process States Diagram

New : The new process is created when a specific program calls from secondary memory to primary memory.

Ready : In a ready state, the process should be loaded into the primary memory, which is ready for execution.

Waiting : The process is waiting for the allocation of CPU time and other resources for execution.

Executing : The process is an execution state.

Blocked : It is a time interval when a process is waiting for an event like I/O operations to complete.

Suspended : It defines the time when a process is ready for execution but not been placed in the ready queue by OS.

Terminated : Terminated state specifies the time when a process is terminated.

Process Control Block : PCB stands for Process Control Block. It is also known as Task Control Block. Every process is represented in the operating system by PCB. It is a data structure that is maintained by the operating system for every process. The PCB should be identified by an integer Process ID (PID). It stores all information required to keep track of all the running processes. The information is quickly updated in the PCB by the OS as soon as the process makes the state transition.

Process number which identifies the process
Pointers which point to the PCB of other processes
Current state of the process
Program counter, which stores the location of the next instruction to be executed

Contents of **CPU registers**

Upper and lower bounds of the memory required for the process

List of files opened by the process

Priority of the process

Status of all I/O devices needed by the process

Process control block

Pointer	Process state
Process number	
Program counter	
CPU registers	
Memory allocations	
Event information	
List of open files	
○ ○ ○	

Some Popular operating systems

The following are the most popular operating systems.

1. MS-DOS
2. Windows Operating System
3. Unix OS
4. Linux OS

MS-DOS

MS-DOS stands for Microsoft Disk Operating System. It is a Single-User and Single-Tasking operating system. MS-DOS is an Character User Interface. It is acts as an interface between hardware and user.

MS-DOS Commands are classified into two parts.

1. Internal Commands
2. External Commands

Internal commands :

Once the computer is booted, the **command.com** file is loaded into main memory of the computer. These are also called memory-resident commands.

Command	**Description**	**Example**
CLS	Clears the screen	C:\> CLS

DIR	Lists the files and directories	C:\>DIR /p Page wise
		C:\>DIR/w Width wise
		C:\> DIR/ad Displays directories only

VER	Displays current version	C:\> VER Windows 98 [Version 4.10.2222]

DATE	Displays and allows to change system date	C:\> DATE
		Current date is : Fri 02-15-2002
		Enter new date (mm-dd-yy):

TIME	Displays and allows to change system time	C:\> TIME Current time is 8:38:47.70a Enter new time:
COPY CON	To create a new file	C:\> COPY CON proverbs.txt A friend in need is a friend in deed. ^z or Press F6 Function key 1 file(s) copied
TYPE	To display the content of file	C:\> TYPE filename
REN	To rename a file	C:\> REN old-file-name New-file-name
DEL	To delete a file	C:\> DEL filename
MD	Create a new directory	C:\> MD directory-name
CD	Change directory	C:\> CD directory-name
RD	Remove directory	C:\> RD directory-name

Extenal commands :

These are stored in a secondary storage device. They help to perform some specific task. External commands are known as Disk-residence commands.

1. **TREE** : It displays directories and file names in each sub-directory.

```
C:\> TREE
```

2. **SORT** : This command is useful when we want to sort a file. When we run this command the result can be get to display device or file.

```
Syntax :  C:\> SORT  /R <file name>   // sorts in reverse order
          C:\> SORT Player.txt
```

3. **MOVE** : Move command is used for moving one file or multiple files from one location to another location or from one disk to another disk.

   Syntax :
   ```
   C:\> MOVE <file name> <path name>
   C:\SONGS> MOVE     *.MP3        C:\ SONGS\OLD SONGS\
   ```

4. **FIND** : The FIND command is used to search a file for a text string.

   Syntax :
   ```
   C:\> FIND "String to search" <File name>
   C:\TEST>FIND  "friend"  proverb.txt
   ---------- proverb.txt
   A friend in need is a friend in deed.
   ```

5. **DOSKEY** : Once we install doskey , our dos will memorize all commands. We can recall those commands using up or down arrow keys.

   Syntax :
   ```
   C:\> DOSKEY
   ```
   To recall commands, we can use UP, DOWN arrows.

   Esc clears current command

6. **MEM** : This command displays free and used amount of memory in the computer.

   Syntax:
   ```
   C:\> MEM
   ```

7. **DISKCOPY** : DISKCOPY copies the contents of a floppy disk to another.

   Syntax :
   ```
   C:\> DISKCOPY <Drive1> <Drive2>
   C:\> DISKCOPY   A:   B:
   ```
   This command will be copy all contents of A drive to B drive.

8. **FORMAT** : This command creates new Track & Sectors in a disk.

   Syntax:-
   ```
   C:\> FORMAT [drive name]
   C:\> FORMAT   A:
   ```
   this command will format the floppy disk.

