**Tutorial – 3**

1. Create a class Vehicle with run() .Create class Bike that extends Vehicle class and by defining run () method demonstrate method overriding.

**Code:**

class Vehicle {

    public void run() {

        System.out.println("The vehicle is running.");

    }

}

class Bike extends Vehicle {

    @Override

    public void run() {

        System.out.println("The bike is running.");

    }

}

public class Main1 {

    public static void main(String[] args) {

        Vehicle vehicle = new Vehicle();

        vehicle.run();

        Bike bike = new Bike();

        bike.run()

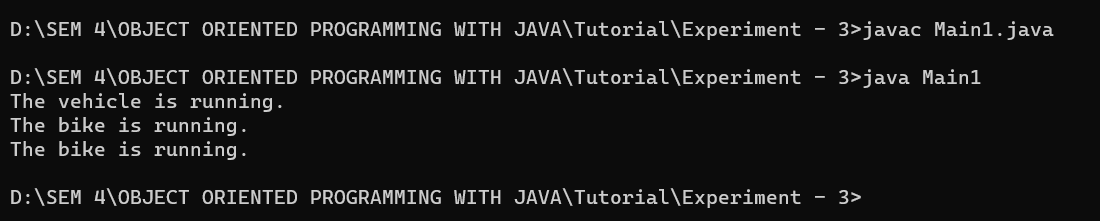
        Vehicle anotherBike = new Bike();

        anotherBike.run();

    }

}

**Output Screenshot:**

****

1. Creates classes Car, Maruti and Maruti800. We have done a setup – class Maruti extends Car and class Maruti800 extends Maruti. With the help of this Multilevel hierarchy setup our Maruti800 class is able to use the methods of both the classes (Car and Maruti). In Car have method name Speed() along with data member fuel. Also Maruti have gear() along with geartype data member and Maruti800 have getModel() along  with data member gear type. Display all details using Maruti800 class.

**Code:**

class Car {

    int fuel;

    void speed() {

        System.out.println("Car's speed method.");

    }

}

class Maruti extends Car {

    String gearType;

    void gear() {

        System.out.println("Maruti's gear method.");

    }

}

class Maruti800 extends Maruti {

    String model;

    void getModel() {

        System.out.println("Maruti800's getModel method.");

    }

}

public class Main2 {

    public static void main(String[] args) {

        Maruti800 car = new Maruti800();

        car.fuel = 10;

        car.gearType = "Manual";

        car.model = "Maruti 800";

        car.speed();

        car.gear();

        car.getModel();

        System.out.println("Fuel: " + car.fuel);

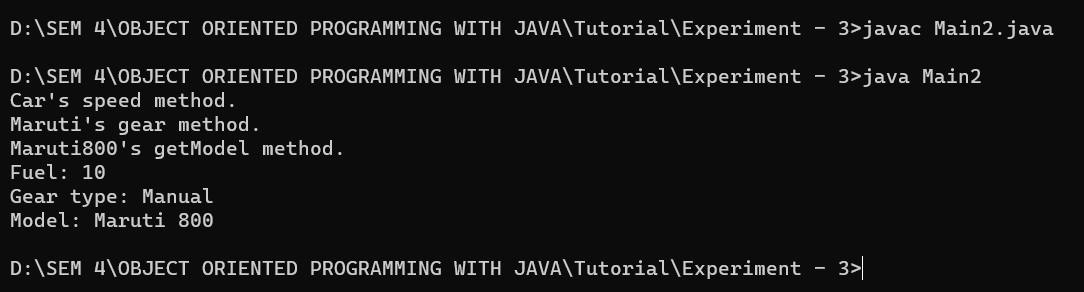
        System.out.println("Gear type: " + car.gearType);

        System.out.println("Model: " + car.model);

    }

}

**Output Screenshot:**

****

1. Create a class Bank with method name GetInterestRate()  as a super class. Implement hierarchy inheritance with child class ICICI ,SBI,BOB along with method GetInterestRate demonstrate method overriding.

