Introduction to OOP

Object oriented programming is a type of programming which uses objects and classes for its functioning. The object-oriented programming is based on real world entities like inheritance, polymorphism, data hiding, etc. It aims at binding together data and function work on these data sets into a single entity to restrict their usage.

The main intent of introducing the C++ programming language was to add object-oriented features to the C language. OOPs offer several benefits or advantages to the designer and user, and there are various areas where OOPs play an essential role.

Why Do You Need Object-Oriented Programming?

In object-oriented programming, it is easy to maintain the code with the help of classes and objects. Using inheritance, there is code reusability, i.e., you don't have to write the same code again and again, which increases the simplicity of the program. Concepts like encapsulation and abstraction provide data hiding as well.

Advantages of OOPs

There are various advantages of object-oriented programming.

OOPs provide reusability to the code and extend the use of existing classes.

In OOPs, it is easy to maintain code as there are classes and objects, which helps in making it easy to maintain rather than restructuring.

It also helps in data hiding, keeping the data and information safe from leaking or getting exposed.

Object-oriented programming is easy to implement.

Some basic concepts of object oriented programming are −

1. CLASS

2. OBJECTS

3. ENCAPSULATION

4. POLYMORPHISM

5. INHERITANCE

6. ABSTRACTION

1. Class − A class is a data-type that has its own members i.e. data members and member functions. It is the blueprint for an object in object oriented programming language. It is the basic building block of object oriented programming in C++. The members of a class are accessed in programming language by creating an instance of the class.

Some **important properties** of class are −

Class is a user-defined data-type.

A class contains members like data members and member functions.

Data members are variables of the class.

Member functions are the methods that are used to manipulate data members.

Data members define the properties of the class whereas the member functions define the behaviour of the class.

A class can have multiple objects which have properties and behaviour that in common for all of them.

Syntax

```
class class_name {
  data_type data_name;
```

```
  return_type method_name(parameters);
}
```

2. Object − An object is an instance of a class. It is an entity with characteristics and behaviour that are used in the object oriented programming. An object is the entity that is created to allocate memory. A class when defined does not have memory chunk itself which will be allocated as soon as objects are created.

Syntax

```
class_name object_name;
```

3. Encapsulation- In object oriented programming, encapsulation is the concept of wrapping together of data and information in a single unit. A formal definition of encapsulation would be: encapsulation is binding together the data and related function that can manipulate the data.

Due to the concept of encapsulation in object oriented programming another very important concept is possible, it is data abstraction or Data Hiding. it is possible as encapsulating hides the data at show only the information that is required to be displayed.

4. Polymorphism- The name defines polymorphism is multiple forms. which means polymorphism is the ability of object oriented programming to do some work using multiple forms. The behavior of the method is dependent on the type or the situation in which the method is called.

In C++ programming language, polymorphism is achieved using two ways. They are operator overloading and function overloading.

Operator overloading- In operator overloading and operator can have multiple behavior in different instances of usage.

Function overloading - Functions with the same name that can do multiple types based on some condition.

5. **Inheritance** - It is the capability of a class to inherit or derive properties or characteristics other class. Inheritance is very important and object oriented program as it allows reusability i.e. using a method defined in another class by using inheritance. The class that derives properties from other class is known as child class or subclass and the class from which the properties are inherited is base class or parent class.

C++ programming language supports the following types of inheritance -

single inheritance

multiple inheritance

multi level inheritance

Hierarchical inheritance

hybrid inheritance

6. **Abstraction**- Data abstraction or Data Hiding is the concept of hiding data and showing only relevant data to the final user. It is also an important part object oriented programing.

In C++ programming language write two ways using which we can accomplish data abstraction −

using class

using header file