AN INSIGHT TO C++

Presented By,

Amina Khan

HISTORY OF C++

- C++ is a multi-paradigm programming language that supports object oriented programming (OOP) created by Bjarne Stroustrup in 1983 at Bell labs, C++ is an extension of C programming and the programs written in C language can run in C++ compiler.
- The development of C++ actually began four years before its release, in 1979. It did not start with the name C++. Its first name was C with classes.

- In the late part of 1983, C with classes was first used for AT&T's internal programming needs. Its name was changed to C++ later in the same year.
- It is of course also used in a wide range of other application domains, notable graphics programming. C++ supports inheritance through class derivation. Dynamic binding is provided by Virtual class function.

DIFFERENCE BETWEEN C AND C++

C	C++
C is Procedural Language.	C++ is non-Procedural i.e. Object oriented Language.
Top down approach is used in Program Design.	Bottom up approach adopted in Program Design.
Multiple Declaration of global variables are allowed.	Multiple Declaration of global variables are not allowed.
C requires all the variables to be defined at the starting of a scope.	C++ allows the declaration of variable anywhere in the scope i.e. at time of its First use.
In C, malloc () and calloc () Functions are used for Memory Allocation and free () function for memory Deallocating.	In C++, new and delete operators are used for Memory Allocating and Deallocating.

USES OF C++ LANGUAGE

- C++ is used by programmers to develop computer software
- It is used to create general system software
- Used to build drivers for various computer devices
- Software for servers and software for specific applications
- Used in the creation of video games.

ADVANTAGE OF C++

- ✓ C++ is relatively-low level and is a systems programming language.
- ✓ It has a large community.
- ✓ It has a relatively clear and mature standard.
- ✓ Modularity'
- ✓ Reusability and readability

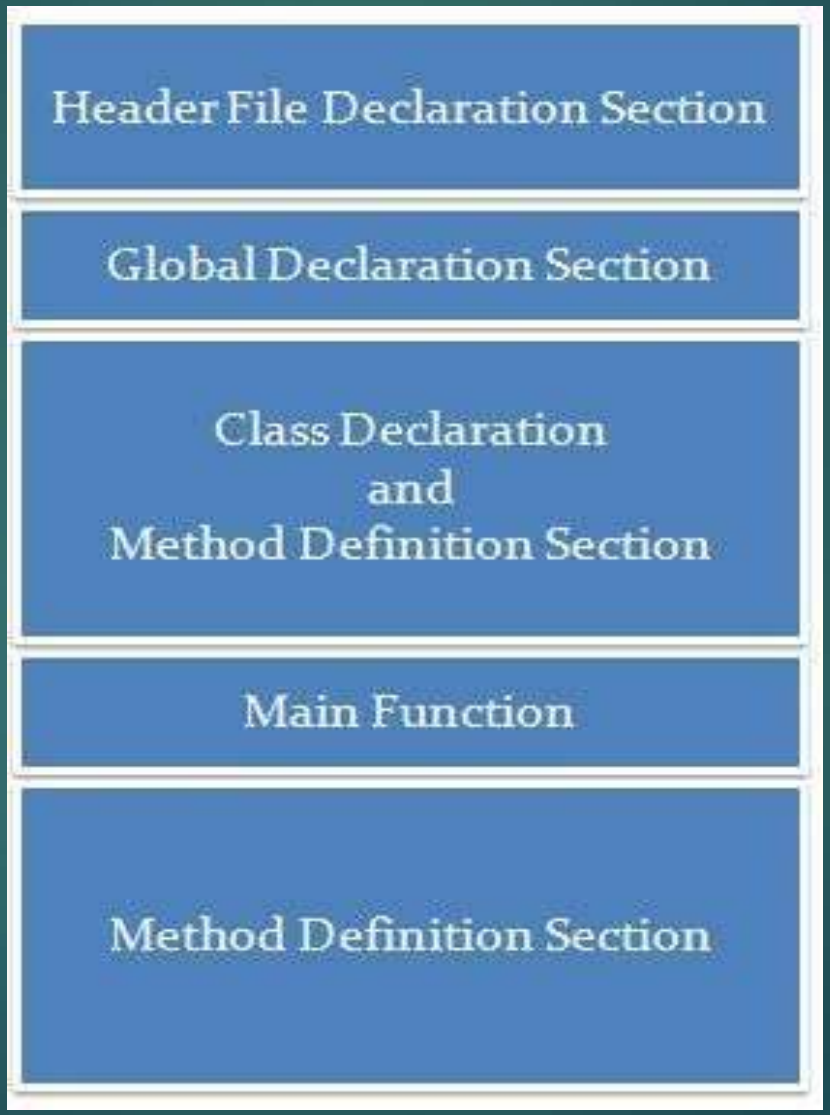
DISADVANTAGE OF C++

- Data is global or local.
- It emphasis on instructions bur not on data.
- It can be generally heavy if not careful.
- Data is global and global data does not have security.

