

A Comparison between Python and C++

Ramesh Kumar C, Computer Science and Engineering
Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh
E-mail id - C.RAMESH@Galgotiasuniversity.edu.in

ABSTRACT: *Programming language is nothing but a set of instructions, algorithms which have some set of rules or syntax that governs the language, it acts as a medium of interaction between the machine and a user. C++ and python are one of the most basic and widely used programming languages in this modern era. Both of the languages have their own advantages and disadvantages which are compared and analysed in this paper. On comparing the results python comes out to be a more chosen or selected language because of its easy syntax, easy to read and debug, faster interpretation.*

KEYWORDS: C++, Python, Programming language.

INTRODUCTION

Computers only understand binary language i.e., 0's and 1's. For a computer to understand human written language, it needs to be translated into a computer understandable language first. There is a medium which translates human thoughts or language into machine/computer understandable language which is called a programming language. Programming language is a code, set of instructions, algorithms which help computers in understanding the logic behind the code and compile or interpret it accordingly. Programming languages are used for various purposes such as web development, game development, mobile app development, artificial intelligence and many more.

A programming language can be classified into mainly two types- (a) High level language (b) Low level language[1], which can be further classified into 1st to 5th generations. First and second generation language comes under Low level language and third, fourth, fifth generation language comes under High level language[2].

In this paper main focus will only be on C++ and python that come under the high level language, C++ is an advancement in the C language and it was developed by Bjarne Stroustrup in 1979 at Bell Labs. It is based on object oriented programming system language (oops) concepts utilizes the data abstraction, encapsulation, polymorphism, inheritance, classes and objects in the programming style. C++ is most widely used language in development of embedded systems, web development, mobile applications etc., Python is an interpreted, high-level and general-purpose programming language developed by the Guido van Rossum in 1991. Python is very easy to understand, have a shorter syntax, dynamically typed i.e., one should not have to declare data types, many of the programmers choose python as their first programming language. Python is mainly used in artificial intelligence (AI), data science, machine learning, web scraping, data mining etc. Python is best suitable for a big feed data or a very complex data, it is very easy for python to interpret the complex code.

1. History-

1949	• Assembly Language
1952	• Autocode
1954-57	• FORTRAN
1958	• ALGOL
1959	• COBOL
1959	• LISP
1972-73	• C
1979	• C++
1991	• Python

Fig. 1: History of the programming languages.

Fig. 1 shows the history of some of the most widely used programming languages, it is said that Ada Lovelace's "Algorithm for the Analytical Engine" is the first computer language developed in the late 18's. Its sole motive was to help Charles Babbage with his Bernoulli number computations. Ada's programming was designed to solve any type of problem with severe complexity. However, Charles Babbage only designed the machine but never implemented it.

In 1949 Assembly Language came which was a low level programming language and in 1952 Autocode appeared which was the first programming language which was translated directly into machine understandable language through a compiler[3].

However, all of these languages were not easy to understand and were slow, in the mid 50's a new language came which was called FORTRAN, invented by John Backus. It was the first major language of programming and was used for scientific computing, complex statistical, mathematical, and scientific work. It was the first language of programming to use English as a language and uses data types like real number, the integer, and the Boolean. After Fortran there came many languages in the 1950's such as ALGOL, LISP, COBOL[3], [4].

Algol (1958) short for Algorithmic Language was invented in 1958 for scientific purposes and came as a building block for Java, C, C++, and Pascal. It was the first language to use the block structure[3].

COBOL short for Common Business Oriented Language was developed by Grace Murray Hopper in 1959. It was designed by keeping in mind that it can be run on all different types of computers and for the business sector. It is still in existence in various applications such as debit cards, ATMs, etc[3].

LISP short for List Processing was designed in 1959 primarily for Artificial Intelligence research. It is one of the oldest high-level programming languages and is still in existence where Python is used[3].

C programming language was the very first high-level language designed or invented by Dennis Ritchie and Brian Kernighan at Bell Laboratories in early 1970's. It was the most widely used programming language which was compiler based, it is a sequential or a procedural language which is easy to learn and understand. It was influenced mainly by the two languages: B language and the BCPL language[5].

However, there were some disadvantages of using C as a programming language such as time consuming, very slow calculations, missing of an oops concept, low level abstraction and lack of exception handling. All these disadvantages were overcome by the extension of the C language called C++ language, where ++ meant as a feature addition in the predecessor C. It was developed by Bjarne Stroustrup in 1979 at Bell

Laboratories. C++ is a general purpose object oriented programming language and implements the data abstraction, encapsulation, polymorphism, inheritance, classes and objects. It supports at least seven different styles of programming[5].

Python is an interpreted, interactive, object-oriented programming language, first designed during 1885-1990 and was released in 1991 by Guido van Rossum. It supports functional, structural and oops concepts, it is more user friendly and easy to read, interpret as it uses English language with minimum syntactical and punctuations[6].

