Programming Languages:

Machine language:

A programming language in which each instruction is written in binary codes is called Machine Language. An instruction in machine language has two parts: i) OPCODE and ii) OPERANDS. It is a low level language and is machine dependent.

The general form of a machine instruction is OPCODE OPERAND (S)

Assembly Language:

A programming language which uses Mnemonics to write program is known as Assembly Language. In this language mnemonics are used to represent OP codes and symbolic addresses are allowed to express numeric addresses. Assembly language is a low level language and is machine dependent.

An instruction in assembly language consists of three parts, as shown below:
LABEL OPCODE OPERANDS

OPCODE and OPERANDS are mandatory but an instruction may or may not have a label
e.g.

```
01 add al, ah; al = al + ah
02 sub bx, 10; bx = bx - 10
mov ebx, 3; 3 is stored into ebx
reg.
```

`add, sub, mov` are mnemonics.

Comparison between Machine Language & Assembly Language:

Machine Language	Assembly Language
Program is written in binary notations	Program is written with Mnemonics
Difficult to write program	Program development is not that difficult
Computer understand and execute program directly	Translator is required to convert assembly language program into machine language
Program execution is very fast	Program execution is relatively slow

Low Level Language:

A Low Level Language is one in which each instruction is translated into an equivalent single machine code. This language has very simple instruction and is machine dependent.

e.g. Machine language, Assembly language.

High Level Language:

A High Level Language is a programming language which has advanced commands possessing similarity to the commonly used words in English. This language is machine independent and user friendly.

e.g. FORTRAN, C, COBOL, PASCAL.

Comparison between Assembly Language & High Level Language:

Assembly Language	High Level Language
Program is written with the help of Mnemonics	Program is written using simple notations of English language
Program written in this language is converted to machine language by assembler	Program written in this language is converted to machine language by compiler or interpreter
Difficult to understand and read	Relatively easy to understand and read
Debugging is very difficult	Debugging is not that much difficult
Machine dependent	Machine independent
Program execution speed is very high	Program execution speed is relatively slow

Source Program & Object Program:

Programs written in assembly language or any high level language is known as Source Program. A computer only understands machine language and cannot execute any high level language or assembly language program directly. Therefore any source program needs to be translated into machine code for execution. The equivalent program in machine language corresponding to the original source program is called Object program.

Program Translators:

Translators are required to convert source program written in assembly or any high level language into machine code that the computer can execute. There are three types of translators, namely, Assembler, Interpreter and Compiler.

Assembler

Computer cannot understand an assembly language program written using mnemonics, and therefore it must be translated into equivalent machine code for execution. An assembler is thus a special program required to convert an assembly language program into machine form.

Compiler

A Compiler is a program that translates a high level language into machine code. Compiler first stores the source program, scans it and then translates the whole program into equivalent machine language program.

Interpreter

An Interpreter is a program that translates high-level source code into executable machine code. An interpreter first translates one instruction at a time into machine code and executes the same before translating the next instruction. Thus it takes more time for execution.

Difference between Interpreter and Compiler:

Compiler	Interpreter
Translates the entire source code into equivalent machine code at a time	Translates the source code into machine code line by line.
Store compiled code for future usage.	Cannot retain object code.
More useful for commercial purposes.	More useful for learning purpose.
Execution time is less.	Execution time is relatively more.
Slow for debugging	Relatively fast

Some High Level Languages:

FORTRAN: FORTRAN is the abbreviation of FORmula TRANslation. This language has been developed by IBM in 1957. It was standardized by American National Standard Institutes (ANSI) in 1966. This is one of a popular programming language for numerical computations, solving problems in scientific and engineering fields. The presently used upgraded versions of Fortran are Fortran 77, Fortran 90, Fortran 95.

BASIC: BASIC is the short form of Beginners All-purpose Symbolic Instruction Code. This language was developed by John Kemeny and Thomas Kurtz in 1964. It is used for computation purpose, business purpose and various entertaining purpose. BASIC is easy to learn and extensively used in school and college purpose.

C: This language is developed by Brain Kernighan and Dennis Ritchie in early seventies. This language provides the facility of writing structured high level language as well as low level language. C is a most widely used high level programming language at present.