Introduction to Programming Languages

- A programming language is a vocabulary and set of grammatical rules for instructing a computer or computing device to perform specific tasks.
- The term programming language usually refers to high-level languages, such as C, C++, COBOL, Java, FORTRAN, Adak and Pascal.

Categorization of programming languages

In general, programming languages are categorized in three ways

- Machine Language
- Assembly Language
- High-level Language

Machine Language

- A Computer programming language consisting of binary instructions which a computer can respond to directly.
- Sometimes, it is referred to as **machine** code or object code**. Machine language** is a collection of binary digits or bits that the computer reads.
- A computer cannot directly understand the **programming languages** used to create computer programs, so the program code must be compiled. Example: 01001000, 01100101,01101100, 01101100 etc.

Advantages:

- This language makes fast and efficient use of the computer.
- It requires no translator to translate the code. It is directly understood by the computer5.

Disadvantages:

- All memory addresses have to be remembered.
- All operation codes have to be remembered.

Assembly Language

- An assembly language is a low-level programming language in which there is a very strong correspondence between the program's statements and the architecture's machine code instructions.
- A program written in assembly language consists of a series of mnemonic processor instructions and meta-statements. Assembly language instructions usually consist of an opcode mnemonic followed by a list of data arguments pr parameters.
- These are translated by an assembler into machine language instructions that can be loaded into memory and executed.

- Example: MOV AL 61h;(Meaning-Load AL with 61 hex, MOV is abbreviation of Move)

Advantages Of Assembly Language

- Programs written in machine language are replaceable by memories which are easier to remember.
- Memory Efficient.
- It is not required to keep track of memory locations.
- Faster in speed.
- Easy to make insertions and deletions.
- Hardware Oriented.
- Requires fewer instructions to accomplish the same result.

Disadvantages of Assembly Language

- Long programs written in such languages cannot be executed on small sized computers.
- It take lot of time to code or write the program, as it is more complex in nature.
- Difficult to remember the syntax.
- Lack of portability of program between computers of different makes.

High Level Language

- A high-level language is a programming language that enables development of a program in a much more user-friendly programming context.
- This language is a programming language with storng abstraction about the details of the computer in contrast to low-level programming language (Assembly Language).
- Ex: C, C++, Java
- High level languages are grouped in two categories based on execution model – **compiled** or **interpreted** languages.
- Compiler and interpreter are used to convert the high level language into machine level language. The program written in high level language is known as source program and the corresponding machine level language program is called as object program.
- Compiler read the program at-a-time and searches the error and lists them. If the program is error free than it is converted into object program.
- When program size is large then compiler is preferred. Whereas **interpreter** read only one line of the source code and convert it to object code.

Advantages of High level language

- High level languages are programmer friendly. They are easy to write, debug and maintain.
- It provide higher level of abstraction from machine languages.
- It is machine independent language.

- Easy to learn.
- Less error prone, easy to find and debug errors.
- High level programming results in better programming productivity.

Disadvantages of High level language

- It take additional translation times to translate the source code to machine code.
- High level programs are comparatively slower than low level programs.
- Compared to low level programs, they are generally less memory efficient.
- It can not communicate directly to the hardware.