Index

Course Title: Object Oriented Programming Lab

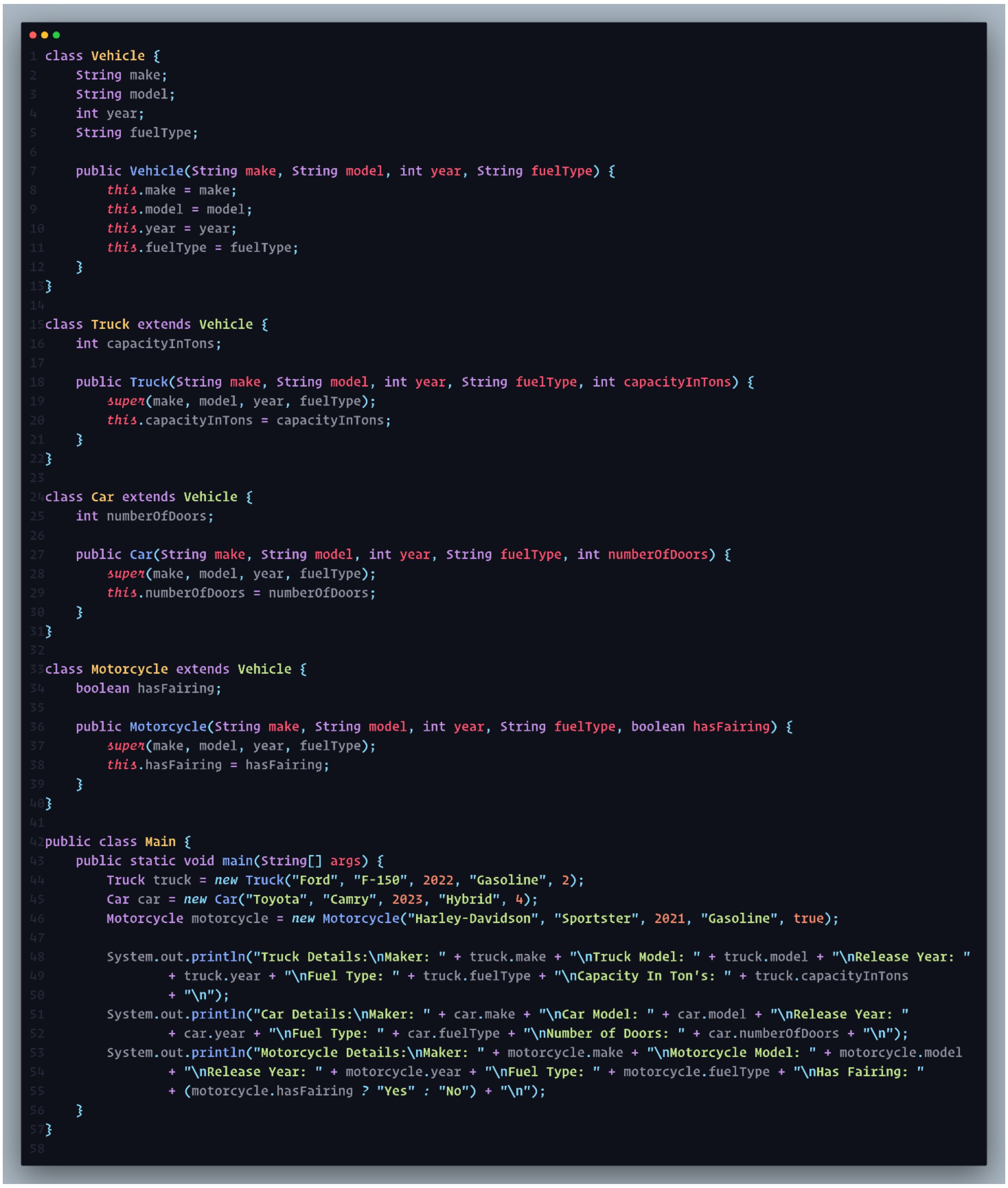
Course Code: CSE 212

Sl	Experiment Title	Marks			
		Contents (60%)	Organization(20%)	Writing skills (20%)	Total
1	Inheritance				
2	Method Overloading and Method Overriding				
3	Exception Handling				
4	Multithreading				
5	Abstraction & Interface				
6	Java Graphics				

