ST. LAWRENCE HIGH SCHOOL

A Jesuit Christian Minority Institution

FOR GOD AND COUNTRY

STUDY MATERIAL - 6

Subject: COMPUTER SCIENCE **Class - 11** **F.M:15**

Chapter: Software and Languages **Date: 10/07/2020**

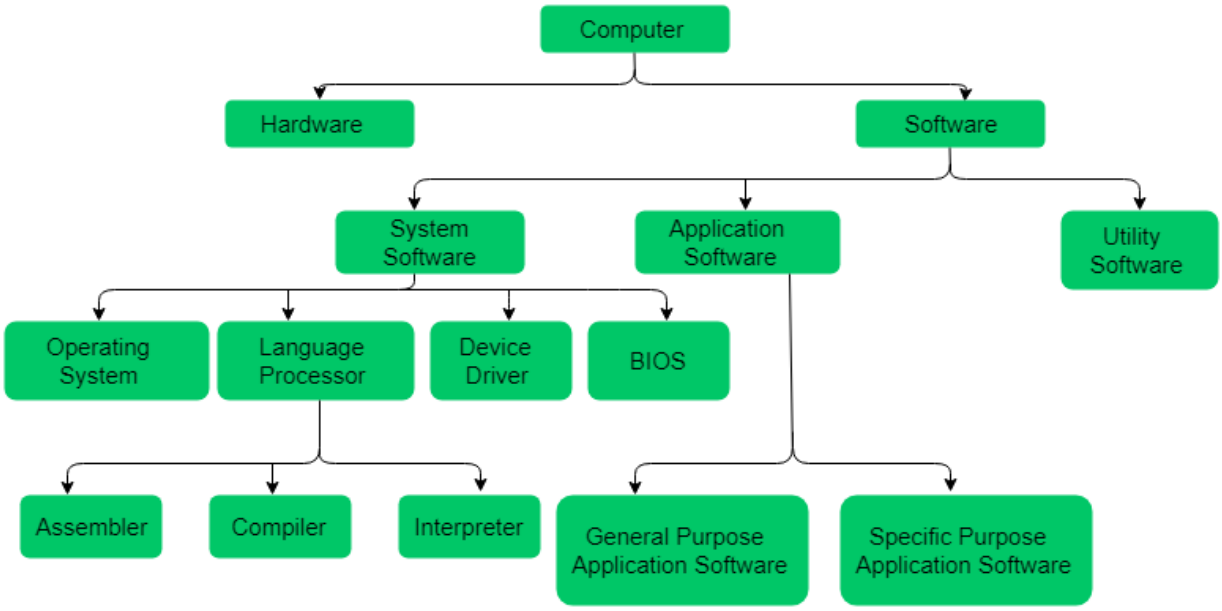
TYPES OF SOFTWARE

A Computer is an electronic device that can perform various operations of computation at a greater speed than what an ordinary machine or human mind can do. It is driven by many entities including the physical and tangible components that we can touch or feel, called the **Hardware** and programs and commands that drive the hardware, called the **Software**.
The Software refers to the set of instructions fed in form of programs to govern the computer system and process the hardware components. For example:

- The antivirus that we use to protect our computer system is a type of Software.
- The media players that we use to play multimedia files such as movies, music etc. are Software.
- The Microsoft Office we use to edit the documents is a Software.

Depending on its use and area of implementation, Softwares can be divided into 3 major types:

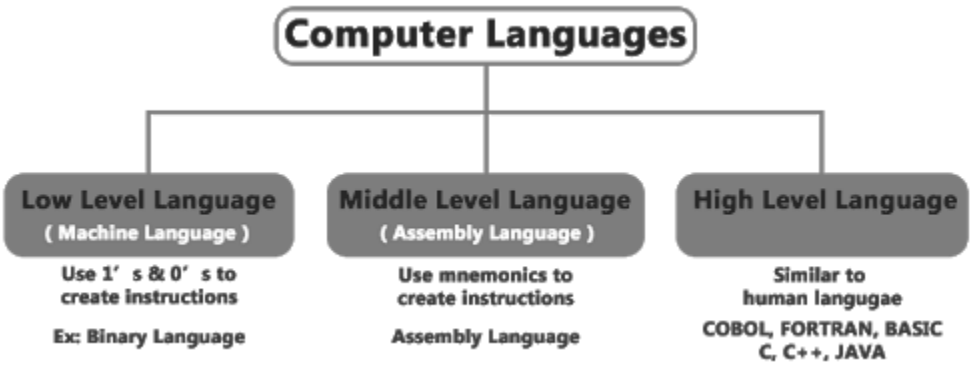
1. **System Software**
2. **Application Software**
3. **Utility Software**



