

# Introduction to OOP

The idea behind OOP is to encapsulate private data & methods into objects, thereby making each object semi-autonomous.

E.g., Imagine you have placed several pipes, pumps, compressor & all kinds of switches in your kitchen to keep food cool. Every time the temperature of the food gets too high, we turn on the compressor, open the valves and start the pumps manually. Now suppose we wrap all these functionalities into an object let's say a refrigerator in which all these operations are handled automatically inside the object with proper co-ordination between the various parts.

This is the concept behind encapsulation or in other words encapsulation hides the complexity of the class by providing an object of the class. The object handles all the work internally and can be easily conceptualized similar to the preceding example of refrigerator.

## # Explanation of OOP concepts -:

The OOP provides basic concepts which help in programming. JAVA being an OOP language implements and applies the concept of OOP.

The basic concepts of OOP are as follows,

**1. Objects-** These are the basic run time entities in OOP. An object can represent a person, a place, a bank account etc. When a program is executed, the objects interact with each other by sending messages.

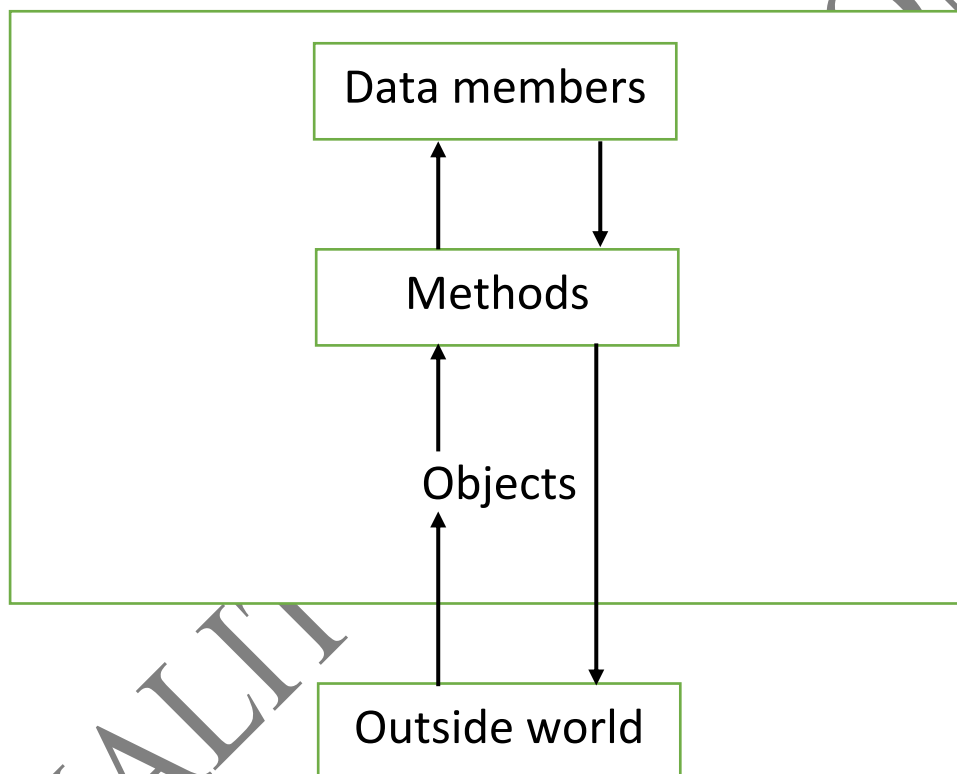
e.g. If employee & salary are two objects in a company program, the company object may send a message to the salary object requesting a salary. Each object contains data in methods to manipulate the data.

OOP allows de-composing a problem into a number of entities known as objects & then the methods are invoked using the objects. Therefore, an object is a combination of data and methods.

**2. Classes** - In OOP classes provide a sort of template for objects. We use a class to create an object & then we can call the objects, methods from our code.

In other words, objects are variables of type class. After defining a class any number of objects belonging to class can be declared. A class is a collection of objects of similar types.

### 3. Data hiding & Encapsulation



The concept of wrapping data and functions into a single unit is known as encapsulation. It is a mechanism that binds class members (variables & methods) together & prevents them from being accessed by other classes. The term data hiding implies that, data is not accessible by other classes & only the methods which are within the class can access it. The methods that serve as a mediator between the objects data & the program are known as member methods.