ID:	**0222210005101118**	**Final Marks**	
Name:	**Mohammad Hafizur Rahman Sakib**		

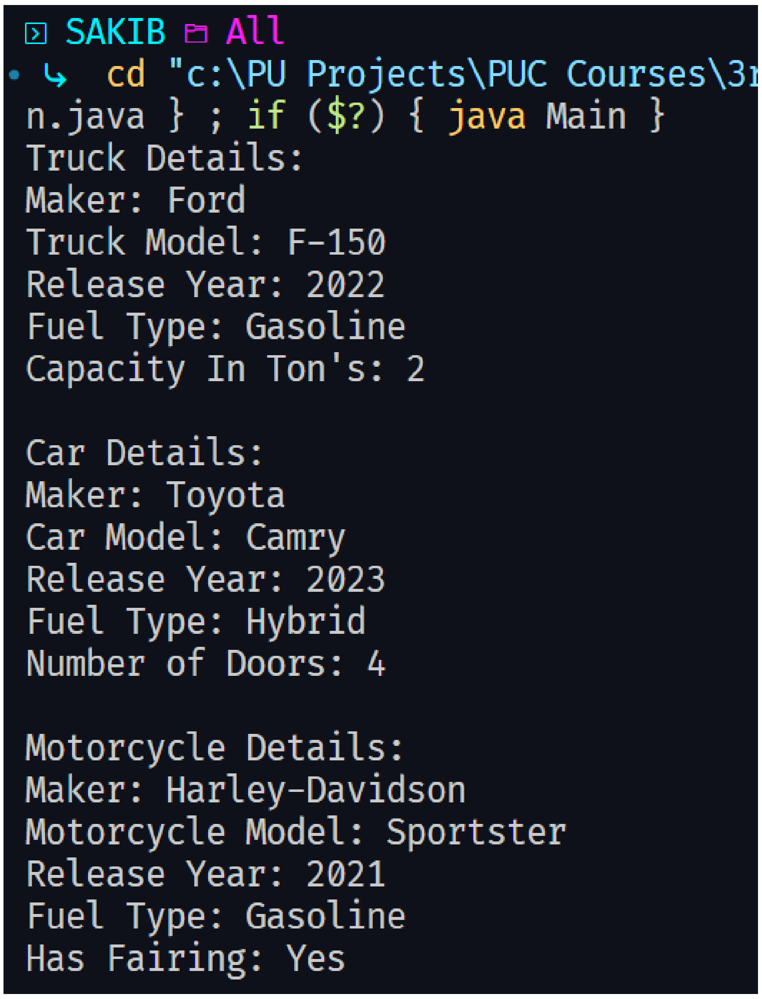
Problem Statement : Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle, with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type.

