# **First step to learn Programming Language:**

When we think about the programming language then the first word comes in our mind is that what is programming so first we break this in two words “Programming” + “language”.

**Language** is used to establish the communication between two ends and there should be a common language which both ends understands. We can understand this with this diagram example:

John

(English)

Peter

(English)

Message in English

Message in English

**Communication between two humans**

So here we can understand with this diagram that two persons John and Peter are communicating with one language i.e. “English”.

If they both don’t know one common language then we need a translator between them.

­­

John

(English)

Peter

(Spanish)

(English)

**Communication with translator**

Translator

(English & Spanish)

(English)

(Spanish)

(Spanish)

So with this description we have understood that for communication either there should be a common language or translator between two ends.

Now we talk about the machine why we use machine and how machine works. We will try to understand with this example, if we have to solve any problem like addition of two numbers then a human brain easily adds two numbers because here calculation is simple and data size is also small. But when data size is large or calculation is difficult then a normal human being will take long time to solve any mathematical or logical problem. So in this case we pass this problem with some instructions to machine (Computer) and that perform task according to the condition.

Here we are talking about the communication between human and machine. And we can understand that thing with this diagram:

Source code

(Program)

User

(English)

Computer

(Binary)

Compiler /Interpreter

Output

(Binary (0,1))

**Communication between machine and human**

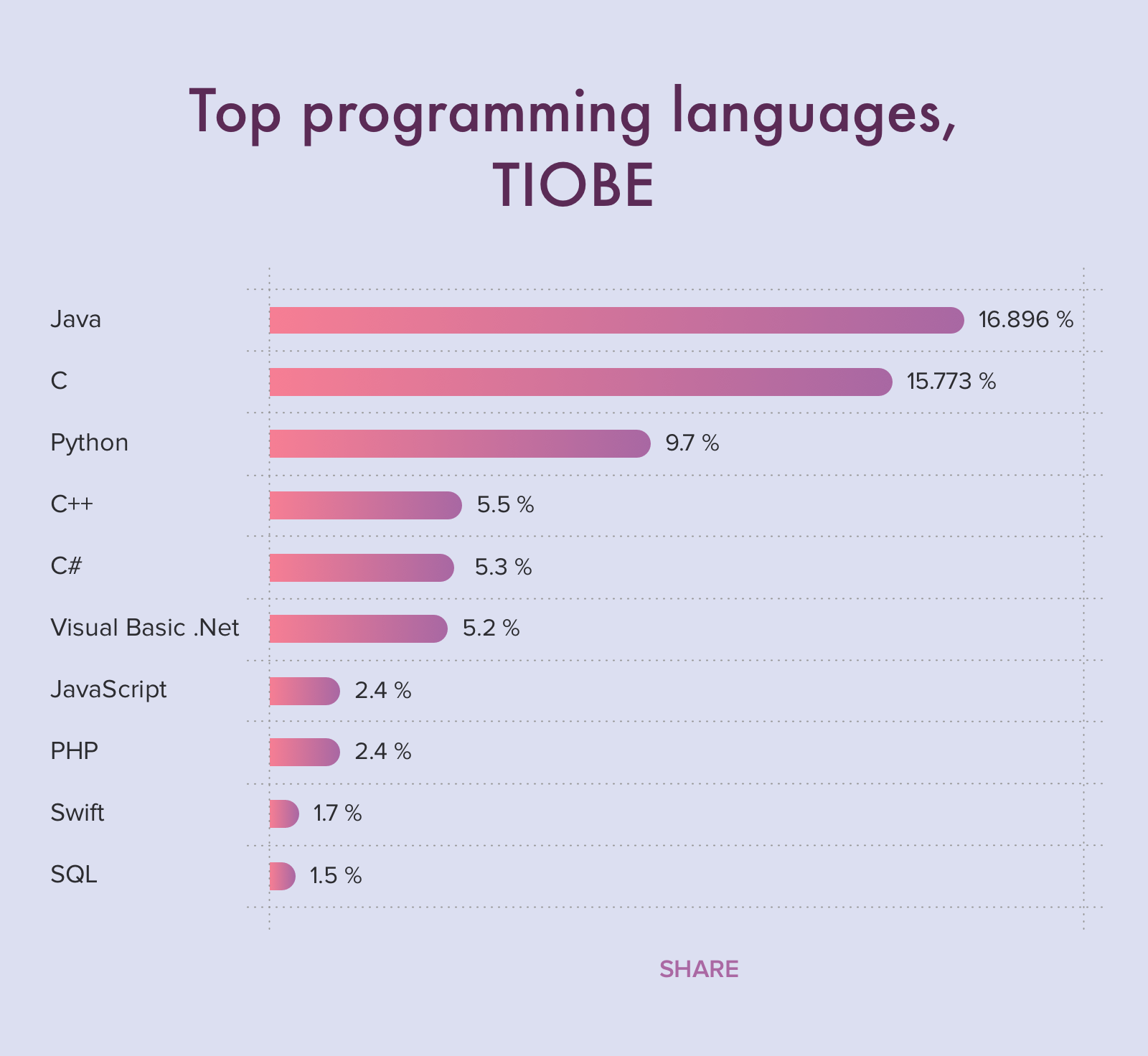
Machine code

In above diagram we can understand that how we pass program to computer (Machine) and compiler and interpreter convert program in binary (0s and 1s) because the machine understands nothing but bits, 1s, and 0s, the combination of which creates meaning.

**Program:** Program is a set of instructions which are passed to machine (Computer) to perform a specific task. Computer read those instructions and perform operations and generate output.

**Programming languages:** This program is written in a programming language. There are dozens of programming languages used in the industry today. These languages allow computers to quickly and efficiently process large and complex swaths of information. But when a program use these programming languages then he/she has to follow syntax. Each programming language has some set of rules and programmer has to do programming under those rules. Each programming language has their own syntax and rules.

According to TIOBE, Java is this and last years’ winner – with just -0.01% change in rating.



## **What is Code?**

The code is pretty much like writing a paragraph of instruction or creating a to-do list to computers. Unlike us humans, the to-do list and instructions you write for the computer has to be extremely detailed and written in some logic.

**Type of programming language:**

## **High Level Languages:**

When we think about computer programmers, we are probably thinking about people who write in high-level programming languages.