**4. Inheritance** - It is one of the OOP concepts by which a class acquires the properties of another class. A class i.e. inherited from another class is known as a subclass and the class from which the subclass is inherited is known as the superclass or the base class. Without inheritance, we need to define all the characteristics of each object explicitly.

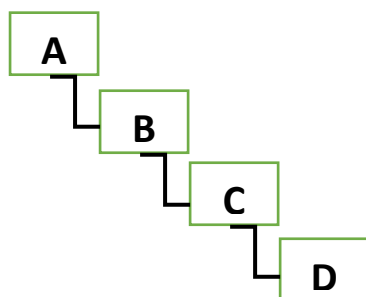
base class or super class,

class A

```
{  
    
}
```

class B

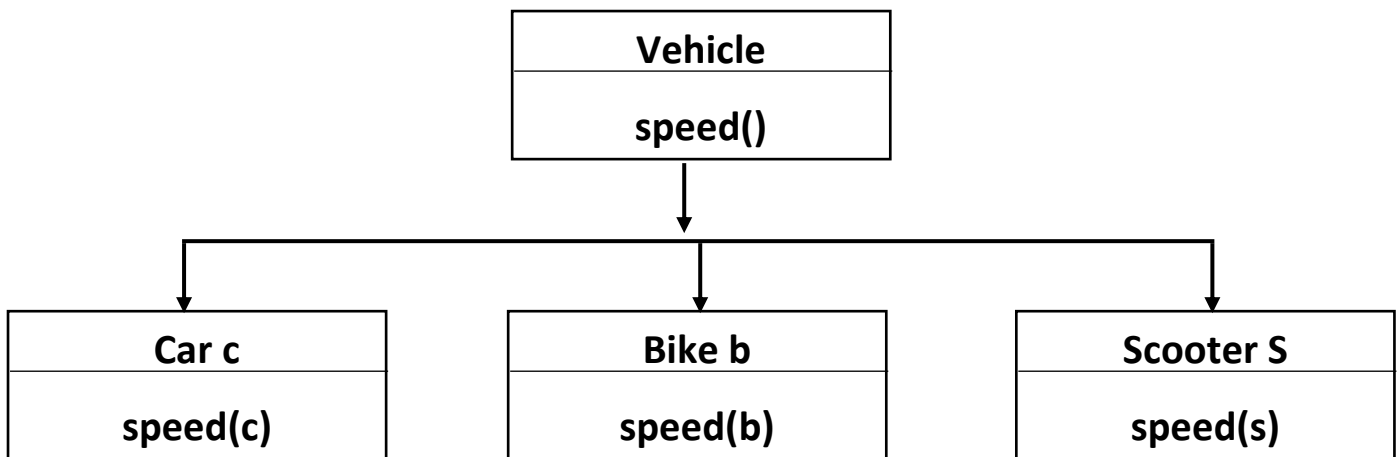
```
{  
    
}
```



It supports hierarchical classification. In OOP the inheritance concept facilitates re-usability as we can append additional Features in an existing class without modifying it by deriving a new class from the existing one. The sub-class derived from the super class contains the combined features of both the class.

**5. Polymorphism** - This term comes from two Greek words that is 'poly' means many and 'morphs' means forms. Therefore, polymorphism means many forms. It allows us to perform various operations by using the same methods. The behavior of an operation depends upon the type of data used in operation. e.g. let us consider the operation of addition, which can be addition of two no's or joining of two strings.

In first case, the operands are of integer type and in the other case the arguments are strings. The concept of using single method name to perform different types of tasks is known as method overloading.



In above diagram, the speed method of the vehicle class is used in different classes that is in the car, bike & scooter. The objects of the respective classes are passed as an argument to the speed method. In short one method or the function to perform different operations which depends upon the types of data used in operations.

**6. Dynamic binding** - The process of linking a method to call the executable code in response to the call is known as binding. This process executes at run time so it is called dynamic binding.

This means that when a method is called, the method call is bound to the method body during the runtime of the program. The java compiler is not aware of the method to be invoked during compilation. The JVM (Java Virtual Machine) invokes the relevant method during runtime. Hence, this is called as dynamic binding.

e.g.

class book

{

    Int bookcode;

    char name[10];

    int cost;

    public void getdata ();

    public void putdata ();

```
}  
book b1, b2;  
b1.getdata();  
b1.putdata();  
b2.getdata();  
b2.putdata();
```

```
getdata()
```

```
{
```

```
.....
```

```
}
```

```
putdata()
```

```
{
```

```
.....
```

```
}
```

EQUALITY EDUCATION