

# CHAPTER-1

## **Evolution of programming paradigm**

1. Procedure oriented programming Vs. Object-oriented programming (OOP)
2. C++ Features
- 3. Object oriented programming concepts**
  - a) Objects
  - b) classes
  - c) encapsulation and abstraction
  - d) inheritance
  - e) polymorphism
  - f) dynamic binding
  - g) message passing

# CPP history

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.

## Procedure oriented programming Vs. Object-oriented programming

BASIS FOR COMPARISON	POP	OOP
Basic	Procedure/Structure oriented .	Object oriented.
Approach	Top-down.	Bottom-up.
Basis	Main focus is on "how to get the task done" i.e. on the procedure or structure of a program .	Main focus is on 'data security'. Hence, only objects are permitted to access the entities of a class.
Division	Large program is divided into units called functions.	Entire program is divided into objects.
Entity accessing mode. (secure data)	No access specifier observed.	Access specifier are "public", "private", "protected".
Inheritance	There is no provision of inheritance.	Inheritance achieved in three modes public private and protected.

## Procedure oriented programming Vs. Object-oriented programming

Overloading/Polymorphism	Neither it overload functions nor operators.	It overloads functions, constructors, and operators.
Data sharing	Global data is shared among the functions in the program.	Data is shared among the objects through the member functions.
Friend functions/classes	No concept of friend function.	Classes or function can become a friend of another class with the keyword "friend". Note: "friend" keyword is used only in c++
Virtual classes/ function	No concept of virtual classes .	Concept of virtual function appear during inheritance.

## Procedure oriented programming Vs. Object-oriented programming

Data hiding & security	There is no proper way of hiding the data, so data is insecure	Data is hidden in three modes public, private, and protected. hence data security increases.
Example	C, VB, FORTRAN, Pascal	C++, JAVA, VB.NET, C#.NET.

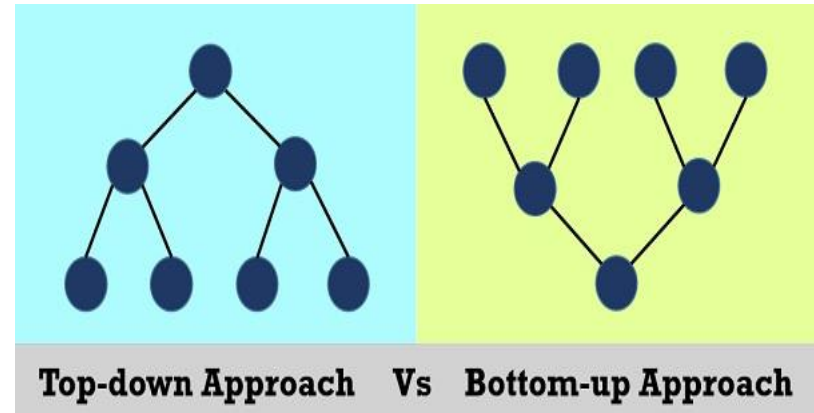
## Difference between top-down and bottom-up approach

The algorithms are designed using two approaches that are the top-down and bottom-up approach.

In the top-down approach, the complex module is divided into submodules.

On the other hand, bottom-up approach begins with elementary modules and then combine them further.

The prior purpose of an algorithm is to operate the data comprised in the data structure. In other words, an algorithm is used to perform the operations on the data inside the data structures.



## **Advantage of OOPs over Procedure-oriented programming language**

OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.

OOPs provide data hiding whereas in Procedure-oriented programming language a global data can be accessed from anywhere.

OOPs provide ability to simulate real-world event much more effectively. We can provide the solution of real word problem if we are using the Object-Oriented Programming language.

# **C++ OOPs Concepts**

The major purpose of C++ programming is to introduce the concept of object orientation to the C programming language

## **OOPs (Object Oriented Programming System)**

Object means a real word entity such as pen, chair, table etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

