

HLL to Machine Language

- Suppose we want to add two numbers, the instructions that CPU actually carries out might be something like this:-

- Load the number from memory location 2001 into CPU.
- Load the number from memory location 2002 into CPU
- Add two number in CPU
- Store the result in location 2003.

Its complicated as instructions and numbers are represented in binary.

**In High Level Languages, addition can be like
 $c = a+b.$**

But HLL needs to be converted into the language that computer can understand i.e. machine language.

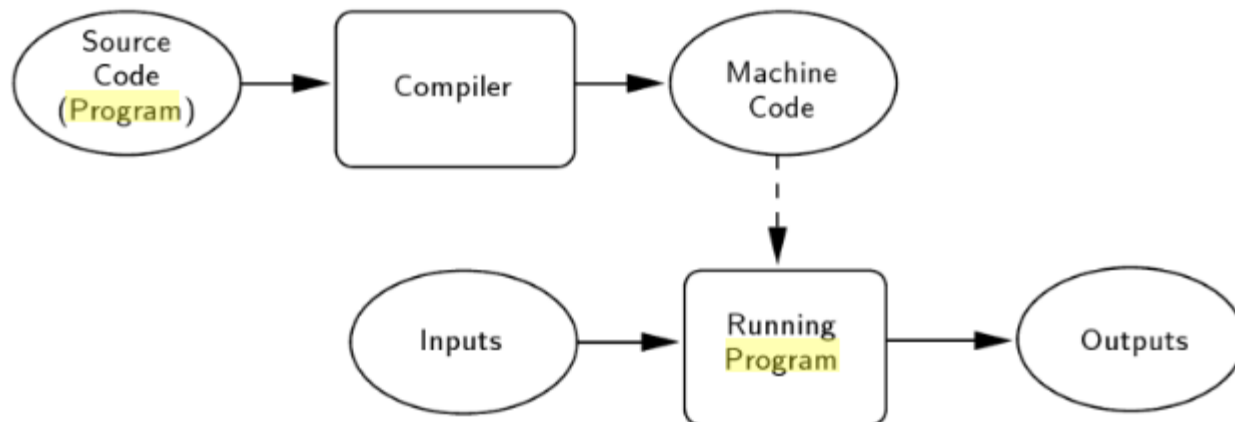
HLL to Machine Language

There are two ways:

HLL can either be compiled or interpreted.

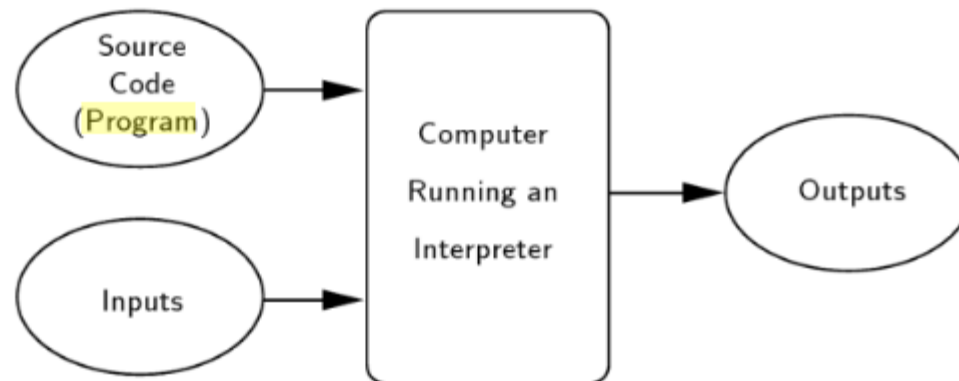
Compiler

- A **compiler** takes entire program and converts it into object code which is typically stored in a file. The object code is also referred as binary code and can be directly executed by the machine after linking. Examples of compiled programming languages are C and C++.



Interpreter

- An **Interpreter** directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code. Rather than translating the source program into a machine language equivalent, the interpreter analyzes and executes the source code instruction by instruction as necessary.
- Examples of interpreted languages are Perl, Python and Matlab.



Difference

- Compiling is a **one-shot translation**, once a program is compiled, it may be run over and over again **without further need of the compiler or the source code**. In the interpreted case, the **interpreter and the source are needed every time the program runs**.
- Compiled programs tend to be faster since the translation is done once and for all, but **interpreted languages** lend themselves to a more **flexible programming environment** as programs can be developed and run interactively.

Python

- Python is a **high-level, interpreted, interactive, object-oriented and dynamic** typed scripting language.
- Python was created by Guido Van Rossum and first released in 1991.

Features of Python

- **Easy to Code:**

- Python is high level programming language. Python is easy to learn language as compared to other languages like C, C#, Java Script , java etc.

- **Free and Open Source:**

- Python language is freely available at official website. Since, it is open-source, this means that the source code is also available to the public.

Features of Python

- **Object-Oriented Language:**
 - Python supports object oriented language features i.e. concept of classes, objects encapsulation etc.
- **GUI Programming Support:**
 - Graphical User Interfaces can be made using module such as PyQt5, Tk in python.
- **High Level Language:**
 - Python is a high level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

Features of Python

- **Extensible feature:**

- Python is extensible language, we can write our some python code into c or C++ language and can also compile that code in C/C++ language.

- **Python is Portable language:**

- If we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run the same code on any platform.

Features of Python

- **Python is integrated language:**
 - We can easily integrate python with other language like C or C++ etc.
- **Interpreted Language:**
 - Python is an interpreted language because python code is executed line by line at a time.
- **Large standard library:**
 - Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

Features of Python

- **Dynamically Typed Language:**
 - That means the type for a variable is decided at run time not in advance because of this feature we don't need to specify the type of variable.