STANDARD LIBRARIES

- The C++ Standard Library can be categories into two parts:
- **The standard function library:** This library consists of general-purpose, stand-alone function that are not part of any class. The function library is inherited from C.
- **The object oriented class library:** This is a collection of classes and associated function.
- Standard C++ library incorporates all the standard C libraries also, with small additions and changes to support type safety.

STRUCTURE


Header File Declaration Section
Global Declaration Section
Class Declaration
and
Method Definition Section
Main Function
Method Definition Section

SIMPLE PROGRAM C++

```
#include<iostream.h>/*Header File*/
int main()/*Main Function*/
{
cout<<"\n*HELLO*\n";
/*Output Statements*/
}
```

C++ DATA TYPES

Primary data type	int, float, char, void
User defined data type	structure, union, class, enumeration
Derived data type	array, function, pointer, reference

C++ VARIABLES SCOPE

- A scope is a region of the program and broadly speaking there are three places, where variables can be declared –
- Inside a function or a block which is called **local variables**,
- In the definition of function parameters which is called **formal parameters**.
- Outside of all functions which is called **global variables**.

LOCAL VARIABLES

```
#include <iostream.h>
 int main ()
 {
        int a, b;
        int c;
                    // Local variable declaration
        a = 10;
        b = 20;
                    // actual initialization
 c = a + b;
cout << c;
return 0;
 }
```

Output = ?

Output = 30

GLOBAL VARIABLES

```
#include <iostream.h>
        // Global variable declaration:
                Int g;
int main ()
{
        // Local variable declaration:
                int a, b;
         // actual initialization
                a = 10;
                b = 20;
   g = a + b;
   cout << g;
   return 0;
}
```

Output = ?

Output = 30

OPERATORS

- Arithmetic operators
- Relational operators
- Logical operators
- Bitwise operators
- Assignment operators

BITWISE OPERATOR

Bitwise operator works on bits and perform bit-by-bit operation.

P	Q	P&Q	P|Q	P^Q
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Assume if A = 60; and B = 13; now in binary format they will be as follows:

A = 0011 1100 ----> **Binary Number for 60**

B = 0000 1101 ----> **Binary Number for 13**

A&B = 0000 1100

A|B = 0011 1101

A^B = 0011 0001

~A = 1100 0011

```
#include <iostream.h>
int main() {
int a = 7; // a = 111
int b = 5; // b =  101
cout << "Bitwise Operators\n";
cout << "a & b = " << (a&b) << "\n";
cout << "a | b = " << (a|b) << "\n";
cout << "a ^ b = " << (a^b) << "\n";
cout << "~a = " << (~a) << "\n";
cout << "~b = " << (~b) <<  "\n";
cout << "a >> b = " << (a>>b) << "\n";
cout << "a << b = " << (a<<b) <<  "\n";
}
```

Output = ?

Output

Bitwise Operators
a & b = 5
a | b = 7
a ^ b = 2
~a = -8
~b = -6
a >> b = 0
a << b = 224

ASSIGNMENT OPERATOR

An assignment operator, in the context of the C programming language, is a basic component denoted as "=".

```
int x = 25;
x = 50;
```

FUNCTIONS

- Function is a set of statements to perform some task.
- Every C++ program has at least one function, which is **main()**, and all the most trivial programs can define additional functions.
- **Syntax of Function**

```
return-type function-name (parameters)
{
 // function-body
}
```

DECLARING, DEFINING AND CALLING FUNCTION

```
#include < iostream.h>

int sum (int x, int y);    //declaring function

int main()

{

 int a = 10;

 int b = 20;

 int c = sum (a, b);    //calling function

 cout << c;

}

int sum (int x, int y)    //defining function

{

 return (x+ y);

}
```

Output = ?

Output = 30

CALLING A FUNCTION

- Functions are called by their names. If the function is without argument, it can be called directly using its name. But for functions with arguments, we have two ways to call them,

- **Call by Value**
- **Call by Reference**

ARRAYS

- Array is defined as a set of homogeneous data items. An Array is a group of elements that share a common name that are differentiated from one another by their positions within the array.
- It is a data structure which allows a collective name to be given to a group of elements which all have the same type.

Syntax

datatype arrayname[array size];

- The Array which is declared as above is called single-dimension array

- Example: **float salary[10];**

  float → data type

  salary → array name

  [10] → array size(integer)

- The size of an array must be an integer constant and the data type can be any valid C++ data type.

C++ ARRAY IN DETAIL

CONCEPT	DESCRIPTION
Multi-dimensional arrays	C++ supports multidimensional arrays. The simplest form of the multidimensional array is the two-dimensional array.
Pointer to an array	You can generate a pointer to the first element of an array by simply specifying the array name, without any index.
Passing arrays to functions	You can pass to the function a pointer to an array by specifying the array's name without an index.
Return array from functions	C++ allows a function to return an array.

POINTERS

- Pointer is a user defined data type which creates special types of variables which can hold the address of primitive data type like char, int, float, double or user defined data type like function, pointer etc. or derived data type like array, structure, union, enum.

What Are Pointers?

- A pointer is a variable whose value is the address of another variable. Like any variable or constant, you must declare a pointer before you can work with it.