System Software

These are the software that directly allows the user to interact with the hardware components of a computer system. As the humans and machines follow different languages, there has to be an interface that will allow the users to interact with the core system, this interface is provided by the software. The system software can be called the main or the alpha software of a computer system as it handles the major portion of running a hardware. This System Software can be further divided into four major types:

1. **The Operating System** – It is the main program that governs and maintains the inter-cooperation of the components of a computer system. For eg., Microsoft Windows, Linux, Mac OS etc.
2. **The Language Processor** – The hardware components present in the computer system does not understand human language. There are three types of languages involved in the world of human-machine interaction:

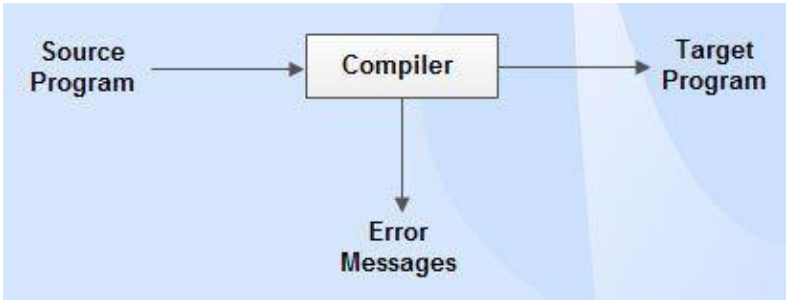


   - ***Machine-Level Language:*** The machines only understand the digital signals or the binary codes or the binary language which consist of strings of 0's and 1's. These are totally machine dependent language.
   - ***Assembly-Level Language:*** These are the Low-Level Language(LLL), that forms a correspondence between machine level instruction and general assembly level statements. Assembly language uses a mnemonics to represent each low-level machine instruction or operation-code also called the op-codes. For eg., ADD instruction is used to add two entities, the HALT instruction is used to stop a process etc. It is a machine dependent language and varies from processor to processor.
   - ***High-Level Language:*** These are the simple English statements, that humans use to program and code as it is easy to read and understand to the human world. For eg., Java, C, C++, Python etc.

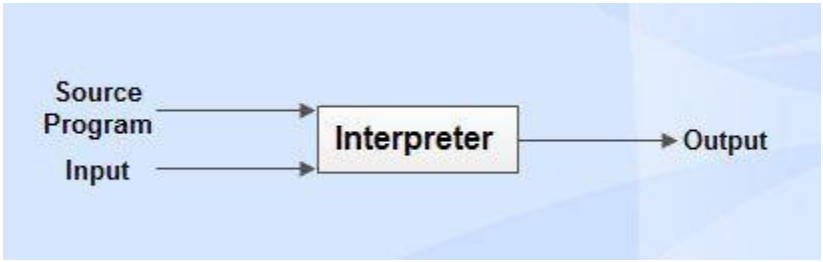
   The machine level language is very complex to understand and code, therefore the users prefer the *High-Level Language* or the *HLL* for coding. These codes need to be converted into the machine language so that the computer can easily understand and work accordingly. This operation is performed by the Language Processor which is made up of further three components:

- ***Assembler:*** This language processor is used to convert the assembly language into machine level language.

- ***Compiler:*** This language processor is used to convert High-Level Language into machine level language in one go, thus execution time is fast. The error detection is difficult in a compiler. Programming Languages like C, C++ and Scala use compiler.



- ***Interpreter:*** This language processor is also used to convert High-Level Language into machine level language line-by-line, thus execution time is slow. Error-detection is easier in an interpreter as it reports as soon as a bug is caught and restarts the process. This consumes unnecessary memory. Programming Languages like Python, Ruby and Java uses an interpreter.