9. **CHKDSK** : This command is used to check the status of a disk.

   Syntax :
   ```
   C:\> CHKDSK
   ```

Windows Operating System

The Windows operating system (Windows OS) refers to a family of operating systems developed by Microsoft Corporation. Windows provides a graphical user interface (GUI), virtual memory management, multitasking and support for many peripheral devices.

Version	**Year**
Windows 1.0	1985
Windows 95	1995
Windows NT (New Technology)	1996
Windows 98	1998
Windows 2000, Windows ME	2000
Windows XP	2001
Windows Vista	2006
Windows 7	2009
Windows 8	2012
Windows 10	2015
Windows 11	2020

Windows Features :

1. Desktop
2. Start menu
3. Control panel
4. Taskbar
5. Windows search box
6. Task Manager
7. Notification area
8. System Information
9. File Explorer
10. Device Manager
11. Event Viewer
12. Disk Cleanup

Desktop : The desktop is a fundamental part of Windows operating system. When you power on your computer and log in to Windows, the first screen that appears is called desktop. It is a space where you can organize applications, folders and documents which appear as icons. It is always background, behind any other applications you are running. You can access the installed programs by double clicking any application short cuts on the desktop.

Start menu : The Start menu is a list of applications and utilities installed on your computer. You can open it by clicking Start on the left side of your taskbar. You can open the Start menu by pressing the windows key in the keyboard. You can access the installed programs on your computer from the Start menu.

Control panel : The Control Panel is a collection of tools help you manage the resources on your computer. You can change settings for printers, video, audio, mouse, keyboard, date and time, user accounts, installed applications, network connections, power saving operations etc.,

Taskbar : The Windows taskbar shows programs that are opened currently.

Windows search box : The Windows search box is a convenient way to search for documents, pictures, videos, applications etc.,

Task Manager : The **Task Manager** gives you an overview of what is running on your computer. You can see how much of your system resources is used by each application (task), sorting by CPU, RAM and disk I/O usage. If a program is frozen or not responding, you can right-click it in Task Manager and end the task, forcing it to quit.

Notification area : It is also known as the System tray. It is located on the right side of the taskbar. It shows the date and time, programs running in the background. It also provides Internet connection’s status and a speaker icon for adjusting sound volume.

System Information : The **System Information** utility provides information about the computer. You can find out the details about your computer’s hardware, including processor, memory, video card and sound card etc., You can also view the environment variables, device drivers, services etc.,

File Explorer : It is also called as **Windows Explorer**. It provides you with a view of the files and folders on the computer. You can browse the contents of your hard disk and **removable disks**. You can search for files and folders, and open, rename or delete them from the File Explorer.

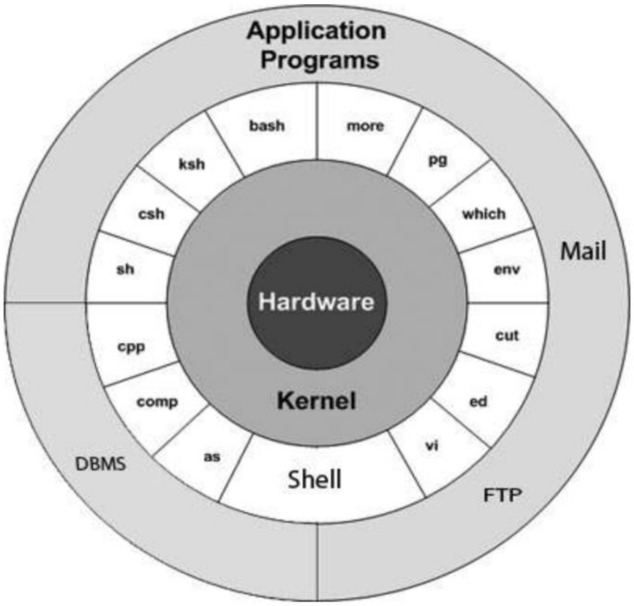
Device Manager : The Device Manager lists the hardware devices installed in a computer. It allows users to see what hardware is installed, view and update hardware drivers. You can also be used to uninstall hardware.

Event Viewer : The Event Viewer is an **administrator tool** displays errors and important events that happen on your computers. It helps troubleshoot advanced problems on your windows operating system.

Disk Cleanup : The Disk Cleanup helps increase free disk space on your computer by removing **temporary** or **unnecessary files**. It improves the **performance of the computer**.

Unix Operating System

The UNIX operating system is a multiuser, portable, time-sharing and multitasking operating system designed by AT&T employees at Bell Laboratories in 1969. The terms multiuser and multitasking mean that many different users can perform tasks simultaneously on one machine. Each user interacts with their own instance of the shell and can start one or more applications within that shell.