2. Classification of the programming language

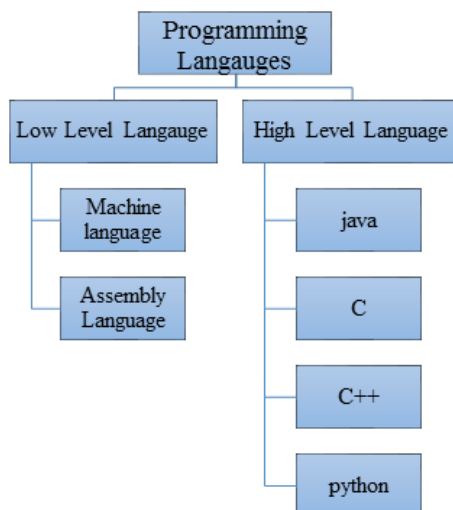


Fig. 2: Classification of Programming languages

Fig. 2 shows that the programming languages can be classified into two types: low level language and high level language.

2.1 Low level language

Low level programming language is a type of programming language which consists of a set of instructions or code perceived by the computer machine. It is basically of two types - machine language and assembly language. Humans cannot understand the low level language as it is written in machine language (i.e., binary language). Low level language is very fast to implement and high memory efficient. It is also called a programming language which is very close to the hardware, as it is usually used to program features like drivers, kernels, boot setup etc. Low level programming language can be converted directly into the machine language without any requirement of a compiler interpreter[1].

2.1.1 Machine language-

Machine language is the lowest level of programming language, it contains only binary codes and hexadecimal values which are easily understandable and recognized by the machine or computer. Ex of machine language M will be written as a 01001101 in binary language and as 4D in hexadecimal[1].

2.1.2 Assembly language-

Assembly language is higher than machine code in hierarchy. As it gets quite complex and difficult to understand and write the code in the machine language this is where assembly language comes into play as it uses a one liner code with human understandable words to write a program. Instructions comprises ADD (add), and SUB (subtract), MOV (move) etc. These instructions do basic operations, such as addition, subtraction, moving values into storage registers etc. Assembly language can be converted to the machine language using an assembler. Assembly language can be easily translated into machine level language[1].

2.2 High Level Language

High level language is a type of programming language that enables humans to understand and interpret the language easily. It contains a syntax which is much closer to the human languages. A machine can only understand a binary language, which is quite complex and difficult to understand. High level languages are easy to debug and translated easily into machine language using translators such as compiler or interpreter. Various examples of high level languages are java, C, C++, python, ALGOL, COBOL, FORTRAN etc.[2][1].

In this paper the main focus is on comparing and analyzing mainly two high level programming languages: C++ and Python. Various parameters are taken to compare such as speed, syntax, code length, memory allocation, portability.

Research Methodology

1. Parameters required for the comparison of the programming languages-

- Speed- the execution and the run time of the programs were recorded and analyzed to see which one is faster in execution of the program.
- Readability- how easy it is to read and understand the code, syntaxes of the program. High level language is easy to understand as it uses more human understandable syntax and words rather than machine code.
- Code length- how long or short the program is? How many lines of code is required to write a particular program?
- Portability - in simple terms means that whether the code can be used in different platforms i.e., cross platform or not.

2. System configuration & Software Requirements-

System Configuration -

Processor - Intel i3- 4130 @3.40 GHZ

Ram- 8.00 GB

System type - 64 Bit

Software requirement -

Turbo C++, python 3, or online compiler tools.

3. Program code-

3.1. Program in C++

```
#include <iostream>
using namespace std;
int main()
{
    int x, y, z;
    cout << "1st number: ";
```

```

cin >> x;

cout << endl;
cout << "2nd number: ";
cin >> y;

cout << endl;
z = x + y;
cout << "Hello World" << endl;
cout << x << "+" << y << "=" << z;
return 0;
}

```

3.2. Program in python

```

x = input("Enter 1st number: ")
y = input("Enter 2nd number: ")
z = int(x) + int(y)
print("Hello World")
print(x,"+",y,"=",z)

```

RESULTS AND DISCUSSION

1. Speed

Readings of the both the programs were taken and recorded at least 10 times and shown in Table 1.

Table 1: Observation Table

S.No.	Time taken by C++ program (seconds)	Time taken by Python program (seconds)
1.	1.377751	1.48864
2.	1.45481	1.51373
3.	1.390862	1.40313
4.	1.477951	1.71306
5.	1.481000	1.58068
6.	1.491163	1.67518
7.	1.490740	1.86223
8.	1.363457	1.44859
9.	1.318106	1.51199
10.	1.332107	1.60832
Mean	1.4177947	1.580555

