JAVA:

OOP’S IN JAVA:

**I) Inheritance:**

Inheritance is the concept of taking the properties of the parent class to the child class so that the attributes and method of parent class can be reused in a better way we can access, override the members of the parent class as we want

*Note:*

* *Object class is always inherited to the classes by default*
* *We cannot inherit a final class*
* *Only the members which are defined by public, static, protected can be accessed by the child class and the one which are declared by private cannot be accessed by the child class*

We can define the child class either by the name of itself or by the name of the parent class.

**II) Polymorphism:**

Polymorphism can be defined as “many forms” it occurs when many classes are related to each other by inheritance.

It can be uses the methods and attributed of another class (super class) in different ways.

class vehicle {

    String brand = null;

    void sound()

    {

        System.out.println("This is the sound in the vehicle class");

    }

}

class car extends vehicle{

    String brand = "BMW";

    int sound()// ERROR (EXPLANATION BELOW)

    {

    }

}

You are trying to override the sound() method from the vehicle class in the car class. In Java, method overriding requires that the method in the subclass have the same return type as the method in the superclass. Since the sound() method in vehicle has a void return type, the overridden method in car must also have a void return type. Changing it to int causes a compilation error.

Discontinued should be added with the classes

Inner classes:

We can declare a class inside a class (inner class) it makes code readable and easily maintainable

We make inner classes when many of the classes belong together

To create the object of the inner class we should first create the object of the outer class and using that object we create a object of the inner class.

* We cannot access the inner class if it is declared by private keyword

class Outer

{

    int x=20;

    private class inner\_class{

        int y=10;

        void innerMethod()

        {

            System.out.println("This is innner class");

        }

    }

}

public class private\_innner\_class {

    public static void main(String[] args)

    {

        Outer out = new Outer();

        Outer.inner\_class in=out.new inner\_class();

        // System.out.println(out.x+" "+in.y);

    }

}

Output:

