Computer languages are broadly classified as:

1. Low Level Language:

The term low level means closeness to the way in which machine understand. The low level languages are:

a. Machine Language:

This is the language (in the form of 0's and 1's, called binary numbers) understood directly by the computer. It is machine dependent. It is difficult to learn and even more difficult to write programs.

Advantages

- Machine language requires less memory space than other languages.
- Programs in machine language can be executed directly.
- It does not require any translating program.

Disadvantages

- The programs in machine language are not portable across machines unless the processors are the same as the language is machine dependent.
- It is very difficult to locate and debug errors in machine language. The machine language programmer has to remember all the operation codes, what each code does and how it affects various registers of the processor.
- It requires deep knowledge of the internal structure of the computer and hence only computer experts can program a computer in machine language.

b. Assembly Language:

This is the language where the machine codes comprising of 0's and 1's are substituted by symbolic codes (called mnemonics) to improve their understanding. It is the first step to improve programming structure. Assembly language programming is simpler and less time consuming than machine level programming, it is easier to locate and correct errors in assembly language than in machine language programs. It is also machine dependent. Programmers must have knowledge of the machine on which the program will run.

Advantages

- Assembly language is easier to understand and use than machine language as it employs easy to understand symbols in place of numeric codes.
- It is easier to locate and correct errors in assembly language. Fewer errors occur while coding and these errors can be more easily removed.
- The program in assembly language can be more easily modified which is very difficult in machine language. Recording of the logic is preferred to modifying a program in machine language in case of errors.
- Greater flexibility in writing programs in assembly language matching the computer as all the details of the processor are available to the programmer.

Disadvantages

- It is machine dependent. The programs are not portable across machines as each program is written keeping in mind the specific design features of the machine.
- Programming in assembly language is difficult. It requires expert knowledge of the internal structure of the processor and the assembly language programming techniques.

- Assembly language programming is time consuming.

2. High Level Language

You know that low level language requires extensive knowledge of the hardware since it is machine dependent. To overcome the limitation, high level language has been evolved which uses normal English like, easy to understand statements to solve any problem. Higher level languages are computer independent and programming becomes quite easy and simple. Various high level languages are given below:

- BASIC (Beginners All Purpose Symbolic Instruction Code): It is widely used, easy to learn general purpose language. Mainly used in microcomputers in earlier days.
- COBOL (Common Business Oriented language): A standardized language used for commercial applications.
- FORTRAN (Formula Translation): Developed for solving mathematical and scientific problems. One of the most popular languages among scientific community.
- C: Structured Programming Language used for all purpose such as scientific application, commercial application, developing games etc.
- C++: Popular object oriented programming language, used for general purpose.

Advantages:

- It is easy to learn and use. High-level language is easy to learn and use as it is close to a familiar language like English.
- Portability. Programs written in high level language are portable across machines, as the language is machine-independent.

- Programming efficiency. Programmer productivity is very high as the programmer can code several times more lines of program than in any other language.
- Improved debugging facility. Error detection and removal is easier in high-level language as the program is easily readable and only logical errors need be checked. Syntax errors are detected and displayed by the compiler for correction.
- Fewer errors. Since the programmer is not required to remember all the small steps to be carried out by the machine, fewer errors are likely to be there in the code.
- Programs in high-level languages are easier to maintain than those in the low-level languages.

Disadvantages:

- Lower efficiency. The machine takes more time and main memory to run a program in high-level language. The object program tends to be less efficient in terms of storage utilization and running time.
- Less flexibility. The high level language is less flexible than machine language as the automatic features of the high level languages always occur and are not under the control of the programmer. A large number of rules are to be adhered to in programming.
- High-level language requires translation. A translator program (interpreter or compiler) is required in translating source code into object code.

3.3 Language Processors:

Since a computer hardware is capable of understanding only machine level instructions, so it is necessary to convert the high level language (HLL) into Machine Level Language. There are three Language processors:

A. Compiler: It is translator which converts the HLL language into machine language in one go. A Source program in High Level Language get converted into Object Program in Machine Level Language.

B. Interpreter: It is a translator which converts the HLL language into machine language line by line. It takes one statement of HLL and converts it into machine code which is immediately executed. It eliminate the need of separate compilation/run. However, it is slow in processing as compare to compiler.

C. Assembler: It translate the assembly language into machine code.