

## Software

Computer software, or only software, is a kind of program that enable a user to perform some specific task or used to operate a computer. It directs all the peripheral devices on the computer system - what to do and how to perform a task. PC Software plays the role of mediator between the user and computer hardware. Without software, a user can't perform any task on a digital computer.

In other words, Software is a set of programs, which is designed to perform a well-defined function. A program is a sequence of instructions written to solve a particular problem.

A computer system can be divided into three components: the hardware, the software, and the users. Software can be further divide into mainly two parts: Application software and System Software. Bare use of hardware is not easy, so to make it easy, software created.

### **Application Software**

Applications software also called end-user programs or merely an application. It resides above system software. The end-user uses applications software for a specific purpose. It programmed for simple as well as complex tasks. It either be installed or access online. It can be a single program or a group of small programs that referred to as an application suite.

Application software can be used by the user to complete specific tasks, such as creating word processors documents, spreadsheets, presentations, graphics, CAD/CAM, sending the email, etc.

**a. Word Processors** Such applications are meant for documentation. It also assists in storing as well as formatting and even printing of the documents. Key examples of such software are: MS Word Apple iWork-Pages Corel WordPerfect Google Docs

**b. Database Software** It is used to create as well as manage a database and also known as Database Management System or in short, DBMS. Such software assists in the data organization. Some of the examples of DBMS are: MS Access FileMaker dBase Clipper MySQL FoxPro

**c. Multimedia Software** This is a software which is able to play, create as well as record images, audio or even video files. These software are utilized for animation, video editing, graphics as well as image editing. Due to the high demand for such software, every software product development company has vast avenues in developing them. Some of the examples of such software are: Adobe Photoshop Picasa VLC Media Player Windows Media Player Windows Movie Maker

**d. Web Browsers** These software are utilized to browse the internet. Web browsers assist the users in locating as well as retrieving data well across the web. Some of the key examples of them are: Google Chrome Mozilla Firefox Internet Explorer Opera UC Browser Safari

Other Examples of Application software are the following –

- Payroll Software
- Student Record Software
- Inventory Management Software
- Income Tax Software
- Railways Reservation Software
- Microsoft Office Suite Software
- Microsoft Word
- Microsoft Excel
- Microsoft PowerPoint

Features of application software are as follows –

- Close to the user
- Easy to design
- More interactive
- Slow in speed
- Generally written in high-level language
- Easy to understand
- Easy to manipulate and use
- Bigger in size and requires large storage space

### **System Software**

System Software (the type of computer program) provides a platform to run a computer's hardware and computer application to utilize system resources and solve their computation problem. It is written in a low-level language, like assembly language, so it can easily interact with hardware with the primary level. It controls the working of peripheral devices. System software act as a scheduler for the execution of the processes and arrange the sequence according to their priority and I/O devices requirement and creation of the process. The best-known example of system software is the operating system (OS). It responsible for manages all the other programs on a computer.

The operating system creates an interface between user and hardware and also in application software and hardware. Examples of software are Windows 7, Windows XP, and Windows 10.

Generally, the user does not interact with the System Software directly. The user interacts with the GUI created by System Software. Through this, GUI user interacts with applications installed in the system.

## Types of System Software

**a. Operating System** Being a prominent example for system software, it is essentially a collection of software which handles resources as well as offers general services for various other application which actually run over them. There are different types of operating systems like embedded, real-time, distributed, single-user, multi-user, mobile, internet and much more. Full stack web development services develop apps to operate on a mobile operating system like Android and iOS. Some of the key examples of operating systems are as follows: MS Windows macOS Linux iOS Android CentOS Ubuntu Unix

**b. Device Drivers** This type of software controls particular hardware which is essentially attached to the system. Different hardware devices which require a driver to connect to a system easily consist of displays, printers, sound cards, hard disks, keyboard, and mice. Few of the examples of such drivers are: BIOS Driver Motherboard Drivers Display Drivers ROM Drivers Printer Drivers USB Drivers Sound Card Driver VGA Drivers

**c. Firmware** It is actually a permanent software which is embedded in the system's read-only memory. It is essentially a set of instructions which are permanently stored onto the hardware device. It offers vital information regarding how a particular device interacts with different other hardware. Some of the examples of firmware are: Computer Peripherals Embedded Systems UEFI BIOS