**Code:**

class Bank {

    double getInterestRate() {

        return 0;

    }

}

class ICICI extends Bank {

    double getInterestRate() {

        return 5.0;

    }

}

class SBI extends Bank {

    double getInterestRate() {

        return 4.5;

    }

}

class BOB extends Bank {

    double getInterestRate() {

        return 4.0;

    }

}

public class Main3 {

    public static void main(String[] args) {

        Bank bank1 = new ICICI();

        Bank bank2 = new SBI();

        Bank bank3 = new BOB();

        System.out.println("ICICI Interest Rate: " + bank1.getInterestRate());

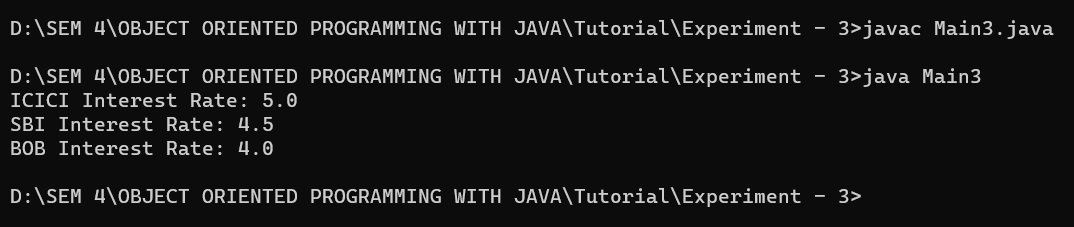
        System.out.println("SBI Interest Rate: " + bank2.getInterestRate());

        System.out.println("BOB Interest Rate: " + bank3.getInterestRate());

    }

}

**Output Screenshot:**

****

1. Create a class Student with static method getName() ,SetName(), ShowDetails()  and with some data member like name,Rollno,Branch,sem,college. Using getName() take input from user and using SetName() change details of mention data member. Finally display updated values of Student.

**Code:**

import java.util.Scanner;

class Student {

    private String name;

    private int rollNo;

    private String branch;

    private int semester;

    private String college;

    public static String getName() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter student name: ");

        return scanner.nextLine();

    }

    public static void setName(Student student, String name, int rollNo, String branch, int semester, String college) {

        student.name = name;

        student.rollNo = rollNo;

        student.branch = branch;

        student.semester = semester;

        student.college = college;

    }

    public void showDetails() {

        System.out.println("Name: " + name);

        System.out.println("Roll No: " + rollNo);

        System.out.println("Branch: " + branch);

        System.out.println("Semester: " + semester);

        System.out.println("College: " + college);

    }

}

class Main4 {

    public static void main(String[] args) {

        Student student = new Student();

        String name = Student.getName();

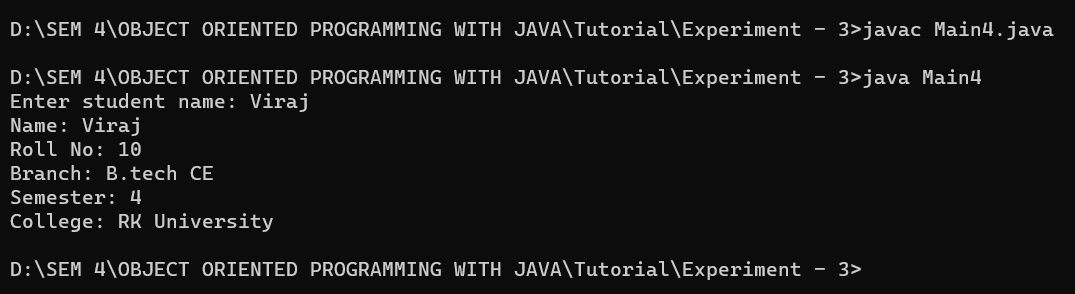
        Student.setName(student, name, 10, "B.tech CE", 4, "RK University");

        student.showDetails();

    }

}

**Output Screenshot:**

****

1. Design a class Circle with data member radius and one as final pie and find area of circle.

**Code:**

class Circle {

    private double radius;

    private final double PI = 3.14159;

    public Circle(double radius) {

*this*.radius = radius;

    }

    public double getArea() {

        return PI \* radius \* radius;

    }

}

public class Main5 {

    public static void main(String[] args) {

        Circle circle = new Circle(5);

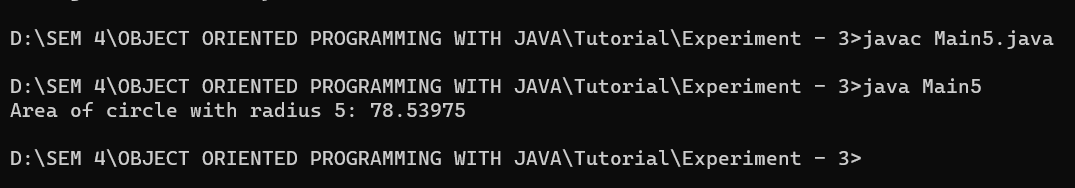
        double area = circle.getArea();

