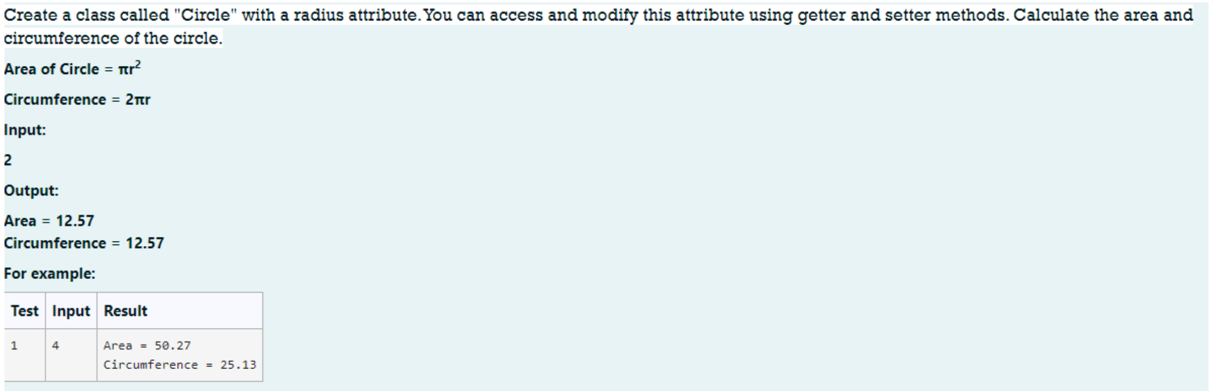
LAB-04-LOGIC BUILDING



Create a class called "Circle" with a radius attribute. You can access and modify this attribute using getter and setter methods. Calculate the area and circumference of the circle.

Area of Circle = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = 12.57
Circumference = 12.57

For example:

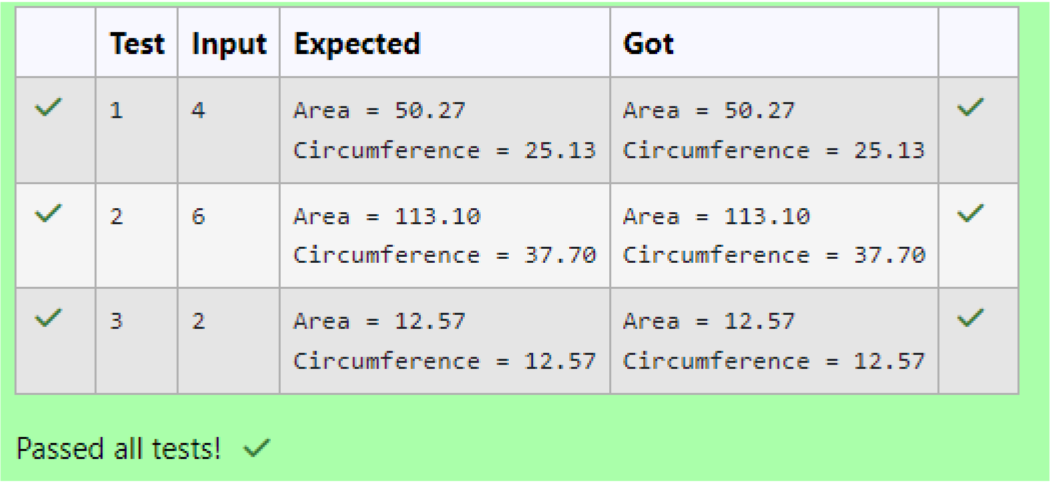
Test	Input	Result

1	4	Area = 50.27
		Circumference = 25.13

CODE:

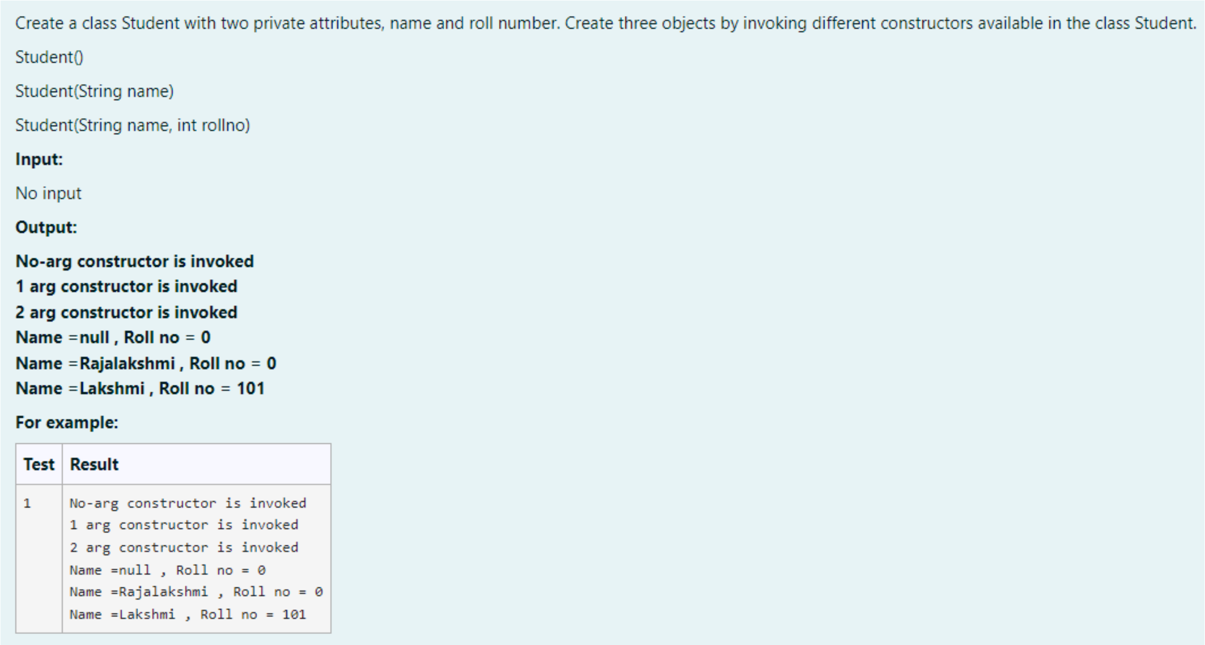
```
import java.util.Scanner;


class Circle {
    private double radius;


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


    public void setRadius(double radius) {
        this.radius = radius;
    }


    public double getRadius() {
        return radius;
    }


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

    public double calculateCircumference() {
        return 2 * Math.PI * radius;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double r = sc.nextDouble();

        Circle circle = new Circle(r);

        System.out.printf("Area = %.2f\n" ,circle.calculateArea());
        System.out.printf("Circumference = %.2f",circle.calculateCircumference());

        sc.close();
    }
}
```