3. **The Device Drivers** – The device drivers and the device programs or the system software that acts as an interface between the various Input-Output devices and the users or the operating system. For eg., the Printers, Web cameras come with a driver disk that is needed to be installed into the system to make the device run in the system.
4. **The BIOS** – *It stands for Basic Input Output System* and is a small firmware that controls the peripheral or the input-output devices attached to the system. This software is also responsible for starting the OS or initiating the booting process.

Application Software

These are the basic software used to run to accomplish a particular action and task. These are the dedicated software, dedicated to performing simple and single tasks. For e.g. single software cannot serve to both the reservation system and banking system. These are divided into two types:

1. **The General Purpose Application Software:** These are the types of application software that comes in-built and ready to use, manufactured by some company or someone. For e.g.
    - *Microsoft Excel* – Used to prepare excel sheets.
    - *VLC Media Player* – Used to play audio/video files.
    - *Adobe Photoshop* – Used for designing and animation and many more.
2. **The Specific Purpose Application Software:** These are the type of software that is customizable and mostly used in real-time or business environment. For e.g.
    - *Ticket Reservation System*
    - *Healthcare Management System*
    - *Hotel Management System*
    - *Payroll Management System*

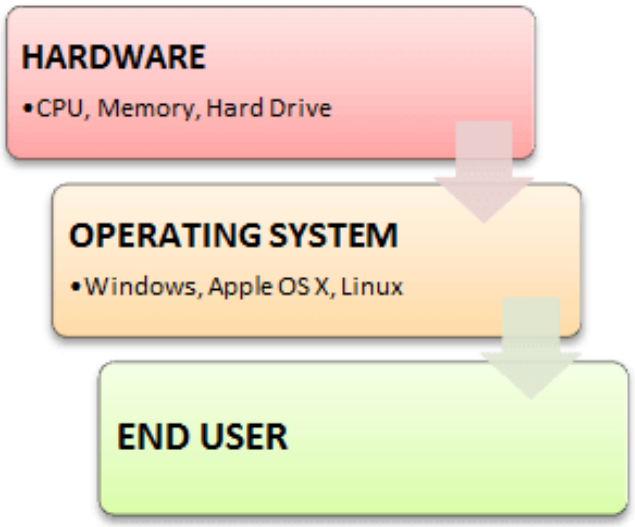
Utility Software

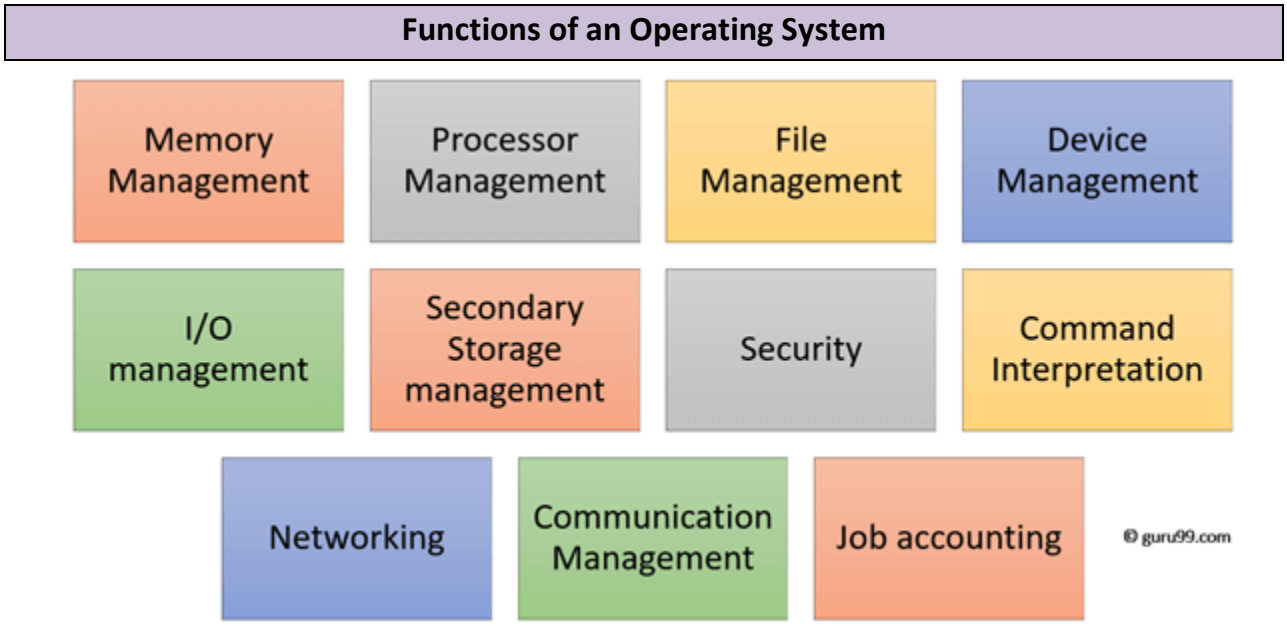
These are the most basic type of software which provides high utility to the user and the system. These perform the basic but daily need tasks. For example:

- *Antivirus Softwares:* These provide protection to the computer system from unwanted malware and viruses. For e.g. QuickHeal, McAfee etc.
- *Disk Defragmenter Tools:* These help the users to analyse the bad sectors of the disk and rearrange the files in a proper order.
- *Text-editors:* These help the users to take regular notes and create basic text files. For e.g. Notepad, Gedit etc.

Operating System

An **Operating system (OS)** is a software which acts as an interface between the end user and computer hardware. Every computer must have at least one OS to run other programs. Applications like Chrome, MS Word, Games, etc needs some environment in which it will run and perform its task. The OS helps you to communicate with the computer without knowing how to speak the computer's language. It is **not** possible for the user to use any computer or mobile device without having an operating system.





Function of an Operating System

In an operating system software performs each of the function:

1. **Process management**:- Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.

2. **Memory management:-** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.

3. **File management**:- It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.

4. **Device Management**: Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.

5. **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of those hardware devices from the user.

6. **Secondary-Storage Management**: Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.

7. **Security**:- Security module protects the data and information of a computer system against malware threat and authorized access.

8. **Command interpretation**: This module is interpreting commands given by the and acting system resources to process that commands.

9. **Networking:** A distributed system is a group of processors which do not share memory, hardware devices, or a clock. The processors communicate with one another through the network.

10. **Job accounting**: Keeping track of time & resource used by various job and users.

11. **Communication management**: Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

Types of Operating system:

Some of the widely used operating systems are as follows:

1. Single-user: It is an operating system in which the user can manage one thing at a time effectively.
2. Multi-user: It is a computer operating system which allows multiple users to access the single system with one operating system on it. It is generally used on large mainframe computers.
3. Multi-processing: This OS supports running a program on more than one CPU.
4. Multitasking: This type allows more than one program to run simultaneously.
5. Multithreading: Such an operating system allows different parts of a single program to run parallel.
6. Real time: These operating systems are used when there are time requirements are very strict like missile systems, air traffic control systems, robots etc.

ANSWER THE FOLLOWING QUESTIONS:

1. Define system software. Provide suitable examples.

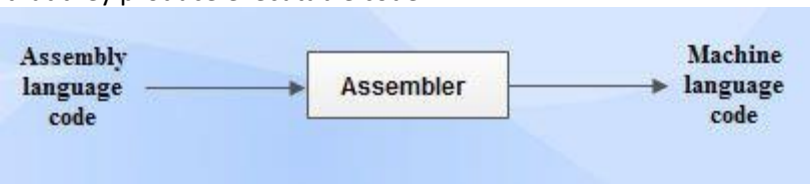
Ans: **System software** is a type of computer program that is designed to run a computer's hardware and application programs. Examples:

- Operating system
- Language processor
- Device drivers

2. Differentiate between software and hardware.
   Ans:

HARDWARE	SOFTWARE
Hardware is a physical part of computer that cause processing of data.	Software is a set of instruction that tells a computer exactly what to do.
It is manufactured.	It is developed and engineered.
Hardware cannot perform any task without software.	software cannot be executed without hardware.
As Hardware are physical electronic devices, we can see and touch hardware.	We can see and also use the software but can't actually touch them.
It has four main categories: input device, output devices, storage, and internal components.	It is mainly divided into System software, Programming software and Application software.
Hardware is not affected by computer viruses.	Software is affected by computer viruses.