**d. Utility** These software are designed to assist in analysing, as well as optimizing, along with configuring and maintaining a given computer system. It provides support to the computer infrastructure. Software like disk cleanup and management tools, anti-viruses, defragmenters, compression tools etc. are all utility software. Some of its examples are: Sign up for Newsletters Check out our popular newsletters and subscribe Norton Antivirus McAfee Antivirus WinRAR WinZip Piriform CCleaner Windows File Explorer Directory Opus Razer Cortex

Classification of Software on the basis of their availability as well as **shareability**

**1. Freeware** These software are available free of cost. A user can easily download them from the internet and can easily use them without paying any charges or fees. However, they don't provide any type of liberty to modify the entire software or charging a fixed fee for its distribution. A best software development company can develop its own freeware to reach out to more customers. Some of the examples of these software are: Adobe Reader Skype ImgBurn Audacity Team Viewer Yahoo Messenger

**2. Shareware** This software is distributed freely to users on a fixed trial basis. It generally comes with a set time limit, and on the expiration of the time limit, the user is finally asked to pay a fixed fee for the continued services. There are different types of shareware such as Freemium, Donationware, Adware, Demoware etc. Few of the examples of shareware are: Adobe Acrobat PHP Debugger WinZip Getright

**3. Open-source** Such types of software are usually available to users along with their source code which means that the user can easily modify and distribute the software as well as add additional features to them. They can either be chargeable or free. Few of the examples of

such software are: Mozilla Firefox Thunderbird GNU Compiler Collection Moodle Apache Web Server

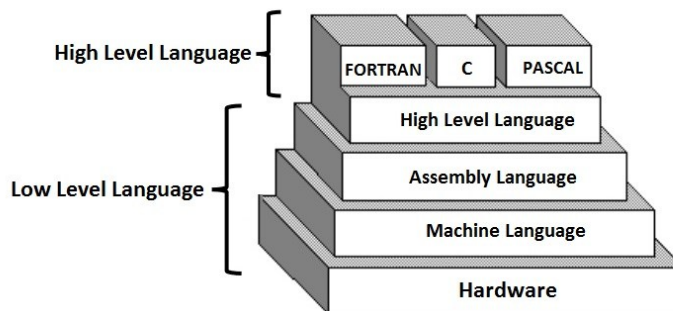
## Computer Languages and Its types

### **Introduction:**

A language is the main medium of communicating between the Computer systems and the most common are the programming languages. As we know a Computer only understands binary numbers that is 0 and 1 to perform various operations but the languages are developed for different types of work on a Computer. A language consists of all the instructions to make a request to the system for processing a task. From the first generation and now fourth generation of the Computers there were several programming languages used to communicate with the Computer. Here we will go in the detail of the Computer language and its types.

### **Computer Language Description:**

A Computer language includes various languages that are used to communicate with a Computer machine. Some of the languages like programming language which is a set of codes or instructions used for communicating the machine. Machine code is also considered as a computer language that can be used for programming. And also HTML which is a computer language or a markup language but not a programming language. Similarly there are different types of languages developed for different types of work to be performed by communicating with the machine. But all the languages that are now available are categorized into two basic types of languages including Low-level language and High level language.



**Computer Language and its Types**

### **Low Level Language:**

Low level languages are the machine codes in which the instructions are given in machine language in the form of 0 and 1 to a Computer system. It is mainly designed to operate and handle all the hardware and instructions set architecture of a Computer. The main function of the Low level language is to operate, manage and manipulate the hardware and system components. There are various programs and applications written in low level languages that are directly executable without any interpretation or translation. The most famous and the base of all programming languages “C” and “C++” are mostly used Low level languages till today. Low level language is also divided into two parts are Machine language and Assembly language.

**Machine Language** is one of the low-level programming languages which is the first generation language developed for communicating with a Computer. It is written in machine code which represents 0 and 1 binary digits inside the Computer string which makes it easy to understand and perform the operations. As we know a Computer system can recognize electric signals so here 0 stands for turning off electric pulse and 1 stands for turning on electric pulse. It is very easy to understand by the Computer and also increases the processing speed.

The main advantage of using Machine language is that there is no need of a translator or interpreter to translate the code, as the Computer directly can understand. But there are some disadvantages also like you have to remember the operation codes, memory address every time you write a program and also hard to find errors in a written program. It is a machine dependent and can be used by a single type of Computer.