        System.out.println("Area of circle with radius 5: " + area);

    }

}

**Output Screenshot:**

****

1. Demonstrate usage of this keyword with suitable example.

**Code:**

class Person {

    private String name;

    private int age;

    public Person(String name, int age) {

*this*.name = name;

*this*.age = age;

    }

    public String getName() {

        return *this*.name;

    }

    public int getAge() {

        return *this*.age;

    }

    public void setName(String name) {

*this*.name = name;

    }

    public void setAge(int age) {

*this*.age = age;

    }

    public void printDetails() {

        System.out.println("Name: " + *this*.getName());

        System.out.println("Age: " + *this*.getAge());

    }

}

public class Main6 {

    public static void main(String[] args) {

        Person person = new Person("Viraj", 20);

        person.printDetails();

        person.setName("Uday");

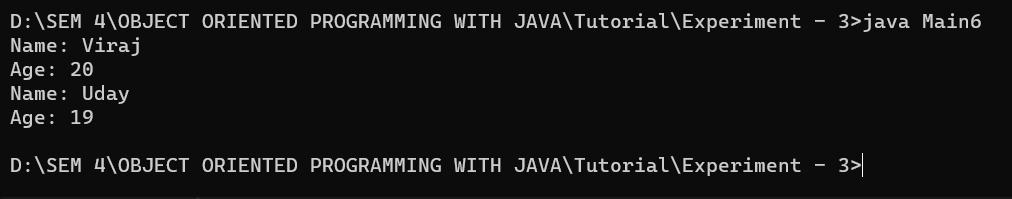
        person.setAge(19);

        person.printDetails();

    }

}

**Output Screenshot:**

****

1. Design class Shape in which draw() declare as final. Extends Shape class along with sub class Rectangle which have draw(). Try to override draw() and take a screen shot of output or error.

**Code:**

class Shape {

    public final void draw() {

        System.out.println("Drawing a shape");

    }

}

class Rectangle extends Shape {

    @Override

    public void draw() {

        System.out.println("Drawing a rectangle");

    }

}

public class Main7 {

    public static void main(String[] args) {

        Shape shape = new Rectangle();

        shape.draw();

    }

}

class Square extends Rectangle {

    @Override

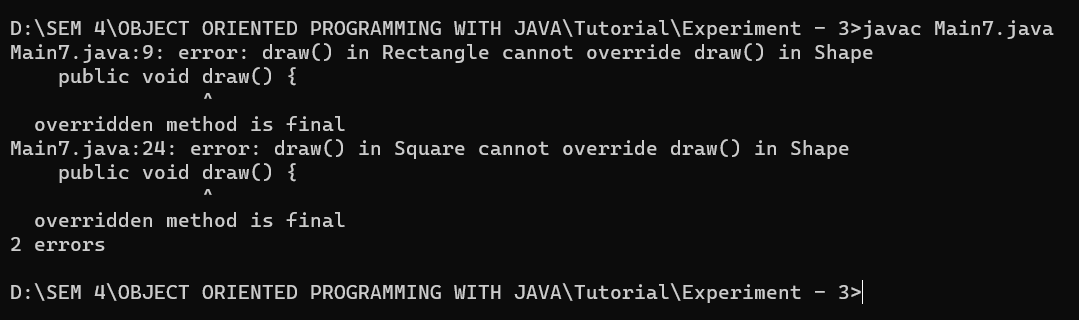
    public void draw() {

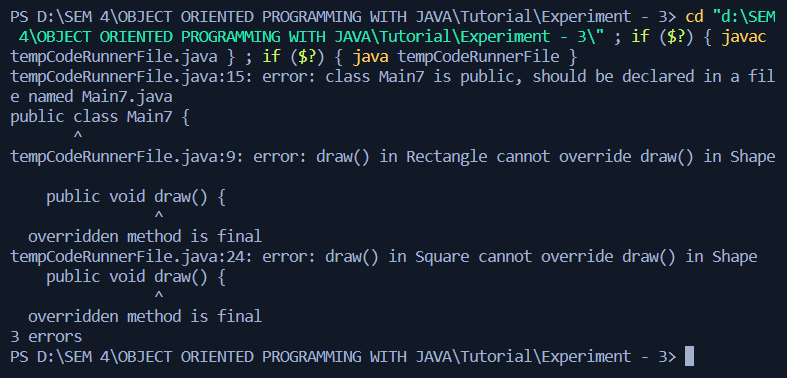
        System.out.println("Drawing a square");

    }

}

**Output Screenshot:**

****

****