Machine language

Machine language is a low-level language. Machine language is lower than assembly level language in hierarchy. Machine language only uses binary digits (1’s and 0’s). High level language code must be compiling to machine code before execute. Using machine language we can directly communicate with hardware so machine code sometimes called hardware language. It is a bits stream of bits 1’s and 0’s. Which are very hard for humans to understand.

Assembly language

Assembly language is also low- level language and one stem above to machine language. Assembly language can be converting into machine code using assembler. Assembly language can read by human

ADD R0, R1, R2	add content of register R2 & R1 and store the result in register R0
SUB R0, R1, R2	subtract content of R2 from R1 and store the result in register R0
MOVE R0, R1	move content of R1 to register R0

Machine Language	Assembly Language
Machine language is hard to remember for Human. It is a hardware language No need to compiler or interpreter.	Assembly language is written in human readable format. Need assembler to code into machine code
In machine language data only represented with the help of binary format (0s and 1s),	In assembly language data can be represented with the help letters symbols. Move, Add, Sub, End etc.
Execution is very fast	Assembly language is easy to understand by the human. Execution is slow compare to machine code.

Low –level programming language

Generally Low-level programming language indicates machine language and assembly language which are closer to computer hardware which means less abstraction. Use binary digit for instruction.

High –level language

High- level language is more abstract and closer to human readable language. This language consist of letters, numbers, symbols etc. we need compiler or interpreter for convert high level language to machine code or in the form of 1’s and 0’s. Platform independent and portable which means we can write code using any high level language(java, python, c++) and using compiler and interpreter we run it any platform (windows / mac / Linux).

Compilers vs Interpreters

Compiler converts code written in high-level programming into machine code at once before run or execute. It converts entire code into machine code at one without checking the errors and bugs. Ex—c, c++, java

On the other hand interpreter converts high level code into machine code one by one statement. Which means at a time interpreter convert one statement into machine code. And if is something wrong happen I.e. error occur it will stop execution until error id fixed. Ex—python, ruby, javascript.

Compiler is faster interpreter is slower.