High level languages are written in a form that is close to our human language, enabling to programmer to just focus on the problem being solved.

No particular knowledge of the hardware is needed as High level languages create programs that are portable and not tied to a particular computer or microchip.

These programmer friendly languages are called ‘high level’ as they are far removed from the machine code instructions understood by the computer.

Examples include: C++, Java, Pascal, Python, Visual Basic.

**Definition:** Instructions of this language closely resembles to human language or English like words. It uses mathematical notations to perform the task. The high level language is easier to learn. It requires less time to write and is easier to maintain the errors. The high level language is converted into machine language by one of the two different languages translator programs; **interpreter or compiler.**

**Advantages**

* Easier to modify as it uses English like statements
* Easier/faster to write code as it uses English like statements
* Easier to debug during development due to English like statements
* Portable code – not designed to run on just one type of machine

**Assembly language :** Assembly language is also known as low-level language because to design a program programmer requires detailed knowledge of hardware specification. This language uses mnemonics code (symbolic operation code like ‘ADD’ for addition) in place of 0s and 1s. The program is converted into machine code by assembler. The resulting program is referred to as an object code.

**Merits:**

¨       It is makes programming easier than 1GL since it uses mnemonics code for programming. E.g.: ADD for addition, SUB for subtraction, DIV for division, etc.

¨       It makes programming process faster.

¨       Error can be identified much easily compared to 1GL.

¨       It is easier to debug than machine language.

**Demerits:**

¨       Programs written in this language is not directly understandable by computer so translators should be used.

¨       It is hardware dependent language so programmers are forced to think in terms of computer’s architecture rather than to the problem being solved.

¨       Being machine dependent language, programs written in this language are very less or not portable.

¨       Programmers must know its mnemonics codes to perform any task.

**Machine Language (1GL)**

Machine language consists of strings of binary numbers (i.e. 0s and 1s) and it is the only one language, the processor directly understands. Machine language has an Merits of very fast execution speed and efficient use of primary memory.

# Top Programming Terms and Definitions for Beginners

1. Algorithm
2. Compilation
3. Keywords
4. Syntax
5. Identifiers
6. Declaration
7. Exception
8. Hardcode
9. Operators
10. Operands

High level programming model

Machine code

(Low level language)

Run time error

Source code

(High level language)

Compiler / Interpreter

Compile time Error

Output

Low level programming model

**Reference**:

<https://www.computerscience.gcse.guru/theory/high-low-level-languages>

<https://tyrocity.com/topic/types-of-programming-languages/>

<https://hackr.io/blog/programming-terms-definitions-for-beginners>

<https://hackr.io/blog/what-is-programming-language>

<https://www.cleveroad.com/blog/programming-languages-ranking>

<https://www.computerscience.org/resources/computer-programming-languages/>

<https://www.webopedia.com/TERM/P/programming_language.html>