Source Code :



```java
class Vehicle {
    String make;
    String model;
    int year;
    String fuelType;

    public Vehicle(String make, String model, int year, String fuelType) {
        this.make = make;
        this.model = model;
        this.year = year;
        this.fuelType = fuelType;
    }
}

class Truck extends Vehicle {
    int capacityInTons;

    public Truck(String make, String model, int year, String fuelType, int capacityInTons) {
        super(make, model, year, fuelType);
        this.capacityInTons = capacityInTons;
    }
}

class Car extends Vehicle {
    int numberOfDoors;

    public Car(String make, String model, int year, String fuelType, int numberOfDoors) {
        super(make, model, year, fuelType);
        this.numberOfDoors = numberOfDoors;
    }
}

class Motorcycle extends Vehicle {
    boolean hasFairing;

    public Motorcycle(String make, String model, int year, String fuelType, boolean hasFairing) {
        super(make, model, year, fuelType);
        this.hasFairing = hasFairing;
    }
}

public class Main {
    public static void main(String[] args) {
        Truck truck = new Truck("Ford", "F-150", 2022, "Gasoline", 2);
        Car car = new Car("Toyota", "Camry", 2023, "Hybrid", 4);
        Motorcycle motorcycle = new Motorcycle("Harley-Davidson", "Sportster", 2021, "Gasoline", true);

        System.out.println("Truck Details:\nMaker: " + truck.make + "\nTruck Model: " + truck.model + "\nRelease Year: "
                + truck.year + "\nFuel Type: " + truck.fuelType + "\nCapacity In Ton's: " + truck.capacityInTons
                + "\n");
        System.out.println("Car Details:\nMaker: " + car.make + "\nCar Model: " + car.model + "\nRelease Year: "
                + car.year + "\nFuel Type: " + car.fuelType + "\nNumber of Doors: " + car.numberOfDoors + "\n");
        System.out.println("Motorcycle Details:\nMaker: " + motorcycle.make + "\nMotorcycle Model: " + motorcycle.model
                + "\nRelease Year: " + motorcycle.year + "\nFuel Type: " + motorcycle.fuelType + "\nHas Fairing: "
                + (motorcycle.hasFairing ? "Yes" : "No") + "\n");
    }
}
```

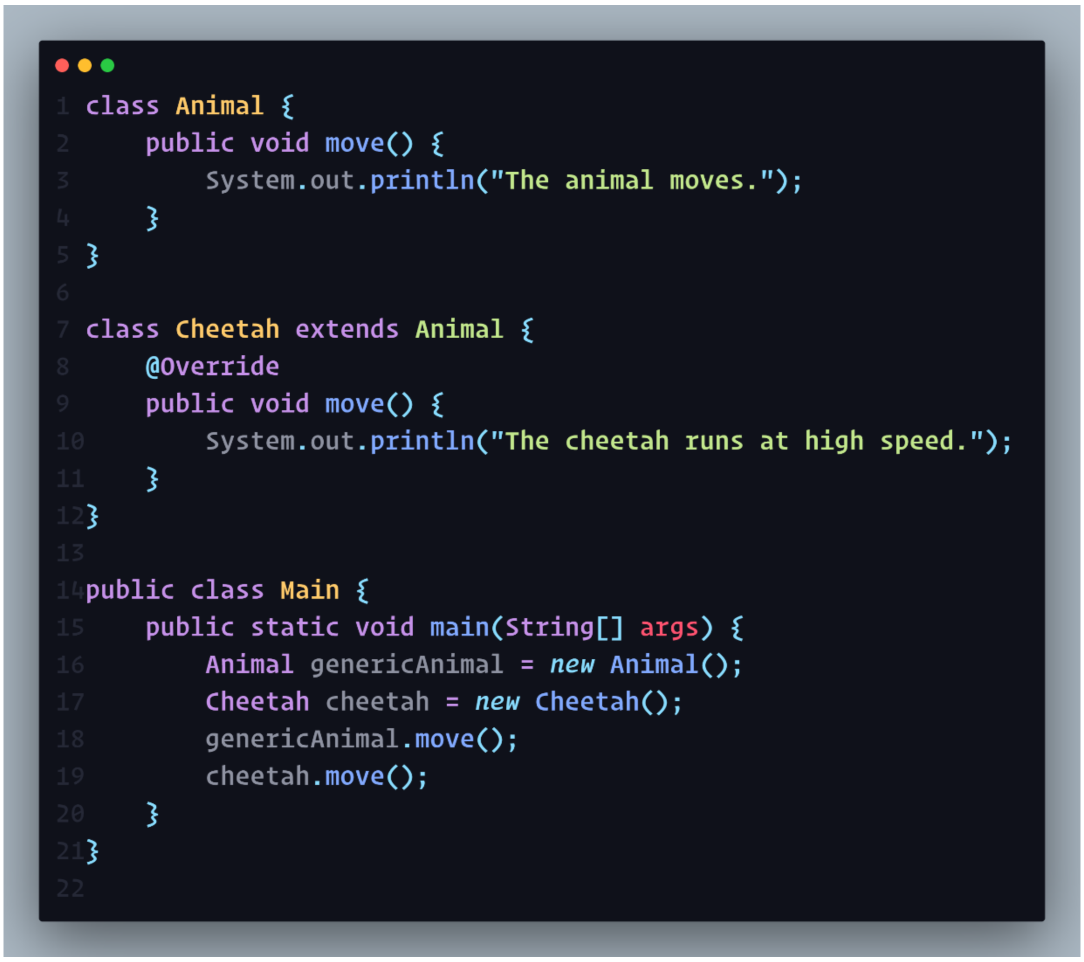
Output :



```
⊡ SAKIB 🗀 All
• ↳  cd "c:\PU Projects\PUC Courses\3r
n.java } ; if ($?) { java Main }
Truck Details:
Maker: Ford
Truck Model: F-150
Release Year: 2022
Fuel Type: Gasoline
Capacity In Ton's: 2

Car Details:
Maker: Toyota
Car Model: Camry
Release Year: 2023
Fuel Type: Hybrid
Number of Doors: 4

Motorcycle Details:
Maker: Harley-Davidson
Motorcycle Model: Sportster
Release Year: 2021
Fuel Type: Gasoline
Has Fairing: Yes
```

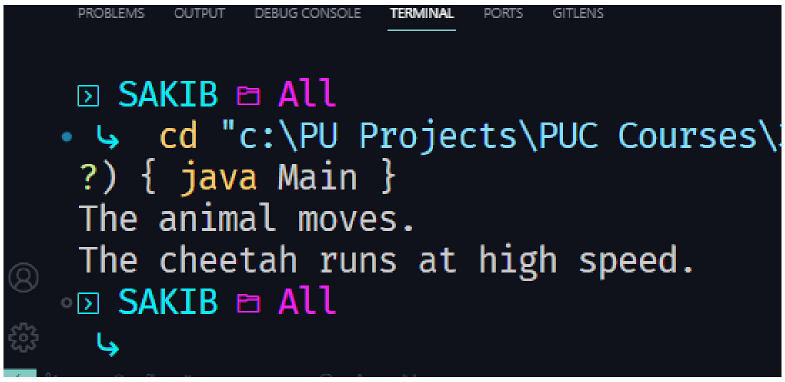
Problem Statement : Write a Java program to create a class called Animal with a method named move (). Create a subclass called Cheetah that overrides the move () method to run.

Source Code :



```java
class Animal {
    public void move() {
        System.out.println("The animal moves.");
    }
}

class Cheetah extends Animal {
    @Override
    public void move() {
        System.out.println("The cheetah runs at high speed.");
    }
}

public class Main {
    public static void main(String[] args) {
        Animal genericAnimal = new Animal();
        Cheetah cheetah = new Cheetah();
        genericAnimal.move();
        cheetah.move();
    }
}

