

## OOPS-Features

The features of OOP languages are--

- 1)Object
- 2)Class
- 3)Data Abstraction
- 4)Encapsulation
- 5)Inheritance
- 6)Polymorphism
- 7)Dynamic Binding

1)Object - An object is an instance of a class. Objects are the basic run time entities in oops. They may represent a person , a place, a table of data or any item that program has to handle. Objects take up space in the memory and have an associated address. When a program is executed, objects interact by sending messages. Objects can interact without having to know details of each other's data or code.

2)Class - A class is a template definition of the methods and variables in a particular kind of object. Objects are variables of the type class. A class is a collection of objects of similar type. Classes are user-defined data types and behave like built-in types of a programming language.

3)Data Abstraction - Abstraction means representing the essential features without including the background details. Data abstraction is a programming technique that relies on the separation of interface and implementation.

4)Encapsulation - The process of wrapping up of data into a single unit (class) is known as encapsulation. The data is not accessible to the outside world and only those functions which are wrapped in the class can access it. These functions provide the interface between the object's data and the program. This insulation of data from direct access by the program is called data hiding or information hiding.

5)Inheritance - Inheritance is the process by which objects of one class acquire the properties of objects of another class. The concept of

inheritance provides the idea of reusability, which means that we can add additional features to an existing class without modifying it. The new class will have combined features of both the classes. One advantage of inheritance is that it allows the programmer to reuse a class that is almost but not exactly what he wants.

6) Polymorphism - More than one form. An operation may exhibit different behaviours in different instances.

The behaviour depends upon the types of data used in the operation.

A single function name can be used to handle different number and types of arguments. Using a single function name to perform different types of tasks is --Function Overloading.

The process of making an operator to exhibit different behaviours in different instances --operator overloading.

7) Dynamic Binding - Also known as late binding. In this, the code associated with a given procedure call is not known until the time of the call at run-time.