

# Object oriented programming

Object oriented programming is a programming pattern and model which is based on the concept of object and class. Inheritance, polymorphism, abstraction, encapsulation are the basic characteristics or we can say structure of object oriented programming.

## Objects

Objects are the real world entity. Object represents state and behaviors of real world things. We can treat anything in the real world as object if it has state and characteristics. And this approach of programming is called object oriented programming.

## Class

A class is a blue print of an object. A class gives us an idea about our object. A class contains the information about our object like how our object will be look like, what kind of characteristic it will have. In more specific a class contains data type, state, methods of an object

Variables in a class are the property of an object.

Methods are the behaviors of an object.

Ex: -- suppose a class call dog. So dogs has some property and behaviors properties are (name, age etc.) which can represent by using variables and behaviors are (bark, run etc.) which can represent by using methods. But till now the class will not represent a real dog when we create object of this class then it will represent a real world living dog. And each dog is called instance of the dog class.

## Abstraction

Abstraction: -- suppose we discuss about a mug with a computer. Now between our conversations every time we say the word “mug” we have to specify its property like volume, color of the mug. Which is funny and bad? So what we can do we can treat mug as an object and hide its unnecessary and internal details and every time when we need to use the word mug we just refer the object. In this way we can make computer remember.

We say forest it will represent entire eco-system (including tree, animals, birds everything in the forest) this is call abstraction. Hiding internal detail and unnecessary detail of an object show only essential information.

To derive a car you no need to know the working process of the engine of the car you don't need a mechanical degree.

# Polymorphism

Polymorphism: -- in biology organism or species can have different state and forms which is called polymorphism. For example a woman at the same time can have different role (a mother, a wife, a sister, a teacher, a soldier, an employee) as per situation she can play any of this role.

Polymorphism is the ability of an object to take on many forms. Sometimes a parent class reference can be used to reflect a child class object. Any Java object which passes IS-A relation can be treated as a polymorphic object.

There are two types of polymorphic behaviors

1. Compile time polymorphism or method overloading (method same name different parameters)
2. Run time polymorphism or dynamic method dispatch or method overriding. (parent VS child class)

# Inheritance

Inheritance: -- In OOPS, classes can be derived from other classes is called inheritance. A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called superclass (also a base class or a parent class). It increases code reusability.

**Inheritance in Java** is a mechanism in which one object acquires all the properties and behaviours of a parent object. We have some common property among the entire living thing or we can categorize them on the basis of common character so that child object can inherit property from parent object.

# Encapsulation

Encapsulation:--encapsulation is a process of wrapping the code and data together into a single unit like a capsule. We can fully create an encapsulated class by declaring data members as private and we can use getter and setter.

Encapsulation may also refer to a mechanism of restricting the direct access to some components of an object, such that users cannot access state values for all of the variables of a particular object. Encapsulation can be used to hide both data members and data functions or methods associated with an instantiated class or object.

Encapsulation describes the ability of an object to hide its data and methods from the rest of the world and is one of the fundamental principles of object-oriented programming. In Java, a class encapsulates the fields, which hold the state of an object, and the methods, which define the actions of the object. Encapsulation enables you to write reusable programs. It also enables you to restrict access only to those features of an object that are declared public.