```

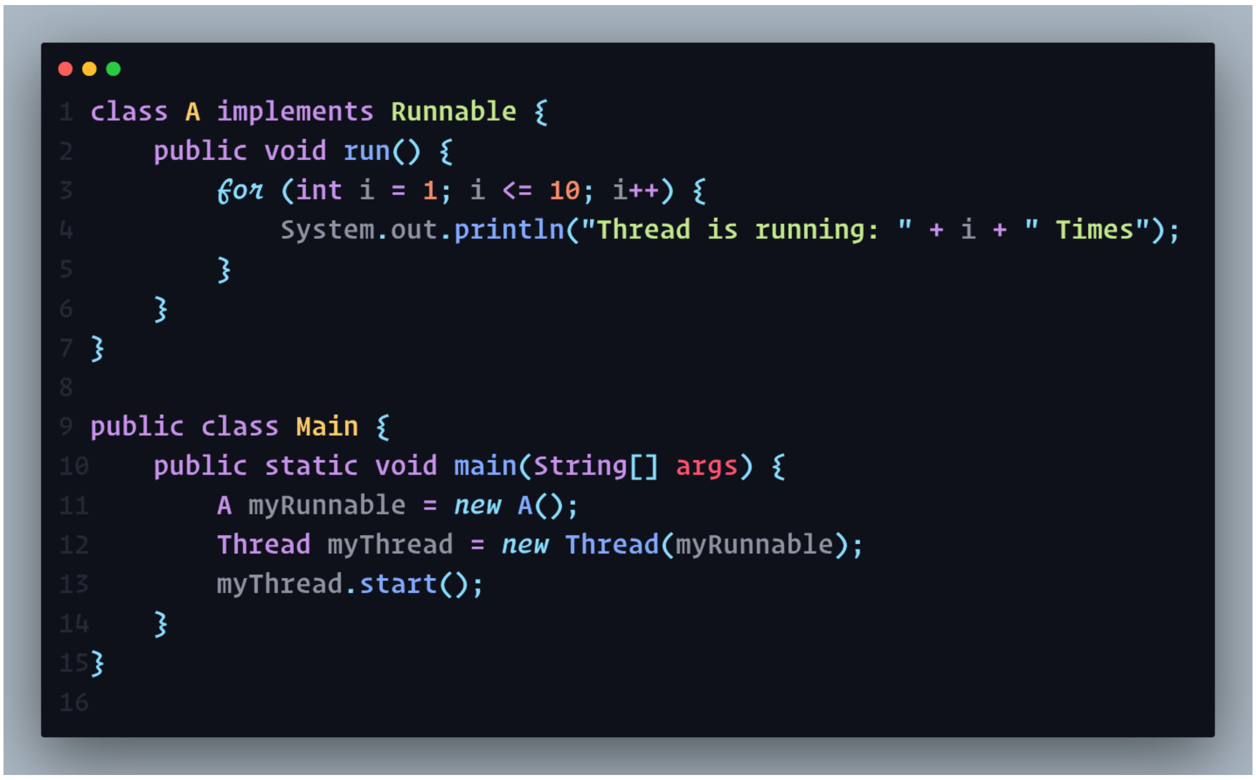
Output :



```
SAKIB All
↳  cd "c:\PU Projects\PUC Courses\
?) { java Main }
The animal moves.
The cheetah runs at high speed.
SAKIB All
↳
```

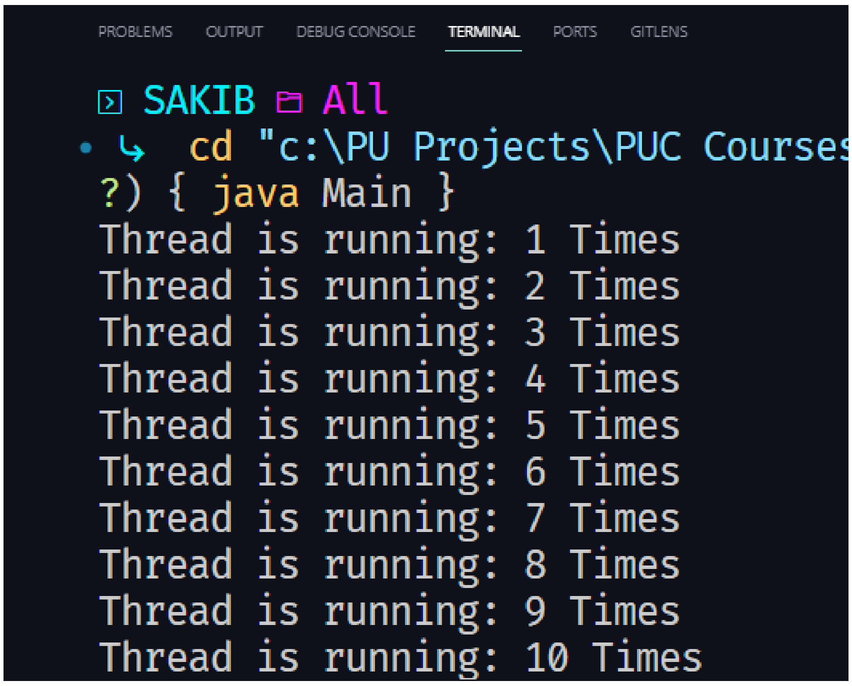
Problem Statement : Write a Java Program to Create a Basic Java Thread(by implementing Runnable Interface) that prints from number 1 to 10.

