

Name of Teacher: Miss Radhika M. Patil

Class: B.Sc. Computer Science (Entire)- II **Semester : 3**

Course Title: Object Oriented Programming using C++

Object-oriented programming

- As the name suggests uses objects in programming.
- Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming.
- The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.
- The core of the pure object-oriented programming is to create an object, in code, that has certain properties and methods.
- While designing C++ modules, we try to see whole world in the form of objects.
- For example a car is an object which has certain properties such as color, number of doors, and the like. It also has certain methods such as accelerate, brake, and so on.

OOPs concepts/Features

1. Class
2. Objects
3. Encapsulation
4. Abstraction
5. Inheritance
6. Polymorphism

1. Class

- The building block of C++ that leads to Object-Oriented programming is a Class.
- It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.
- A class is like a blueprint for an object.
- Collection of objects is called class. It is a **logical** entity.
- Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions define the properties and behavior of the objects in a Class.
- In the example of class Car, the data member will be speed limit, mileage etc and member functions can apply brakes, increase speed etc.
- We can say that a **Class in C++** is a blue-print representing a group of objects which shares some common properties and behavior.

2. Object

- This is the basic unit of object oriented programming.
- In C++, Object is a real world entity that has **state and behavior**. Here, state means data and behavior means functionality.
- It can be physical and logical.
- 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.
- **All the members** of the class can be **accessed through object**.
- When a program is executed the objects interact by sending messages to one another.
- Each object contains data and code (functions) to manipulate the data.
- Objects can interact without having to know details of each other's data or code.

3. Data Encapsulation

- Encapsulation is defined as wrapping up of data and information under a single unit.
- In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them.
- Encapsulation is a process of combining data and function into a single unit like capsule.
- This is to avoid the access of private data members from outside the class.
- To achieve encapsulation, we make all data members of class private and create public functions, using them we can get the values from these data members or set the value to these data members.

4. Data Abstraction:

- Data abstraction is one of the most essential and important features of object-oriented programming in C++.
- Abstraction means displaying only essential information and hiding the details.
- Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.
- For example, a database system hides certain details of how data is stored and created and maintained. Similar way, C++ classes provides different methods to the outside world without giving internal detail about those methods and data.

5. Inheritance:

- The capability of a class to derive properties and characteristics from another class is called Inheritance.
- Inheritance is one of the most important features of Object-Oriented Programming.
- **Derived or Sub Class:** The class that inherits properties from another class is called Sub class or Derived Class.
- **Base or Super Class:** The class whose properties are inherited by sub class is called Base Class or Super class.
- **Reusability:** Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.
- **Example:** Dog, Cat, Cow can be Derived Class of Animal Base Class.

6. Polymorphism

- The word polymorphism means having many forms.
- In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.
- A real-life example of polymorphism, a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee.
- So the same person possesses different behavior in different situations. This is called polymorphism.
- Polymorphism is considered as one of the important features of Object Oriented Programming.

In C++ polymorphism is mainly divided into two types:

1. Compile time Polymorphism
2. Runtime Polymorphism

THANK YOU...