### **C++ OOPs concepts**

Object

Class

Inheritance

Polymorphism

Abstraction

Encapsulation

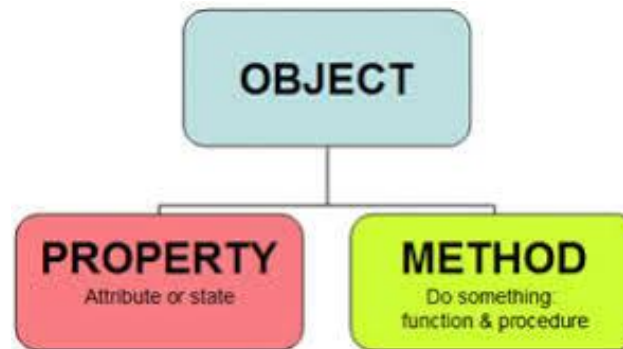
Dynamic Binding

Message Passing



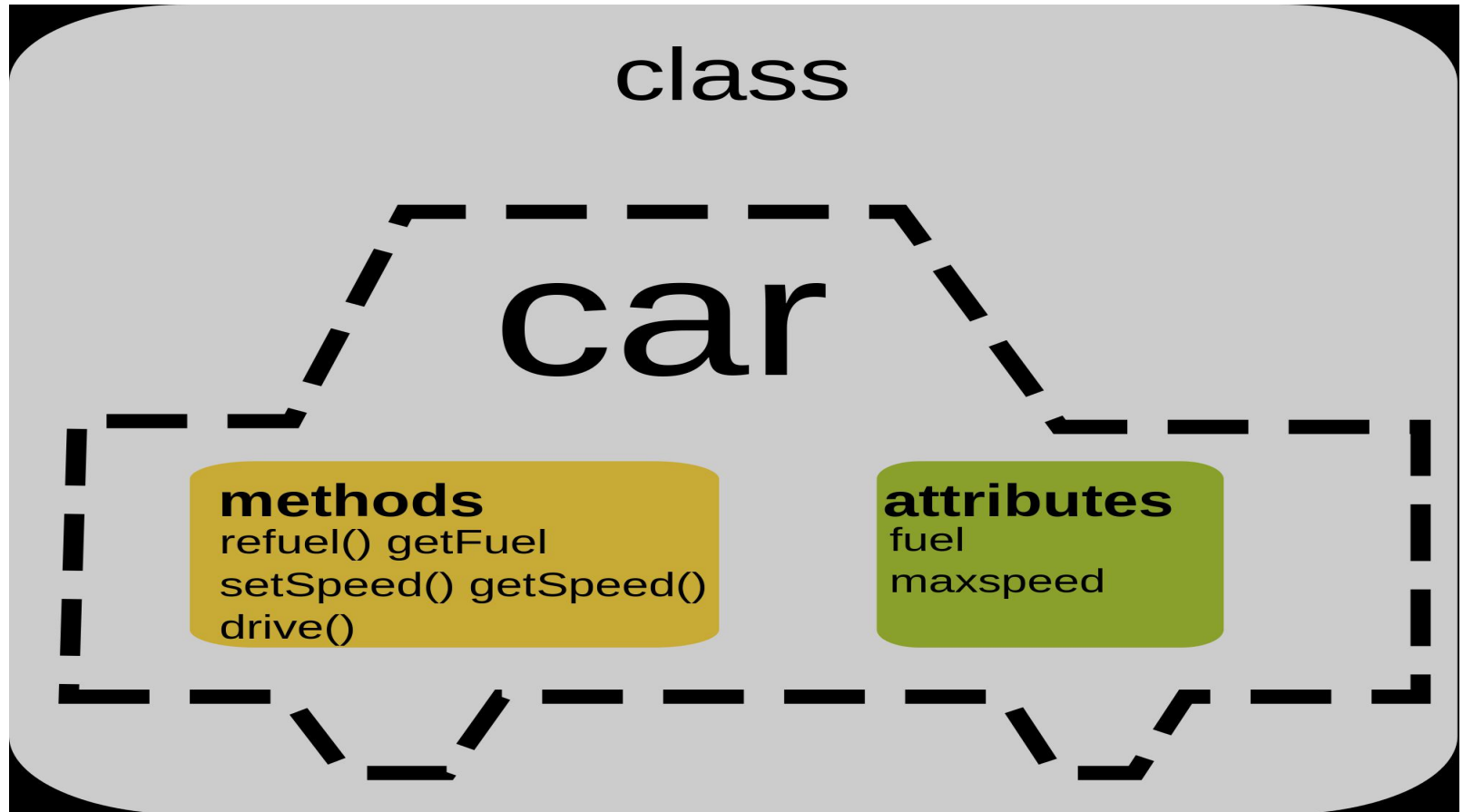
## Object:

Any entity that has state and behavior is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.