3. Write a short note on assembler.
   Ans: An *assembler* is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor. *Assemblers* are similar to compilers in that they produce executable code.



4. Differentiate between an interpreter and compiler.
   Ans:

S.No.	Compiler	Interpreter
1.	Compiler scans the whole program in one go.	Translates program one statement at a time.
2.	As it scans the code in one go, the errors (if any) are shown at the end together.	Considering it scans code one line at a time, errors are shown line by line.
3.	Main advantage of compilers is it's execution time.	Due to interpreters being slow in executing the object code, it is preferred less.
4.	It converts the the instructions into systematic code.	It doesn't convert the instructions instead it directly works on source language.
Eg.	C, C++, C# etc.	Python, Ruby, Perl, SNOBOL, MATLAB etc.

5. What is an operating system? Write any five functions of OS.
   Ans: An **Operating system (OS)** is a software which acts as an interface between the end user and computer hardware. Every computer must have at least one OS to run other programs. Applications like Chrome, MS Word, Games, etc needs some environment in which it will run and perform its task. The OS helps you to communicate with the computer without knowing how to speak the computer's language.

   a. **Process management**:- Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.
   b. **Memory management:-** Memory management module performs the task of allocation and de-allocation of memory space to programs in need of this resources.
   c. **File management**:- It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.
   d. **Device Management**: Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and de-allocation of the devices.
   e. **I/O System Management:** One of the main objects of any OS is to hide the peculiarities of those hardware devices from the user.

6. Explain briefly multi-tasking OS.
   Ans: **Multitasking operating system** allows a user to perform more than one computer task (such as the **operation** of an application program) at a time. The **operating system** is able to keep track of where you are in these tasks and go from one to the other without losing information.

7. Differentiate between high-level language and low-level language.
Ans:

S.NO	High Level Language	Low Level Language
1.	It is programmer friendly language.	It is a machine friendly language.
2.	High level language is less memory efficient.	Low level language is high memory efficient.
3.	It is easy to understand.	It is tough to understand.
4.	It is simple to debug.	It is complex to debug comparatively.
5.	It is simple to maintain.	It is complex to maintain comparatively.
6.	It is portable.	It is non-portable.
7.	It can run on any platform.	It is machine-dependent.
8.	It needs compiler or interpreter for translation.	It needs assembler for translation.

8. Differentiate between application software and system software.
Ans:

S.NO	System Software	Application Software
1.	System Software maintains the system resources and gives the path for application software to run.	Application software is built for specific tasks.
2.	Low level languages are used to write the system software.	While high level languages are used to write the application software.
3.	It's a general purpose software.	While it's a specific purpose software.
4.	Without system software, system can't run.	While without application software system always runs.
5.	System software runs when system is turned on and stop when system is turned off.	While application software runs as per the user's request.
6.	Example of system software are operating system, etc.	Example of application software are Photoshop, VLC player etc.
7.	System Software programming is complex than application software.	Application software programming is simpler as comparison to system software.

9. Differentiate between multi-programming OS and multi-tasking OS.
   Ans:

Sr.no	Multiprogramming	Multi-tasking
1.	Both of these concepts are for single CPU.	Both of these concepts are for single CPU.
2.	Concept of Context Switching is used.	Concept of Contest Switching and Time Sharing is used.
3.	In multiprogrammed system, the operating system simply switches to, and executes, another job when current job needs to wait.	The processor is typically used in time sharing mode. Switching happens when either allowed time expires or where there other reason for current process needs to wait (example process needs to do IO).
4.	Multi-programming increases CPU utilization by organising jobs .	In multi-tasking also increases CPU utilization, it also increases responsiveness.
5.	The idea is to reduce the CPU idle time for as long as possible.	The idea is to further extend the CPU Utilization concept by increasing responsiveness Time Sharing.

10. Differentiate between assembler and compiler.
    Ans:

Key	Compiler	Assembler
Operation	Compiler translates high level programming language code to machine level code.	Assembler converts the assembly level language to machine level code.
Input	Source code in high level programming language.	Assembly level code as input.
Conversion type	Compiler checks and converts the complete code at one time.	Assembler generally does not convert complete code at one time.
Output	Mnemonic version of machine code.	Binary version of machine code.
Examples	C, C++ , Java compilers.	GAS, GNU assemblers.