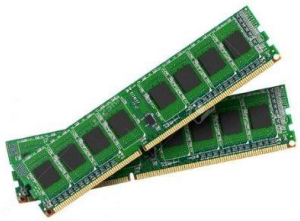
Introduction to Microprocessor

Before you learn any programming language, it is important for one to understand some of the basics about computer and what are the languages that a computer can understand.

Let's have a view on it.

A computer is a collection of hardware components. Let us consider here few hardware components such as:

Microprocessor



RAM

Hard disk

Motherboard

Out of these and many other hardware components, the most important or the heart of the computer is the ***Microprocessor or CPU****.*

Microprocessor or CPU*: A* ***microprocessor*** *is an electronic component that is used by a computer to do its work. It is a central processing unit on a single integrated circuit chip containing millions of very small components including transistors, resistors, and diodes that work together. They are created using a technology called as Semiconductor technology.*

Semiconductor Technology??

Any device which is made up of transistors is referred to as working in Semiconductor Technology.

A ***transistor*** *is a device that regulates current or voltage flow and acts as a switch or gate for electronic signals. The transistors have three terminals emitter, base and collector.*

There are two types of transistors:

1) *NPN transistor.*
2) *PNP transistor.*

Transistors can only store voltages. There are two levels of voltages:
Low Voltage referred to as → 0V
High Voltage referred to as → 5V

If we see the same in Software engineer's view, he/she looks the two levels as:
Low level referred to as → 0
High level referred to as → 1

Therefore, in the perspective of a software engineer a Microprocessor or CPU can understand combinations of 0 and 1.

Programming Languages

Language is the main medium for communicating between the computer systems. A program is a collection of instructions that can be executed by a computer to perform a specific task.

There were several programming languages used to communicate with the Computer.

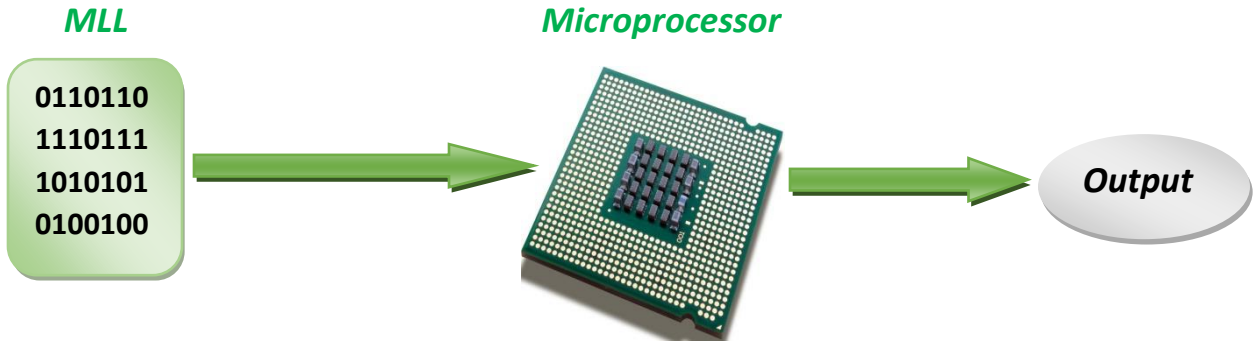
Case-1:

The world's first computer was invented in the year 1940's.
During that time the task of a programmer was not simple.
For example, if they wanted microprocessor to perform any operation then they had to use combinations of 0's and 1's.
During this time all the programs where written in the language called as ***Machine Level Language.*** *It is one of the low-level programming languages.*

The languages that the machines understand are what called as Machine Level Language.

Codes written in 1940's as

To perform addition of two numbers:	**0110110**
To perform subtraction of two numbers:	**1110111**
To perform multiplication of two numbers:	**1010101**
To perform division of two numbers:	**0100100**



The machine level code was taken as input and given to the microprocessor as the machine understands the binary value code and it gives the output.

The main advantage of using Machine language is that there is no need of a translator to translate the code, as the Computer directly can understand.

The disadvantage was, it was difficult for a programmer to write the code or remember the code in this type of language.