Source Code :



```java
class A implements Runnable {
    public void run() {
        for (int i = 1; i <= 10; i++) {
            System.out.println("Thread is running: " + i + " Times");
        }
    }
}

public class Main {
    public static void main(String[] args) {
        A myRunnable = new A();
        Thread myThread = new Thread(myRunnable);
        myThread.start();
    }
}
```

Output:



```
SAKIB  All
• ↳  cd "c:\PU Projects\PUC Courses
?) { java Main }
Thread is running: 1 Times
Thread is running: 2 Times
Thread is running: 3 Times
Thread is running: 4 Times
Thread is running: 5 Times
Thread is running: 6 Times
Thread is running: 7 Times
Thread is running: 8 Times
Thread is running: 9 Times
Thread is running: 10 Times
```

Problem Statement :

You are designing a simple program to calculate and display the areas of different shapes. Implement the necessary classes and interfaces to achieve this functionality using Java interfaces.

Create an interface named Shape with the following method:

double calculateArea(): This method calculates and returns the area of the shape.

Create two classes that implement the Shape interface:

Circle: This class should have an additional attribute: radius (double): The radius of the circle.

Rectangle: This class should have two additional attributes: width (double): The width of the rectangle.

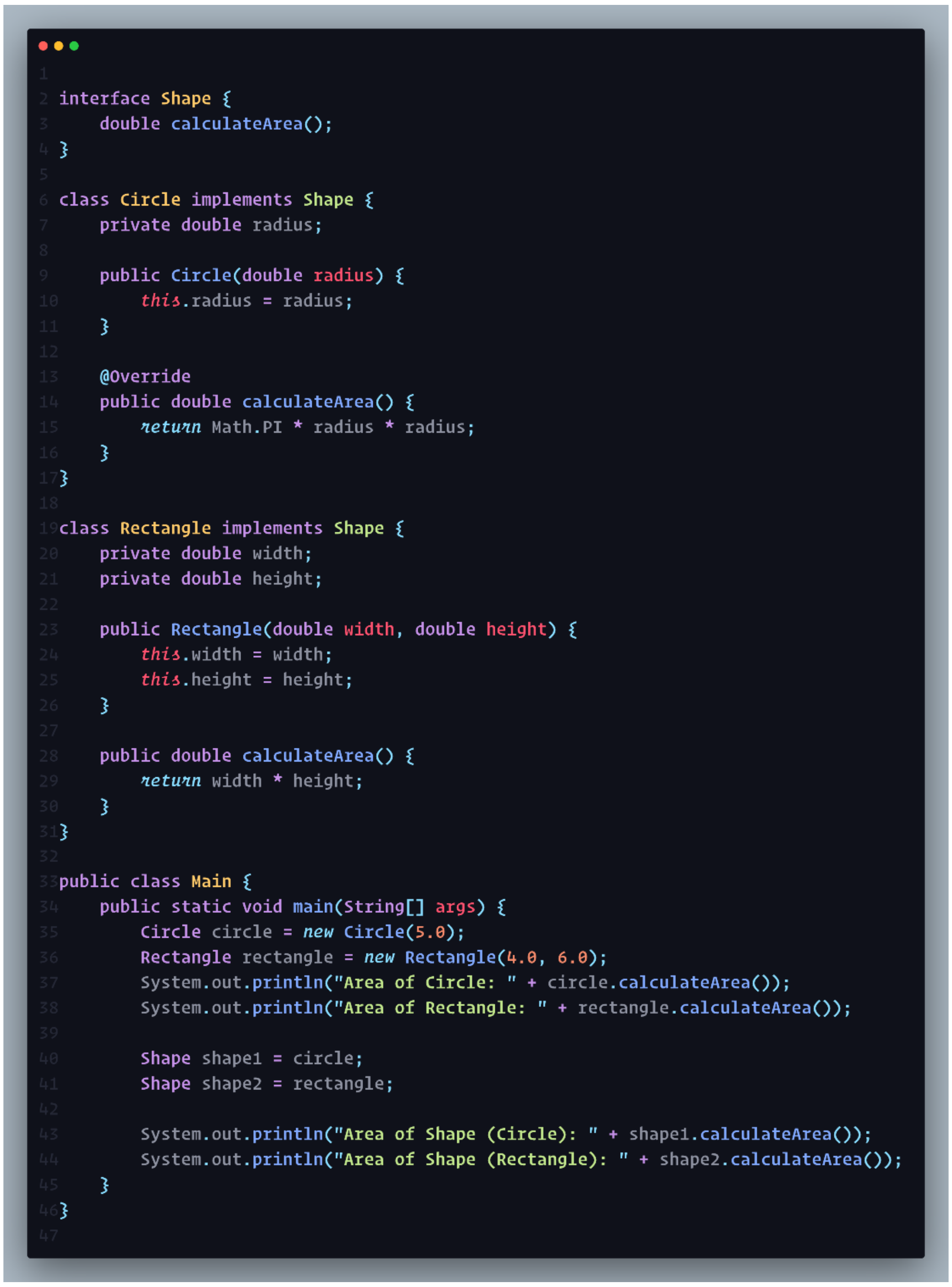
height (double): The height of the rectangle.