Syntax

datatype *var-name;

```
int   *ip;          // pointer to an integer
double              // pointer to a double
*dp;                // pointer to a float
float *fp;          // pointer to character
char
      *ch
```

C++ Vs Python: The Best Ever Comparison Between OOPs

When we talk about the general-purpose programming languages, then the first two languages that come into our mind is C++ and Python. There are a lot of differences between C++ vs Python. The only similarity between these two is the general-purpose languages.Let’s start with a short introduction to C++. We can say that it is the successor of the C language with object-oriented capabilities. On the other hand, Python is the most popular high-level programming language in the world.

Uses of C++

C++ is one of the most popular and widely used general-purpose programming languages. It is used to develop complex systems where the hardware level coding requires. But C++ is not portable like other high-level programming languages.

Uses Of C++

It is used to produce almost every embedded system. e.g., smartwatches, stopwatches, multimedia systems in automobiles, electric cars systems, IoT devices, etc

We can also develop the servers and the high-performance microcontroller programs. Microcontroller programs can be found in factories.

Game development is the key to C++. That is the reason C++ is becoming more popular among game developers.

We can create 3D games, multiplayer, and many more types of games using C++..

Python

Python is the most powerful general-purpose programming language. It is widely used in machine learning in Big Data technologies. Python was created in late 1991 by Guido van Rossum, the dutch programmer. It is the only programming language in the world that has the best code readability. When the Python was created, it was built with the motive to make its programming most straightforward than any programming language.

Uses Of Python

- PYTHON IS WIDELY USED WITH MACHINE LEARNING. IN OTHER WORDS, PYTHON CODE IS COMPATIBLE WITH MACHINE LEARNING.
- PYTHON HAS THE ABILITY TO MANAGE THE MASSIVE AMOUNT OF DATA WITH EASE AND COST-EFFECTIVE WAY.
- PYTHON IS ALSO USEFUL IN BIG DATA TECHNOLOGIES. MOST OF THE SIGNIFICANT DATA FUNCTIONS CAN BE PERFORMED USING PYTHON PROGRAMMING.

C++ Vs Python

1..CODE LENGTH

The code written in C++ becomes more lengthy than Python. But wait C++ offers the fast prototyping features.

You can write simple code in Python and make your code much shorter.

2. .SPEED

The language has to be fast to run the real-time systems. C++ is faster than Python. Python is slower because it is also written in C language.

3. .READABILITY

C++ also lacks the indentation rules. Thus the code is like garbage in some points.

On the other hand, Python has more English like syntax.

C++ Vs Python

C++

```
class
HelloWorld{    public:
      void
PrintHelloCalltutor( )
   {          std::cout
<< “Hello
Calltutor!\ n”;      }};
```

Python

```
print(“Hello Calltutor!”)
```

Declaration

C++

If we want to use any data type in C++. Then we need to declare the data type each time we use. Therefore we do have the proper command over when we have declared the datatype.

Python

On the other hand, if we want to write the code in Python, then we need not declare the type of data. Therefore the length of the code becomes shorter and easier to maintain. For example, in C++ code the data type will declare as int a=10, and in Python a=10

Special Functions

C++ is known as the intermediate level programming language. Apart from that, it also has all the features of the object-oriented programming language.

On the other hand, Python is the most simple and powerful high-level programming language. It has some inbuilt constructor for executing clear code for large and small scale. It is also an object-oriented programming language.

Python Vs C++ Syntax

C++ has complex syntax as compared with Python. It requires the semicolons and brackets to work. You can't run the C++ code without using semicolons and brackets. On the other hand, Python offers a user-friendly approach to code into it. The entire structure of Python programming depends on the code indentation.

Career Aspect

C++ CAREER OPPORTUNITY

- Junior Programmer
- Software Developer
- Quality Analyst
- Game Programmer
- Software Developer
- Engineer
- Programming Architect
- UNIX Shell Scripting

CAREER IN PYTHON

- Python Developer
- Product Manager
- Data Analyst
- Educator
- Financial Advisors
- Data Journalist

Companies

C++

- Google
- Twitch
- Accenture
- Pubu
- Yummly
- QuizUp
- Yext
- FiveStars

Python

- Instagram
- Google
- Spotify
- Netflix
- Uber
- Dropbox
- Pinteres

Conclusion (C++ Vs Python)

CONCLUSION (C++ VS PYTHON)IF WE TALK ABOUT THE CONCLUSION FROM THE ABOVE DISCUSSION ON C++ VS PYTHON, WE GET THAT C++ IS QUITE FASTER THAN THE PYTHON. THUS IT IS THE BEST OPTION TO BUILD REAL- TIME APPLICATIONS.

Both C++ and Python are having a decent future scope. But if you want to be more than just a developer, then you should opt for Python. Otherwise, you can choose to C++ to become the C++ developer. We do have both C++ programming experts and Python programming experts. They have plenty of years of experience in their fields. Therefore they can provide you the best C++ programming help and Python programming help. Along with they are also capable of offering you the best C++ assignment help and C++ homework help.