

Programming with C++

BCAN – E302A

Module – I

Concepts of OOP

Introduction to C++ -

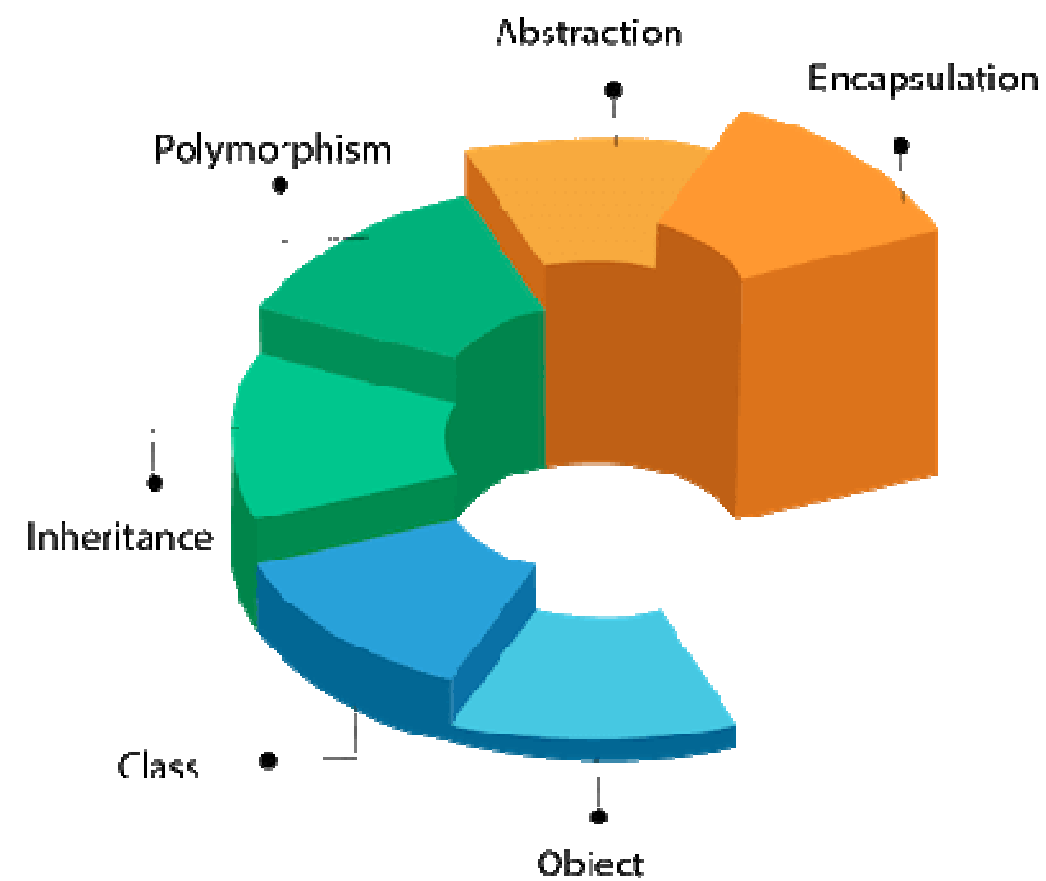
- C++, as we all know is an extension to C language and was developed by **Bjarne Stroustrup** at bell labs.
- C++ is an intermediate level language, as it comprises a confirmation of both high level and low level language features.
- C++ is a statically typed, free form, multi-paradigm, compiled general-purpose language.
- C++ is an Object Oriented Programming language but is not purely Object Oriented, due to its features like Friend and Virtual, violate some of the very important OOPS features, rendering this language unworthy of being called completely Object Oriented.

Benefits of C++ over C Language -

- Following features of C++ makes it a stronger language than C –
 1. There is Stronger Type Checking in C++.
 2. All the OOPS features in C++ like Abstraction, Encapsulation, Inheritance etc makes it more worthy and useful for programmers.
 3. C++ supports and allows user defined operators (i.e Operator Overloading) and function overloading is also supported in it.
 4. Exception Handling is there in C++.
 5. The Concept of Virtual functions and also Constructors and Destructors for Objects.
 6. Inline Functions in C++ instead of Macros in C language. Inline functions make complete function body act like Macro, safely.
 7. Variables can be declared anywhere in the program in C++, but must be declared before they are used.

Object Oriented Programming -

- Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, Encapsulation etc.



Object Oriented Programming -

- **Class** - A class is a blueprint for any functional entity which defines its properties and its functions.
- **Objects** - An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.
- **Polymorphism** - *When one task is performed by different ways i.e. known as polymorphism. In C++, Function overloading and Function overriding to achieve polymorphism.*
- **Inheritance** - When one object acquires all the properties and behaviors of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.
- **Abstraction** - Hiding internal details and showing functionality is known as abstraction. 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.