

# FUNDAMENTALS OF OOP

**OOP:** It is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

**Class:** A class is a blueprint for creating objects, providing initial values for variables and implementations of methods.

**Method:** e.g. Saloon

**Object:** OOP allows us to decompose a problem into a number of entities called Objects.  
Object = Data + Methods

## **Features:**

Data Abstraction

Data Encapsulation

Inheritance

Polymorphism

Dynamic Binding

Message Communication

## Abstraction:

It refers to the act of representing essential feature without including the background details. It tells what it is not how it is.

## Encapsulation:

The wrapping up of data and methods into a single unit (called class) is known as encapsulation. Data encapsulation is the most striking feature of a class. The data is not accessible to the outside world and only those methods, which are wrapped in the class, can access it.

## Inheritance:

Inheritance is the process by which objects of one class acquire the properties of objects of another class.

In OOP, the concept of inheritance provides the idea of **reusability**. It means we can add additional features to an existing class without modifying it by deriving a new class from the existing one.

## Polymorphism:

It means the ability to take more than one form.

## Dynamic Binding:

It means that the code associated with a given procedure call is not known until the time of the call at runtime.

A procedure call associated with a polymorphic reference depends on the dynamic type of that reference.

## Benefits of OOP:

## Application of OOP: