Introduction to C++

- **C++** is a middle-level programming language developed by Bjarne Stroustrup starting in 1979 at Bell Labs.
- **C++** runs on a variety of platforms, such as Windows, Mac OS, and the various versions of UNIX.
- C++ is used to create computer programs, and is one of the most used language in game development.

What is C++

- C++ is a general purpose, case-sensitive, free-form programming language that supports object-oriented, procedural and generic programming.
- C++ is a middle-level language, as it encapsulates both high and low level language features.

Object-Oriented Programming (OOPs)

C++ supports the object-oriented programming, the four major pillar of object-oriented programming (OOPs) used in C++ are:

- Inheritance
- Polymorphism
- Encapsulation
- Abstraction

- **Objects:**

  Any entity that has state and behavior is known as an object, for example, tables, pen etc.
  - It can be defined as an instance of class, it contains an address and takes up some space in memory.
  - They can communicate with each other without knowing the details of each other's data or code.

1. **Class:**
    - Collection of multiple objects is called class, It is a blueprint from which you can create an individual object.
    - They represent broad categories that share attributes.
2. **Inheritance:**
    - When one class acquires all the properties and behaviors of a parent object, it is known as inheritance.
    - It provides code reusability.
3. **Polymorphism:**
    - Polymorphism means having many forms, It is the ability of an object to take on many forms**.**
4. **Abstraction:**
    - It is the property by virtue of which only the essential details are displayed to the user, the non-essential details are hidden from the end users.
5. **Encapsulation:**
    - It is defined as the wrapping up of data under a single unit.
    - It is the mechanism that binds together code and the data it manipulates.

- **There are many advantages of OOPs like reusability, data redundancy, code maintenance, security, better productivity, and design benefits.**

Applications of OOPs

- Real Time Systems
- Client Server System
- Hypertext and Hypermedia
- Object Oriented Database
- Neural Networks and Parallel Programming
- AI Expert Systems

- Simulation and Modeling System
- Office Automation Systems
- CIM/CAD/CAM Systems
- Computer Aided Designs

C++ Features

- C++ is a widely used programming language.
- It provides a lot of features that are given below.

1) 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.

2) Abstract Data types

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

3) Portable

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

4) 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.

5) Structured programming language

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

6) 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>

7) 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.

8) Quicker Compilation

- C++ programs tend to be compact and run quickly.
- Hence the compilation and execution time of the C++ language is fast.

9) 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.

10) Recursion

- In C++, we can call the function within the function.
- It provides code reusability for every function.

11) Extensible

C++ programs can easily be extended as it is very easy to add new features into the existing program.

12) 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.

13) 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.

14) 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.

15) Redefine Existing Operators

- C++ allows the programmer to redefine the meaning of existing operators such as +, -.
- **For Example,** The "+" operator can be used for adding two numbers and concatenating two strings.

C++ Basic Input/Output

- C++ I/O operation is using the stream concept. Stream is the sequence of bytes or flow of data.
- It makes the performance fast.
- If bytes flow from main memory to device like printer, display screen, or a network connection, etc, this is called as **output operation.**
- If bytes flow from device like printer, display screen, or a network connection, etc to main memory, this is called as **input operation.**

I/O Library Header Files

Let us see the common header files used in C++ programming are:

Standard output stream (cout)

- The **cout** is a predefined object of **ostream** class.
- It is connected with the standard output device, which is usually a display screen.
- The cout is used in conjunction with stream **insertion operator** (<<) to display the output on a console
- Let's see the simple example of standard output stream (cout):

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

   cout << "Welcome" << endl;
   cout << " TO SYBBA" << endl;

```

8. }

Output:

```
Welcome
TO SYBBA
```

Standard input stream (cin)

- The **cin** is a predefined object of **istream** class.
- It is connected with the standard input device, which is usually a keyboard.
- The cin is used in conjunction with stream **extraction operator** (>>) to read the input from a console.
- Let's see the simple example of standard input stream (cin):

```
    #include <iostream>
using namespace std;
      int main( ) {
       int age;
        cout << "Enter your age: ";
       cin >> age;
       cout << "Your age is: " << age << endl;
    }
```

Output:

```
Enter your age:22
Your age is: 22
```

Standard end line (endl)

Let's see the simple example of standard end line (endl):

```
#include <iostream>
using namespace std;
int main( ) {
cout << "C++ Tutorial";
cout << " Javatpoint"<<endl;
cout << "End of line"<<endl;
}
```

Output:

```
C++ Tutorial Javatpoint
End of line
```

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	C++ supports the feature of namespace.

	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.

Simple C++ program

```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World!";
  return 0;
}
```

Example explained

Line 1: #include <iostream> is a **header file library** that lets us work with input and output objects, such as cout (used in line 5).

Header files add functionality to C++ programs.

Line 2: using namespace std means that we can use names for objects and variables from the standard library.

Don't worry if you don't understand how #include <iostream> and using namespace std works.

Just think of it as something that (almost) always appears in your program.

Line 3: A blank line. C++ ignores white space.

But we use it to make the code more readable.

Line 4:

- Another thing that always appear in a C++ program, is int main().
- This is called a **function**.
- Any code inside its curly brackets {} will be executed.

Line 5:

- cout (pronounced "see-out") is an **object** used together with the *insertion operator* (<<) to output/print text.
- In our example it will output "Hello World!".
- **Note:** Every C++ statement ends with a semicolon ;.
- **Note:** The body of int main() could also been written as:
  int main () { cout << "Hello World! "; return 0; }

Remember:

- The compiler ignores white spaces.
- However, multiple lines makes the code more readable.

Line 6: return 0 ends the main function.

Line 7: Do not forget to add the closing curly bracket } to actually end the main function.

Omitting Namespace

You might see some C++ programs that runs without the standard namespace library.

The using namespace std line can be omitted and replaced with the std keyword, followed by the :: operator for some objects:

Example

```
#include <iostream>

int main() {
  std::cout << "Hello World!";
  return 0;
}
```

Output :Hello World !