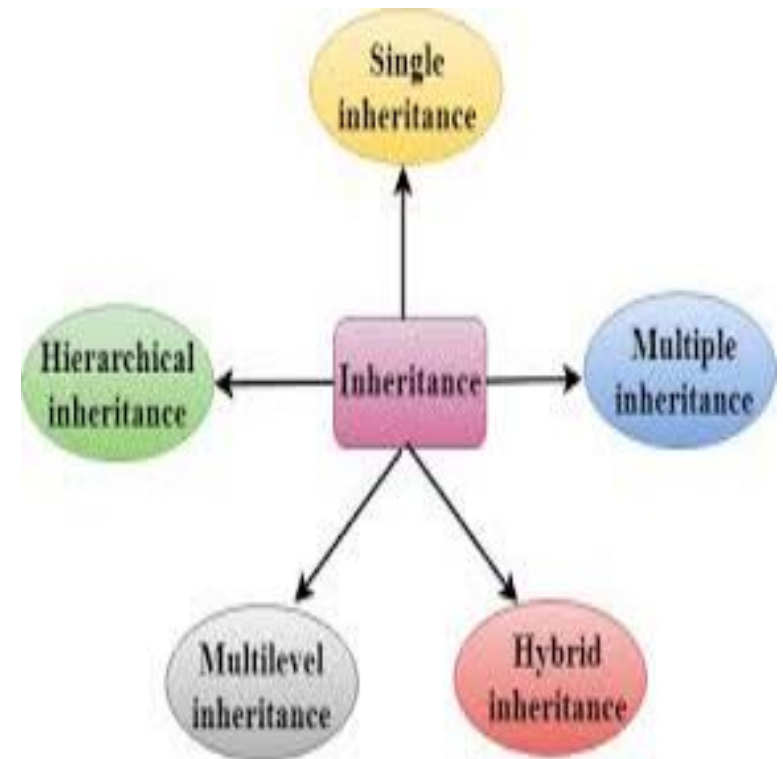
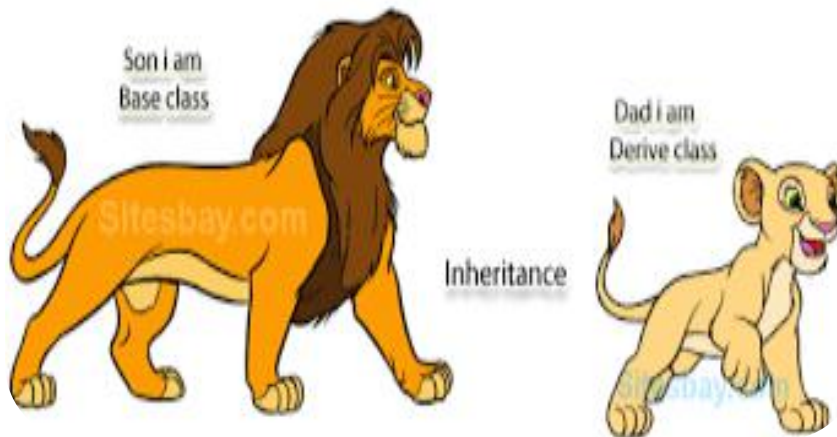
# Class:

Collection of objects is called class. It is a logical entity. It contains data members and member functions



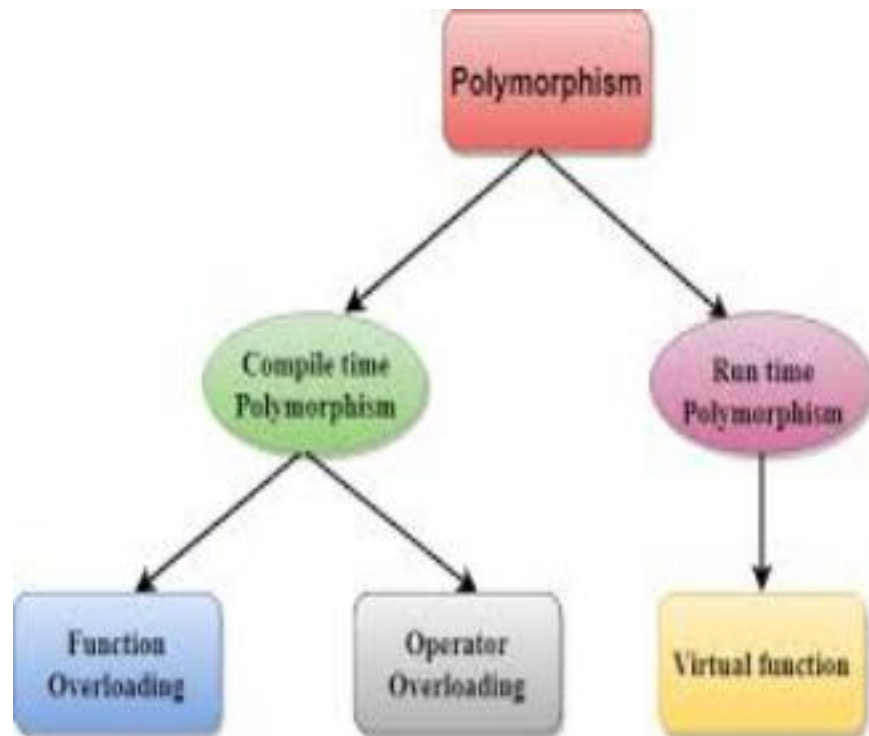
# Inheritance

When one object acquires all the properties and behaviours of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism

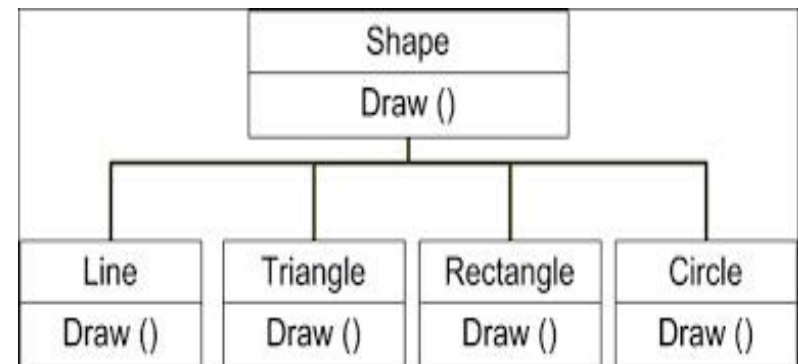


**Polymorphism:** When one task is performed by different ways i.e. known as polymorphism. For example: to convince the customer differently, to draw something e.g. shape or rectangle etc.

In C++, we use Function overloading and Function overriding to achieve polymorphism.



In Shopping malls behave like Customer  
In Bus behave like Passenger  
In School behave like Student  
At Home behave like Son Sitesbay.com



# Abstraction

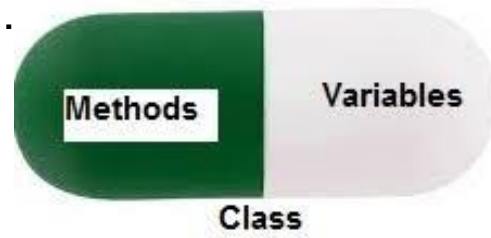
Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.


In C++, we use abstract class and interface to achieve abstraction.

## Encapsulation

Binding (or wrapping) code and data together into a single unit is known as encapsulation. For example: capsule, it is wrapped with different medicines.

### Encapsulation in C++






Sitesbay.com

Real Life Example of Abstraction

1. You know only how to drive a car but don't internal details how car built.
2. Simple example is mobile phone, using mobile we make call and chat but don't know how it work.
3. Another Example is ATM machine you know only how to operate machine but don't have any knowledge about its internal working process.

Sitesbay.com



**Encapsulation:**

For the mathematical equation, shown in the figure assume that complex functions are required -  $\Rightarrow$  But I the end we obtain result for it.

Calculator shows the result of equation but hides the implementation (calculating the result) involved

**Abstraction:**

The calculator shown in the figure has to be powered by a battery source. How the battery module works for the calculator is not necessary to know for the user who uses the calculator.

Using Battery module along with other modules we use calculator  $\Rightarrow$  Thus using Abstraction encapsulation is performed

## **Dynamic Binding:**

Late binding in C++ In C++, late binding (also called "dynamic binding") normally happens when the virtual keyword is used in a method's declaration. C++ then creates a so-called virtual table, which is a look-up table for such functions that will always be consulted when they are called.