**Assembly Language** is the second generation programming language that has almost similar structure and set of commands as Machine language. Instead of using numbers like in Machine languages here we use words or names in English forms and also symbols. The programs that have been written using words, names and symbols in assembly language are converted to machine language using an Assembler. Because a Computer only understands machine code languages that’s why we need an Assembler that can convert the Assembly level language to Machine language so the Computer gets the instruction and responds quickly.

The main disadvantage of this language is that it is written only for a single type of CPU and does not run on any other CPU. But its speed makes it the most used low level language till today which is used by many programmers.

### **High Level Language:**

The high level languages are the most used and also more considered programming languages that helps a programmer to read, write and maintain. It is also the third generation language that is used and also running till now by many programmers. They are

less independent to a particular type of Computer and also require a translator that can convert the high level language to machine language. The translator may be an interpreter and Compiler that helps to convert into binary code for a Computer to understand. There is various high level programming languages like C, FORTRAN or Pascal that are less independent and also enables the programmer to write a program.

The Compiler plays an important role on the Computer as it can convert to machine language and also checks for errors if any before executing. There are several high level languages that were used earlier and also now like COBOL, FORTRAN, BASIC, C, C++, PASCAL, LISP, Ada, Algol, Prolog and Java. It is user-friendly as the programs are written in English using words, symbols, characters, numbers that needs to be converted to machine code for processing.

## Translators

### **Assembler**

In computer science, assembler is a program which converts assembly language into machine code. A computer doesn't understand human languages like English or french, but it deals in a much simpler language called binary language, but a programmer can not write the whole program with its complexity in a binary language therefore we need a program that can convert the human written language (assembly language) into binary language, these softwares are called assemblers.

In assembler, a programmer can write a program into sequence of assembler instructions, the sequence of assembler instruction is known as source code and source program.

### **Compiler**

A Compiler is a program that converts a number of statement of program into binary language, but it is more intelligent than interpreter because it goes through the entire code at once and can tell the possible errors and limits and ranges. But this makes it's operating time a little slower. it is platform-dependent. its help to detect error and get displayed after reading the entire code by compiler.

In other words we can say that, "Compilers turns the high level language to binary language or machine code at only time once", it is known as Compiler.

### **Interpreter**

An interpreter is also a program like a compiler that converts assembly language into binary but an interpreter goes through one line of code at a time and executes it and then goes on to the next line of the code and then the next and keeps going on until there is an error in the line or the code has completed. It is 5 to 25 times faster than a compiler but it stops at the line where error occurs and then again if the next line has an error too, where as a compiler gives all the errors in the code at once. if changes are made on that code which is already compiled then the changed code will need to be compiled and added to compiled code or the entire code need to be re-compiled.

Also, a compiler saves the machine codes for future use permanently but an interpreter doesn't, but an interpreter occupies less memory.

Interpreter is differ from compiler such as,

- Interpreter is faster than compiler.
- It contains less memory.
- Interpreter executes the instructions in to source programming language.

There are several types of interpreter:

- Syntax-directed interpreter
- Threaded interpreter
- Bytecode interpreter

### **Linker**

For a code to run we need to include a header file or a file saved from the library which are pre-defined if they are not included in the beginning of the program then after execution the compiler will generate errors, and the code will not work.

Linker is a program that holds one or more object files which is created by compiler, combines them into one executable file. Linking is implemented at both time, load time and compile time. Compile time is when high level language is turns to machine code and load time is when the code is loaded into the memory by loader.

Linker is of two types:

#### **1. Dynamic Linker:-**

- It is implemented during run time.
- It requires less memory.
- In dynamic linking there are many chances of error and failure chances.
- Linking stored the program in virtual memory to save RAM, So we have need to shared library

#### **2. Static Linker:-**

- It is implemented during compilation of source program.
- It requires more memory.
- Linking is implemented before execution in static linking.
- It is faster and portable.
- In static linking there are less chances to error and No chances to failure.

### **Loader**

Loader is a program that loads machine codes of a program into the system memory. It is part of the OS of the computer that is responsible for loading the program. It is the bare beginning of the execution of a program. Loading a program involves reading the contents of executable file into memory. Only after the program is loaded the operating system starts

the program by passing control to the loaded program code. All the OS that support loading have loader and many have loaders permanently in their memory.