Kernel :
The Kernel is the heart of the UNIX OS. It is a software application that provides the interface between the hardware and the user. It handles the process, memory, file device and network management for the operating system. The kernel is responsible for ensuring the all system and user tasks are performed concurrently.

Shell :
The shell is the program that sits between the user and the kernel. It is the interpreter that translates the commands that are typed into the terminal session. Users can type commands directly into the terminal, or they can create a text file containing a series of commands that can be sent to the shell. The series of commands are called a shell script.

- ✓ There are multiple shells that are used by the UNIX OS. They include
  - The Bourne shell (sh),
  - The C shell (csh),
  - The Korn shell (ksh) and
  - The Bourne Again Shell (bash)
- ✓ Each shell has own set of shell commands.
- ✓ Operating system commands are the same across all the shells.

Utilities and Application Programs :

The final layer of the UNIX OS is the Utilities and Application layer. This layer includes the commands, word processors, graphic programs and database management programs. These programs were accessed by typing the commands to start the program on the command line.

Basic UNIX commands :

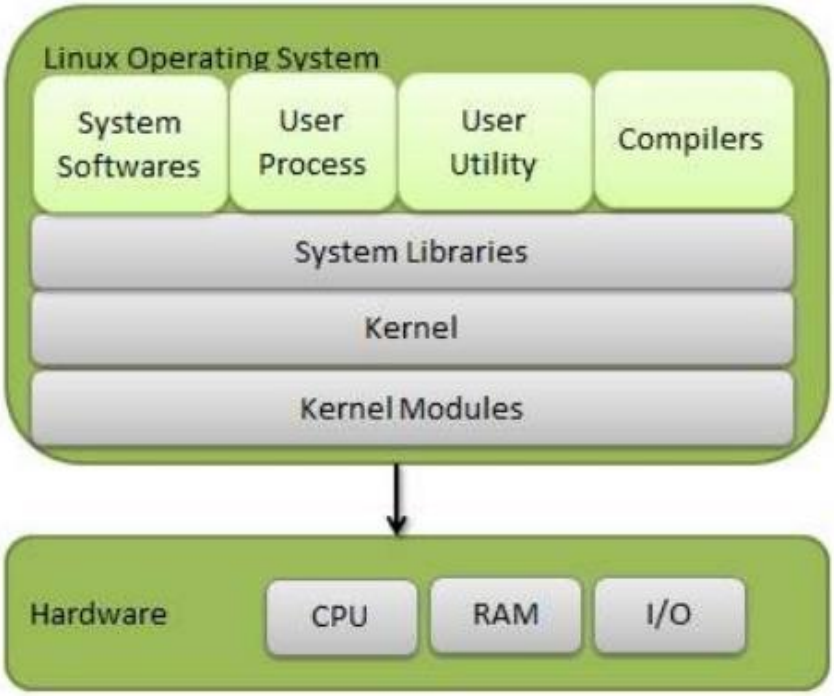
Command	**Description**
who	The '$ who' command displays all the users who have logged into the system currently. Ex : **$who**
pwd	The '$pwd' command stands 'print working directory' and it displays the directory in which we are currently. Ex : **$pwd**
ls	It simply displays the contents of a directory. Ex : **$ls folder_name**
touch	'$touch' command creates a file, you can add an extension such as .txt after file name. Ex : **$touch filename.extension**
cp	The $cp command copies the content of one file to another file. Ex : **$cp source_file destination_file**
mv	The $mv command moves the file from one directory to another directory. Ex : **$mv source_file destination_file**
mkdir	Stands for Make Directory. It creates a new directory. Ex : **$mkdir folder_name**
rmdir	Stands for remore directory. The '$rmdir' command deletes any directory we want to delete. Ex : **$rmdir**
cd	The '$cd' command stands for 'Change directory'. Ex : **$cd folder_name**

Versions : There are multiple versions of the UNIX operating system.

- ✓ Proprietary versions include
  - Sun Microsystem's Solaris, Hewlett Packard's HP UX and IBM's AIX.
- ✓ Open source versions include
  - Free BSD, NetBSD and openBSD.

Linux operating System

Linux is one of popular version of UNIX operating System. It is open source as its source code is freely available. It is free to use.



Features :

Multi-User : Linux is a multiuser system means multiple users can access system resources like memory/RAM/ application programs at same time.

Open Source : Linux source code is freely available from the internet.

Shell : Linux provides a special interpreter program which can be used to execute commands of the operating system.

Multiprogramming : Linux is a multiprogramming system means multiple applications can run at same time.

Kernel : Kernel is the core part of Linux. It is responsible for all major activities of this operating system. It consists of various modules and it interacts directly with the underlying hardware.

System Library : System libraries are special functions or programs using which application programs or system utilities accesses Kernel's features.

System Utility : System Utility programs are responsible to do specialized, individual level tasks.