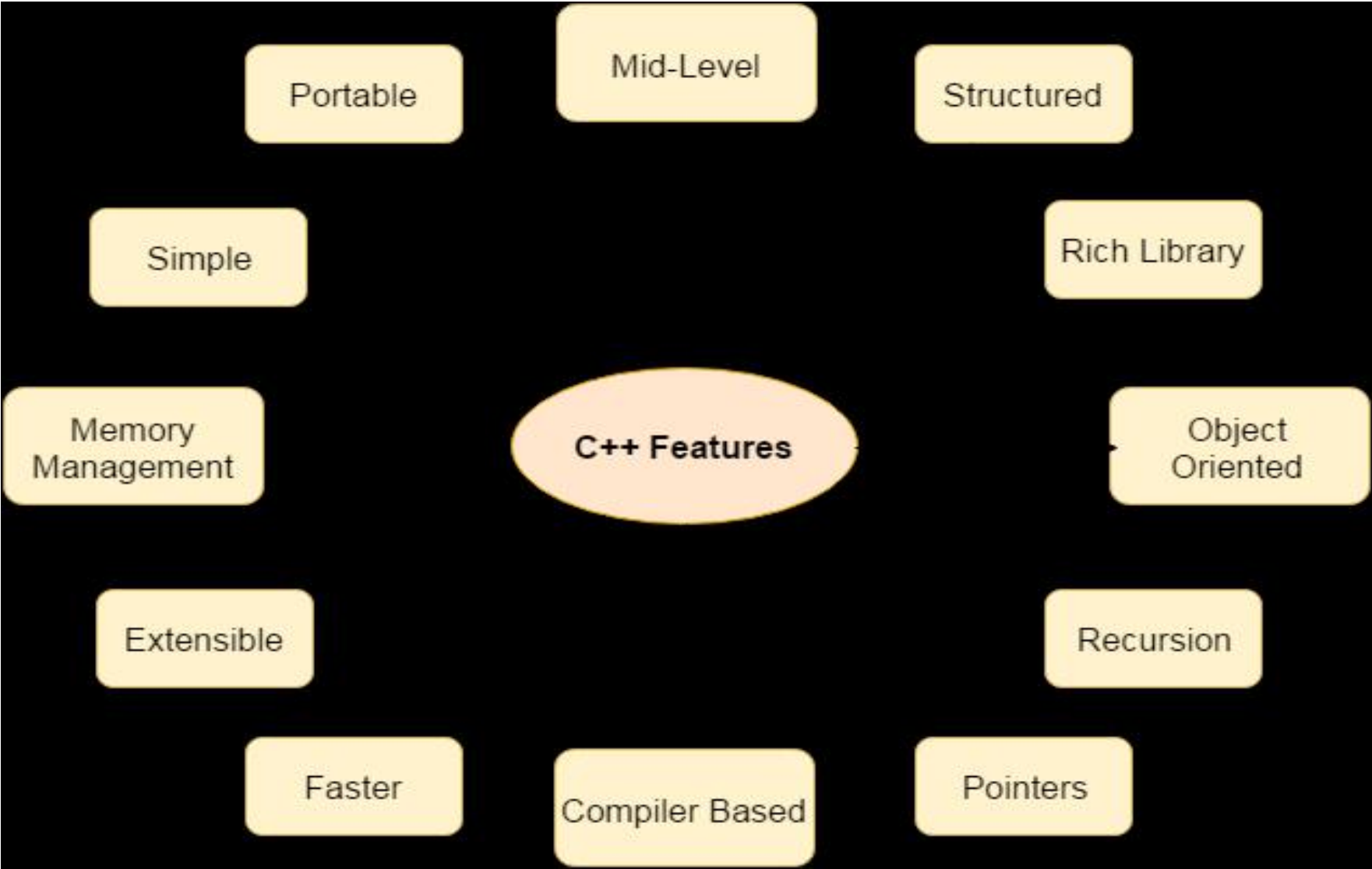
## **Message Passing**

Message Passing is nothing but sending and receiving of information by the objects same as people exchange information. So this helps in building systems that simulate real life. Following are the basic steps in message passing.

- Creating classes that define objects and its behaviour.
- Creating objects from class definitions
- Establishing communication among objects

In OOPs, Message Passing involves specifying the name of objects, the name of the function, and the information to be sent.

# C++ Features



# C++ Features

C++ is object oriented programming language. It provides a lot of features that are given below.

## C++ Features

- Simple
- Machine Independent or Portable
- Mid-level programming language
- Structured programming language
- Rich Library
- Memory Management
- Fast Speed
- Pointers
- Recursion
- Extensible
- Object Oriented
- Compiler based



## **1) Simple**

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

## **2) Machine Independent or Portable**

Unlike assembly language, c++ programs can be executed in many machines with little bit or no change. But it is not platform-independent.

## **3) Mid-level programming language**

C++ is also used to do low level programming. It is used to develop system applications such as kernel, driver etc. It also supports the feature of high level language. That is why it is known as mid-level language.

#### **4) Structured programming language**

C++ is a structured programming language in the sense that we can break the program into parts using functions. So, it is easy to understand and modify.

#### **5) Rich Library**

C++ provides a lot of inbuilt functions that makes the development fast.

#### **6) Memory Management**

It supports the feature of dynamic memory allocation. In C++ language, we can free the allocated memory at any time by calling the `free()` function.

#### **7) Speed**

The compilation and execution time of C++ language is fast.

#### **8) Pointer**

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

## **9) Recursion**

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

## **10) Extensible**

C++ language is extensible because it can easily adopt new features.

## **11) Object Oriented**

C++ is object oriented programming language. OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.

## **12) Compiler based**

C++ is a compiler based programming language, it means without compilation no C++ program can be executed. First we need to compile our program using compiler and then we can execute our program.