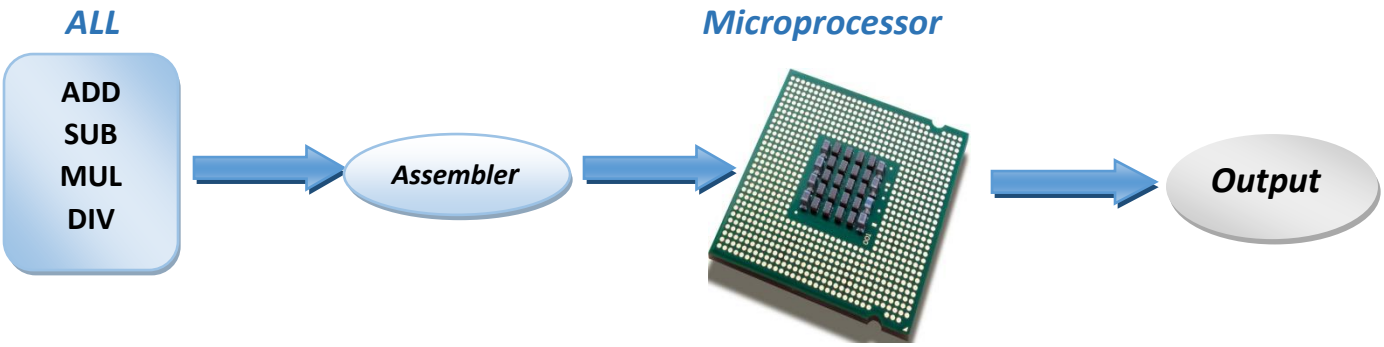
Case-2:

The problem with Machine level code approach was decided to be changed in the year 1950's.

They thought that instead of writing a long sequence of 0's and 1's a single instruction can be given.
For example we use,

	Codes in 1950's written as
To perform addition of two numbers:	**ADD**
To perform subtraction of two numbers:	**SUB**
To perform multiplication of two numbers:	**MUL**
To perform division of two numbers:	**DIV**

*This approach of writing code is what called as **Assembly Level Language**. Instead of using numbers like in Machine languages here we use words or names in English forms.*



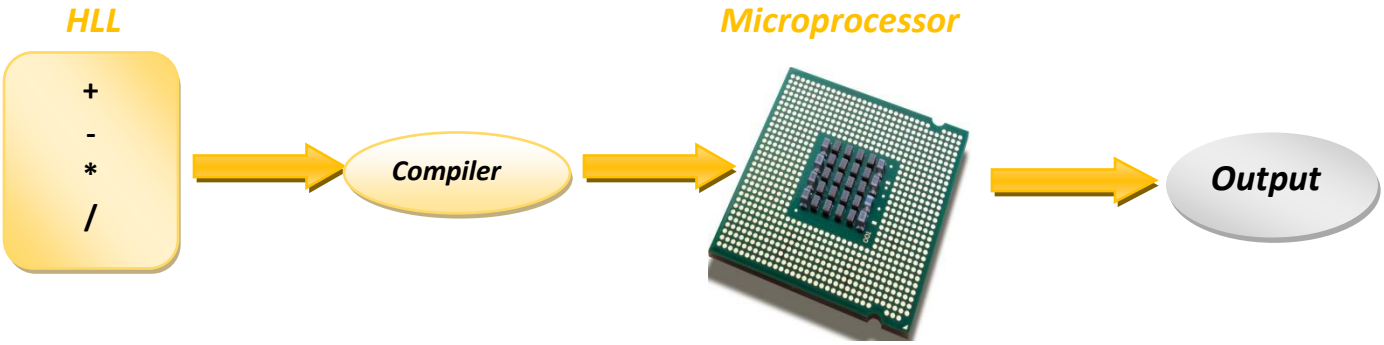
An Assembler is software which takes Assembly Level Language (ALL) programs as input and converts it into Machine Level Language (MLL) program.

Case-3:

People always want the things to be simple and easier so, in 1960's they came up with next type of language called ***High Level Programming Language****.*
High Level Languages are written in a form that is close to our human language, enabling the programmer to just focus on the problem being solved.

For example we use,

	Codes in 1960's written as
To perform addition of two numbers:	**+**
To perform subtraction of two numbers:	**-**
To perform multiplication of two numbers:	*****
To perform division of two numbers:	**/**



A compiler is software which takes High Level Language (HLL) programs as input and converts it into Machine Level Language (MLL) program.