Implement the calculateArea() method in both the Circle and Rectangle classes to

calculate and return the respective areas.

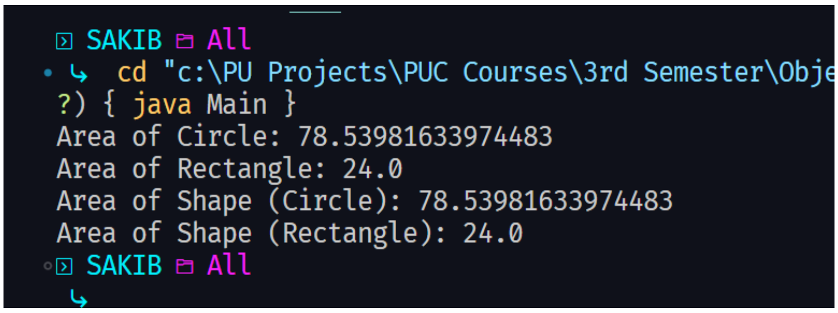
Define the Shape interface and implement it in the Circle and Rectangle classes. Demonstrate the usage of interfaces by creating instances of both shapes and calculating their areas. Ensure that your implementation highlights the concept of interfaces, method implementation, and polymorphism in Java.

Source Code :



```java
interface Shape {
    double calculateArea();
}

class Circle implements Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}

class Rectangle implements Shape {
    private double width;
    private double height;

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    public double calculateArea() {
        return width * height;
    }
}

public class Main {
    public static void main(String[] args) {
        Circle circle = new Circle(5.0);
        Rectangle rectangle = new Rectangle(4.0, 6.0);
        System.out.println("Area of Circle: " + circle.calculateArea());
        System.out.println("Area of Rectangle: " + rectangle.calculateArea());

        Shape shape1 = circle;
        Shape shape2 = rectangle;

        System.out.println("Area of Shape (Circle): " + shape1.calculateArea());
        System.out.println("Area of Shape (Rectangle): " + shape2.calculateArea());
    }
}
```

