Introduction to C++

C++ is a general-purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a **compiled** language.

- C++ is a high-level, general-purpose programming language designed for system and application programming.
- One of the key features of C++ is its ability to support low-level, system-level programming, making it suitable for developing operating systems, device drivers, and other system software.

- C++ has a large, active community of developers and users, and a wealth of resources and tools available for learning and using the language.

❖ Applications of C++:

C++ finds varied usage in applications such as:

- ✓ Operating Systems & Systems Programming. e.g. *Linux-based OS (Ubuntu etc.)*
- ✓ Browsers *(Chrome & Firefox)*
- ✓ Graphics & Game engines *(Photoshop, Blender, Unreal-Engine)*
- ✓ Database Engines *(MySQL, MongoDB, Redis etc.)*
- ✓ Cloud/Distributed Systems

❖ Features of C++

- ### Simple

C++ is a simple language because it provides a structured approach (to break the problem into parts), a rich set of library functions, data types, etc.

- ### Abstract Data types

In C++, complex data types called Abstract Data Types (ADT) can be created using classes.

- ### Portable

C++ is a portable language and programs made in it can be run on different machines.

- ### Mid-level / Intermediate programming language

C++ includes both low-level programming and high-level language so it is known as a mid-level and intermediate programming language. It is used to develop system applications such as kernel, driver, etc.

- ### Structured programming language

C++ is a structured programming language. In this we can divide the program into several parts using functions.

- ### Rich Library

C++ provides a lot of inbuilt functions that make the development fast. Following are the libraries used in C++ programming are:

  - **<iostream>**
  - **<cmath>**
  - **<cstdlib>**
  - **<fstream>**

- ## Memory Management

C++ provides very efficient management techniques. The various memory management operators help save the memory and improve the program's efficiency. These operators allocate and deallocate memory at run time. Some common memory management operators available C++ are new, delete etc.

- ## Pointer

C++ provides the feature of pointers. We can use pointers for memory, structures, functions, array, etc. We can directly interact with the memory by using the pointers.

- ## Object-Oriented

In C++, object-oriented concepts like data hiding, encapsulation, and data abstraction can easily be implemented using keyword class, private, public, and protected access specifiers. Object-oriented makes development and maintenance easier.

- ## Compiler based

C++ is a compiler-based programming language, which means no C++ program can be executed without compilation. C++ compiler is easily available, and it requires very little space for storage. First, we need to compile our program using a compiler, and then we can execute our program.

- ## Reusability

With the use of inheritance of functions programs written in C++ can be reused in any other program of C++. You can save program parts into library files and invoke them in your next programming projects simply by including the library files.

- ## Errors are easily detected

It is easier to maintain a C++ programs as errors can be easily located and rectified. It also provides a feature called exception handling to support error handling in your program.

The list of arguments of every function call is typed checked during compilation. If there is a type mismatch between actual and formal arguments, implicit conversion is applied if possible.

❖ History of C++

- **C++ programming language** was developed in 1980 by Bjarne Stroustrup at bell laboratories of AT&T (American Telephone & Telegraph), located in U.S.A.
- **Bjarne Stroustrup** is known as the **founder of C++ language.**
- It was develop for adding a feature of **OOP (Object Oriented Programming)** in C without significantly changing the C component.
- C++ programming is "relative" (called a superset) of C, it means any valid C program is also a valid C++ program

Let's see the programming languages that were developed before C++ language.