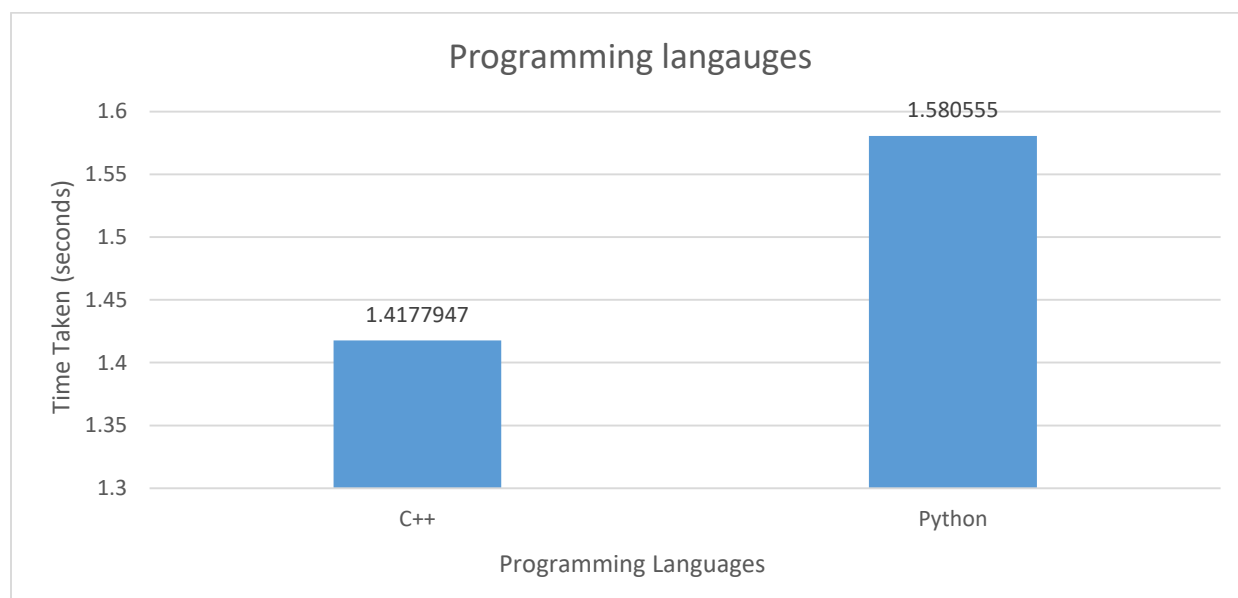


Fig. 3: Graph showing the time taken by both the programming languages in seconds to execute a program.

By comparing the mean result of both the languages in Fig. 3. It is clear that the time taken by the C++ is comparatively less than the python. However, results may vary according to the length of the program and the system configuration (Table 2).

- Readability** - It is quite visible with general observation that the python program is easier to understand as it is much closer or similar to the human language while the C++ has some predefined syntax and structure which makes the C++ code a bit more complex to understand.
- Code Length** - Comparing the code lines of both the program it is evident that python has only 5 lines of code while the C++ has 16 lines. Python code for addition of two numbers and printing Hello World is relatively shorter than the C++ code.
- Portability** - C++ is pre compiled to the machine and it uses a compiler to compile the code and thus it is not possible to run the same code on different machine or operating system while the python uses an interpreter to execute and run the code and thus it can be executed on almost every machine and operating system.

Table 2: Result Check List

Parameter	C++	Python
Speed	✓	
Readability		✓
Code length		✓
Portability		✓

CONCLUSION

In this paper a comparative analysis between two most widely used programming languages C++ & Python has been done with respect to some parameters; which are execution time speed, easy to read and understand the code, length of the code and its cross platform dependency. C++ is the extension of the C language. It is fast, static in nature, and also preferable in game development and embedded systems. However, there are some disadvantages of using C++ such as it does not support cross platform, have complex syntaxes and structure, code length is also quite large. Python is more efficient while doing artificial intelligence, machine learning, web scraping etc. Python is trending nowadays due to its shorter code length, syntax and easy to understand structure. It was concluded that python has more advantages over the C++ but as all the factors

were not included while doing a comparative analysis a debate can arise about why C++ is not comparatively better than python. At the end it's all about personal preference and choice, there is no such perfect programming language which can fulfill each and every purpose of the programming.

REFERENCES

1. Programming Languages: Low Level, machine language, Assembly, High-level Language (HLL); Algorithm and Flow chart: Concept and Uses.
2. High-Level Programming Languages. Nell Dale & John Lewis (adaptation by Michael Goldwasser) - PDF Free Download, <https://docplayer.net/14079788-High-level-programming-languages-nell-dale-john-lewis-adaptation-by-michael-goldwasser.html>, Sep. 2020.
3. Zuse, K., "CMSC 331. Some material UMBC CMSC 331 2 CMSC 331. Some material The 1940s: Von Neumann and Zuse," 1998.
4. Skog, K., "FROM BINARY STRINGS TO VISUAL PROGRAMMING A Nordic perspective on history of programming and programming languages."
5. Stroustrup: FAQ, https://www.stroustrup.com/bs_faq.html#invention, Sep. 2020.
6. Sanner, M.F., Python: A programming language for software integration and development, *J. Mol. Graph. Model.*, 1999.

