1). Write a Java program to create a class called Vehicle with a method called drive().

• Vehicle should have attributes such as make (String), model (String), year (int) and maximumSpeed (int).

• Create a constructor in Vehicle with all fields as constructor parameters.

• Create a subclass called Car and override constructor. Call super()..

Write a function that overrides the drive() method to print (make + " " +

model + " Car is driving”.)

• Also create another subclass Bike extending the vehicle class..

Override the drive() method to print (make + " " + model + " Bike is

driving".)

• Instantiate both Bike and Car class. Print their attributes.

**CODE:**

package carBike;

class Vehicle {

String make;

String model;

int year;

int maximumSpeed;

// Constructor

public Vehicle(String make, String model, int year, int maximumSpeed) {

this.make = make;

this.model = model;

this.year = year;

this.maximumSpeed = maximumSpeed;

}

// Method to be overridden

public void drive() {

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

}

}

//Subclass Car

class Car extends Vehicle {

public Car(String make, String model, int year, int maximumSpeed) {

super(make, model, year, maximumSpeed);

}

@Override

public void drive() {

System.out.println(make + " " + model + " Car is driving.");

}

}

//Subclass Bike

class Bike extends Vehicle {

public Bike(String make, String model, int year, int maximumSpeed) {

super(make, model, year, maximumSpeed);

}

@Override

public void drive() {

System.out.println(make + " " + model + " Bike is driving.");

}

}

//Main class to test the implementation

public class InheritanceVehicle {

public static void main(String[] args) {

// Instantiate Car and Bike

Car car = new Car("Toyota", "Fortuner", 2022, 180);

Bike bike = new Bike("Honda", "Dio", 2017, 100);

// Print their attributes

System.out.println("Car Details:");

System.out.println("Make: " + car.make + ", Model: " + car.model + ", Year: " + car.year + ", Maximum Speed: " + car.maximumSpeed);

car.drive();

System.out.println("\nBike Details:");

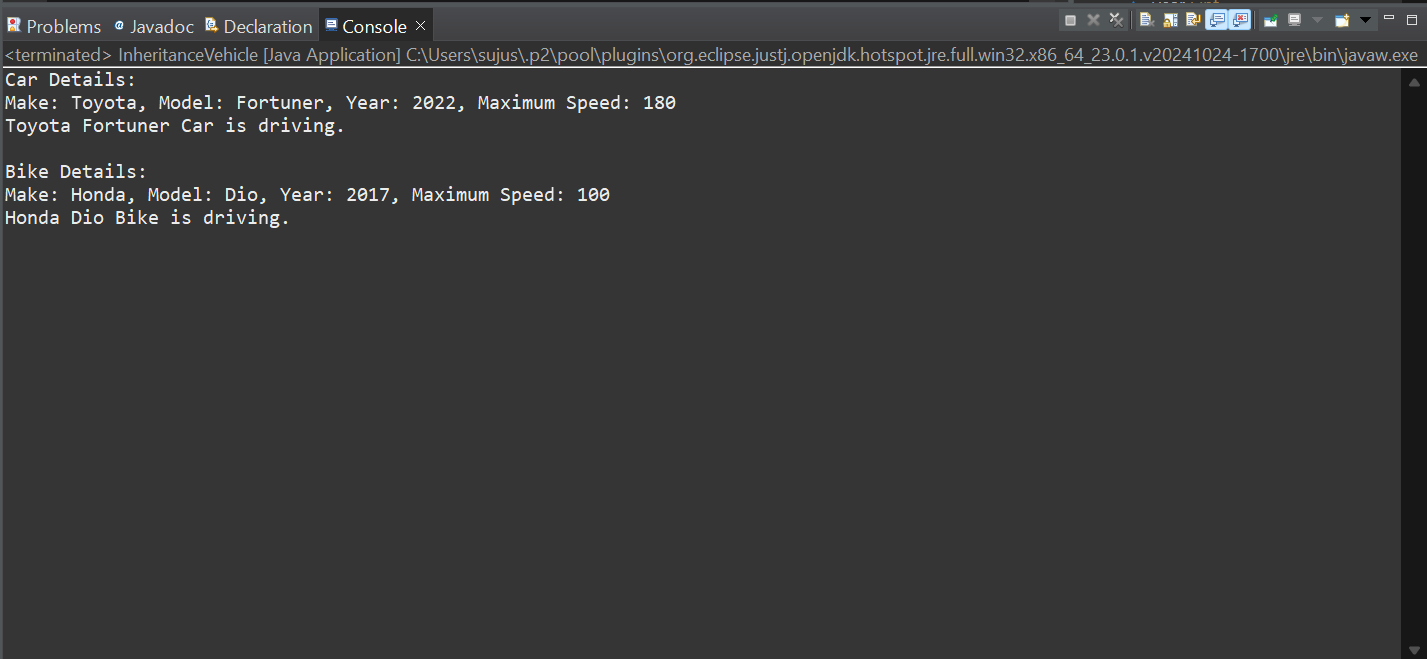
System.out.println("Make: " + bike.make + ", Model: " + bike.model + ", Year: " + bike.year + ", Maximum Speed: " + bike.maximumSpeed);