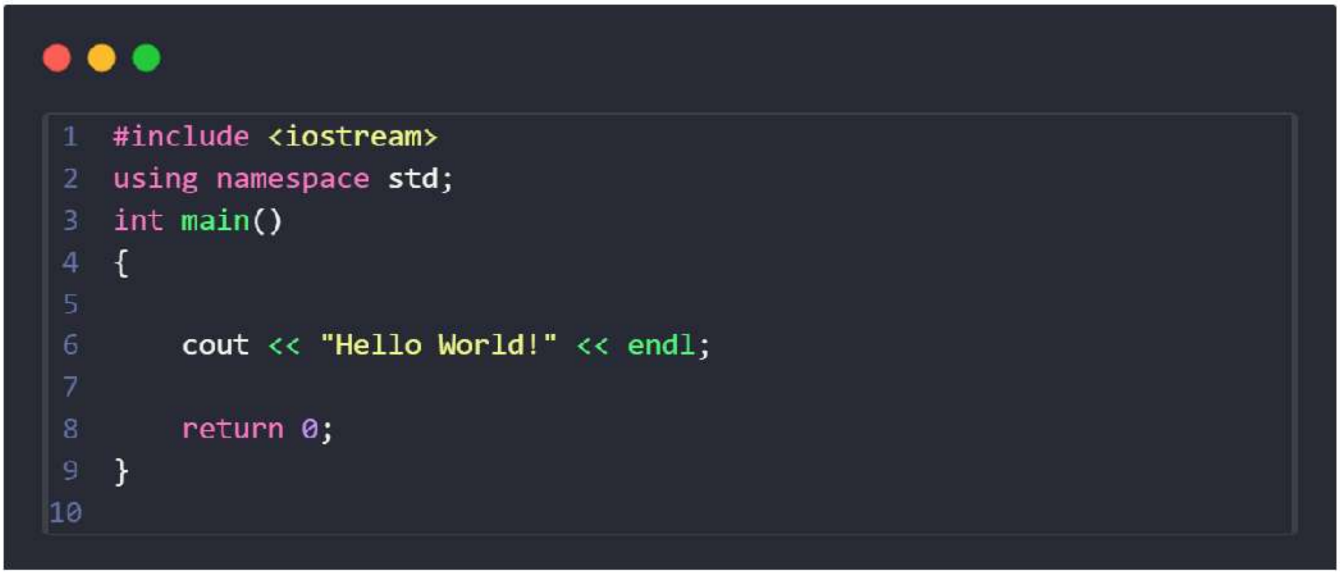
Language	Year	Developed By
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie
C++	1980	Bjarne Stroustrup

❖ Difference between C and C++

No.	C	C++
1)	C follows the **procedural style programming.**	C++ is multi-paradigm. It supports both **procedural and object oriented.**
2)	Data is less secured in C.	In C++, you can use modifiers for class members to make it inaccessible for outside users.
3)	C follows the **top-down approach.**	C++ follows the **bottom-up approach.**
4)	C does not support function overloading.	C++ supports function overloading.
5)	In C, you can't use functions in structure.	In C++, you can use functions in structure.
6)	C does not support reference variables.	C++ supports reference variables.
7)	In C, **scanf()** **and** **printf()** are mainly used for input/output.	C++ mainly uses stream **cin and cout** to perform input and output operations.
8)	Operator overloading is not possible in C.	Operator overloading is possible in C++.
9)	C programs are divided into **procedures and modules**	C++ programs are divided into **functions and classes.**
10)	C does not provide the feature of namespace.	C++ supports the feature of namespace.

11)	Exception handling is not easy in C. It has to perform using other functions.	C++ provides exception handling using Try and Catch block.
12)	C does not support the inheritance.	C++ supports inheritance.

Let’s see the First Program of C++



```cpp
#include <iostream>
using namespace std;
int main()
{

    cout << "Hello World!" << endl;

    return 0;
}

```

❖ Advantages of C++:

- Performance: C++ is a compiled language, which means that its code is compiled into machine-readable code, making it one of the fastest programming languages.
- Object-Oriented Programming: C++ supports object-oriented programming, which makes it easier to write and maintain large, complex applications.
- Standard Template Library (STL): The STL provides a wide range of algorithms and data structures for working with data, making it easier to write efficient and effective code.

- Machine Independent: C++ is not tied to any hardware or processor. If the compiler compiles the program in the system, it will be able to run no matter what the hardware is.
- Large Community: C++ has a large, active community of developers and users, providing a wealth of resources and support for learning and using the language.

❖ Disadvantages of C++:

- Steep Learning Curve: C++ can be challenging to learn, especially for beginners, due to its complexity and the number of concepts that need to be understood.
- Verbose Syntax: C++ has a verbose syntax, which can make code longer and more difficult to read and maintain.
- Error-Prone: C++ provides low-level access to system resources, which can lead to subtle errors that are difficult to detect and fix.