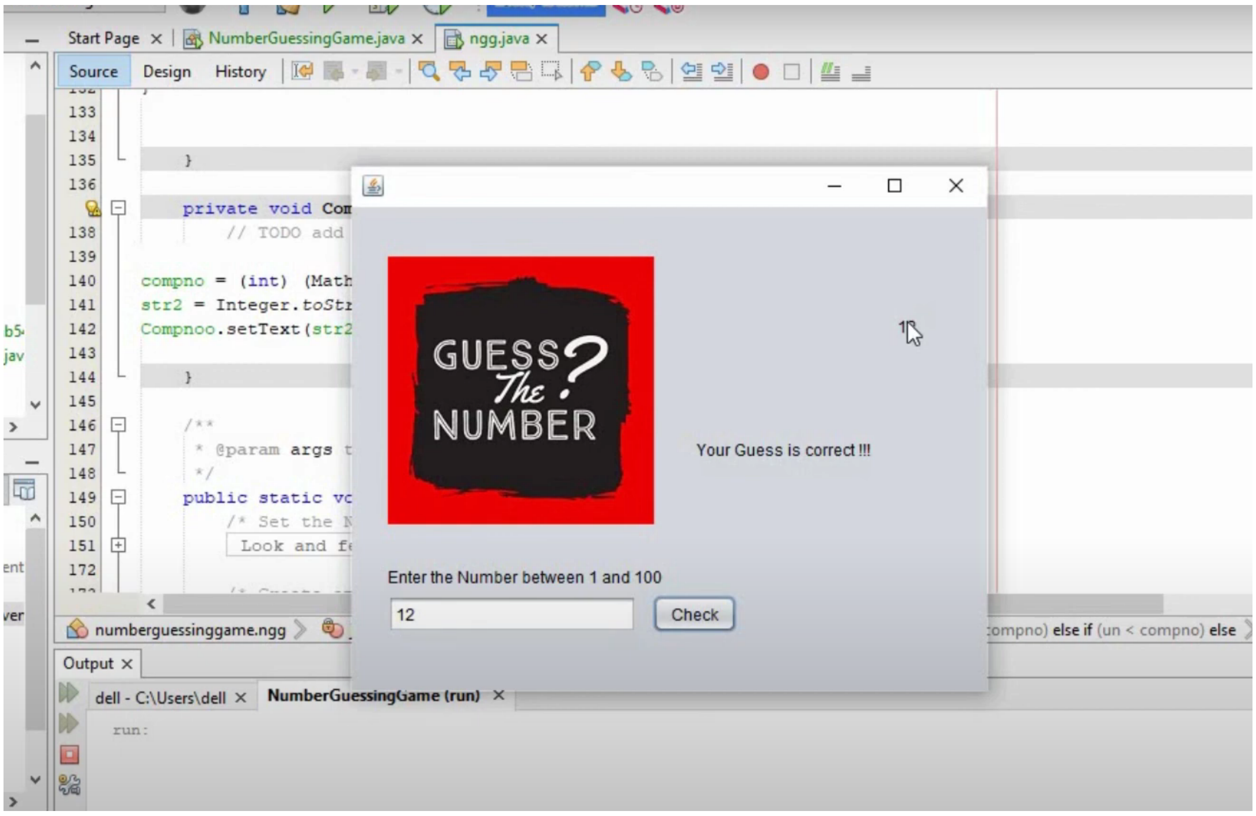
Output :



```
SAKIB  All
↳  cd "c:\PU Projects\PUC Courses\3rd Semester\Obje
?) { java Main }
Area of Circle: 78.53981633974483
Area of Rectangle: 24.0
Area of Shape (Circle): 78.53981633974483
Area of Shape (Rectangle): 24.0
SAKIB  All
↳
```

Statement : Create a Number Guessing Game in Java.

Source Code & Output :



```
    }

    private void Com
        // TODO add

compno = (int) (Math
str2 = Integer.toStr
Compnoo.setText(str2

    }

    /**
     * @param args t
     */
    public static vo
        /* Set the N
         Look and fe
```

compno) else if (un < compno) else

Your Guess is correct !!!

Enter the Number between 1 and 100

12

```
run:
```