bike.drive();

}

}

**OUTPUT**

****

2).Write a Java program to create a class called Shape with a method called getArea().

• Create a subclass called Circle and create a constructor that takes the value of radius(int) as input parameter.

• Override the getArea() method.

• Create a class called square that takes an attribute length. Create aconstructor that takes length as input.

• Override the getArea() method.

• Create a subclass of Shape called Rectangle that takes width and height as input to the constructor.

• Override the getArea() method to calculate the area of a rectangle. Instantiate and call getArea() method.

**CODE:**

package shapes;

//Base class

class Shape {

// Method to be overridden

public double getArea() {

return 0; // Default implementation

}

}

//Subclass Circle

class Circle extends Shape {

int radius;

// Constructor

public Circle(int radius) {

this.radius = radius;

}

@Override

public double getArea() {

return Math.PI \* radius \* radius; // Area of a circle: πr²

}

}

//Subclass Square

class Square extends Shape {

int length;

// Constructor

public Square(int length) {

this.length = length;

}

@Override

public double getArea() {

return length \* length; // Area of a square: side²

}

}

//Subclass Rectangle

class Rectangle extends Shape {

int width;

int height;

// Constructor

public Rectangle(int width, int height) {

this.width = width;

this.height = height;

}

@Override

public double getArea() {

return width \* height; // Area of a rectangle: width × height

}

}

//Main class to test the program

public class ShapeArea {

public static void main(String[] args) {

// Create objects of Circle, Square, and Rectangle

Shape circle = new Circle(3); // Radius = 3

Shape square = new Square(7); // Length = 7

Shape rectangle = new Rectangle(4, 9); // Width = 4, Height = 9

// Call getArea() and print results

System.out.println("Circle Area: " + circle.getArea());

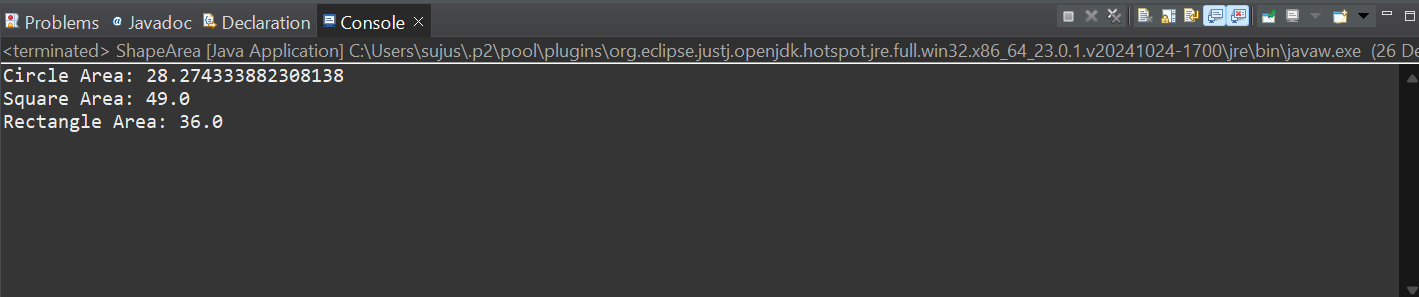
System.out.println("Square Area: " + square.getArea());

System.out.println("Rectangle Area: " + rectangle.getArea());

}

}

**OUTPUT**

****