

Introduction to C++

C++, as we all know is an extension to C language and was developed by **Bjarne Stroustrup** . Bjarne combined C and simula and created a more powerful programming language that support Object Oriented Programming features called C++. C++ was initially named as “C with classes”. C++ is an intermediate level language, as it comprises a confirmation of both high level and low level language features.

C++ is an Object Oriented Programming language but is not purely Object Oriented. Its features like Friend and Virtual, violate some of the very important OOPS features.

Some Facts:

- Simula is considered the first object-oriented programming language.
- Smalltalk is considered the first truly object-oriented programming language.
- The popular object-oriented languages are Java, C#, PHP, Python, C++, etc.

C vs C++

C	C++
C is a structural programming language.	C++ is both structural and object oriented programming language.
C follows top down approach.	C++ follows bottom up approach.
C doesn't support virtual function.	C++ supports virtual function.
C doesn't support Object Oriented features.	C++ supports Object Oriented Features.

Operator overloading and function overloading is not possible in C.	C++ supports operator overloading and function overloading.
Data are less secured in C.	Data are more secured in C++.
Namespaces aren't available in C.	Namespaces are available in C++.
Exception handling is not supported.	Exception handling is supported.
scanf and printf are used for input and output.	cin and cout are used for input and output.
File extension in C is .c.	File extension in C++ is .cpp.
In C, main function can be called from other function.	In C++, main function cannot be called from other function.
C is middle level language.	C++ is intermediate level language.

Object Oriented Programming (OOP)

Object Oriented Programming (OOP) is a programming paradigm which is based on the concept of object and class. Object oriented programming has several features such as inheritance, polymorphism, abstraction, encapsulation, message passing, dynamic binding etc.

Features of OOP (Characteristics)

1. Object

Object means a real-world entity such as a pen, chair, table, computer, watch, etc. In other word, Any entity that has state and behavior is known as an object. An object is a instance of a class. Objects take space in memory.



Example: A dog is an object because it has states like color, name, breed, etc. as well as behaviors like wagging the tail, barking, eating, etc.

2. Class

A class can also be defined as a blueprint from which you can create an individual object. It is the collection of objects of similar type. Unlike object, class doesn't consume any space in memory. A class can have any number of objects associated to it.

An example of class can be student, smart phone, animal , vehicle etc.

3. Inheritance

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. The new object can have its own properties. It provides code reusability.

For example, a child **inherits** the traits of his/her parents.

4. Polymorphism

Polymorphism means ability to take more than one form. Operator overloading and function overloading are generally used to achieve polymorphism.

For example, a function with same name can perform different task.

5. Abstraction

Representin essential features without including background details is called abstraction. In other words, hiding internal details and showing functionality is known as abstraction.

For example phone call, we don't know the internal processing.

6. Encapsulation

Wrapping of data and function into a single unit is called encapsulation. Data isn't accessible from outside. Only those functions which are wrapped in class can have access to these data.



Capsule

For example capsule, it is wrapped with different medicines.

7. Message passing

Objects communicate with each other by sending and receiving information . This is called message passing.

Procedural Vs Object Oriented Programming

In order to overcome the problem of unstructured programming language, procedural programming language came in use. In this approach a new idea came up and a set of execution code was kept in a place and it was called function or procedure. A procedure call is used to invoke the procedure. After the sequence is processed, flow of control proceeds to right after the position where the call was made.

Some of the differences between procedural programming language and OOP language are as follows;

Procedural language	OOP language
In procedural programming, program is divided into smaller parts called function.	In OOP, program is divided into smaller parts called objects.
It doesn't have any mechanism of hiding data so it is less secured.	It provides data hiding as a result it is more secure.
It focuses on procedure rather than data.	It focuses on data rather than procedure.
Data moves openly around the system from function to function.	Object can communicate with each other through functions.
It is a top-down approach.	It is a bottom-up approach.
It needs less memory.	It needs more memory.

It doesn't support any overloading.	It supports operator and function overloading.
Examples: C, Fortran	Examples: C++, Java, Python

Advantages of OOP

1. OOP makes development and maintenance easier .
2. OOP allows code reusability with the help of inheritance.
3. OOP provides the features of data encapsulation and data abstraction.
4. OOP is more secured as it provides data hiding features.
5. OOPs provides the ability to simulate real-world event much more effectively.
6. Object oriented systems can be easily upgraded from small to large systems.
7. Software complexity can be well managed with OOP concept.
8. Message passing techniques for communication between objects makes the system much easier.

Application of OOP

Main application areas of OOP are:

- Real Time Systems such as aircraft control, nuclear power plant.
- Simulation and Modeling
- Object oriented databases
- AI and Expert System
- Neural Networks and parallel programming
- Decision support and office automation systems etc.