OUTPUT:



	Test	Input	Expected	Got	
✓	1	4	Area = 50.27 Circumference = 25.13	Area = 50.27 Circumference = 25.13	✓
✓	2	6	Area = 113.10 Circumference = 37.70	Area = 113.10 Circumference = 37.70	✓
✓	3	2	Area = 12.57 Circumference = 12.57	Area = 12.57 Circumference = 12.57	✓

Passed all tests! ✓



Create a class Student with two private attributes, name and roll number. Create three objects by invoking different constructors available in the class Student.

Student()

Student(String name)

Student(String name, int rollno)

Input:

No input

Output:

No-arg constructor is invoked
1 arg constructor is invoked
2 arg constructor is invoked
Name =null , Roll no = 0
Name =Rajalakshmi , Roll no = 0
Name =Lakshmi , Roll no = 101

For example:

Test	Result

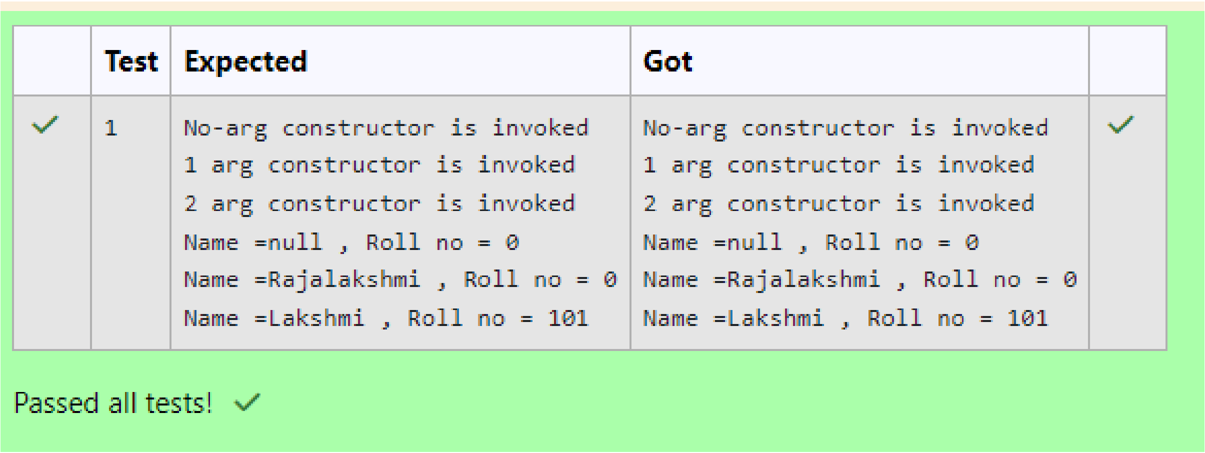
1	No-arg constructor is invoked
	1 arg constructor is invoked
	2 arg constructor is invoked
	Name =null , Roll no = 0
	Name =Rajalakshmi , Roll no = 0
	Name =Lakshmi , Roll no = 101

CODE:

```
public class Student {
   private String name;
   private int rollNo;

   // No-arg constructor
   public Student() {
      this.name = null;
      this.rollNo = 0;
      System.out.println("No-arg constructor is invoked");
   }

   // Constructor with one argument (name)
   public Student(String name) {
      this.name = name;
      this.rollNo = 0;
      System.out.println("1 arg constructor is invoked");
   }
```

```
    // Constructor with two arguments (name, roll number)
    public Student(String name, int rollNo) {
        this.name = name;
        this.rollNo = rollNo;
        System.out.println("2 arg constructor is invoked");
    }

    // Method to display student details
    public void displayInfo() {
        System.out.println("Name =" + this.name + " , Roll no = " + this.rollNo);
    }

    public static void main(String[] args) {
        // Creating objects using different constructors
        Student student1 = new Student();
        Student student2 = new Student("Rajalakshmi");
        Student student3 = new Student("Lakshmi", 101);

        // Displaying student details
        student1.displayInfo();
        student2.displayInfo();
        student3.displayInfo();
    }
}
```

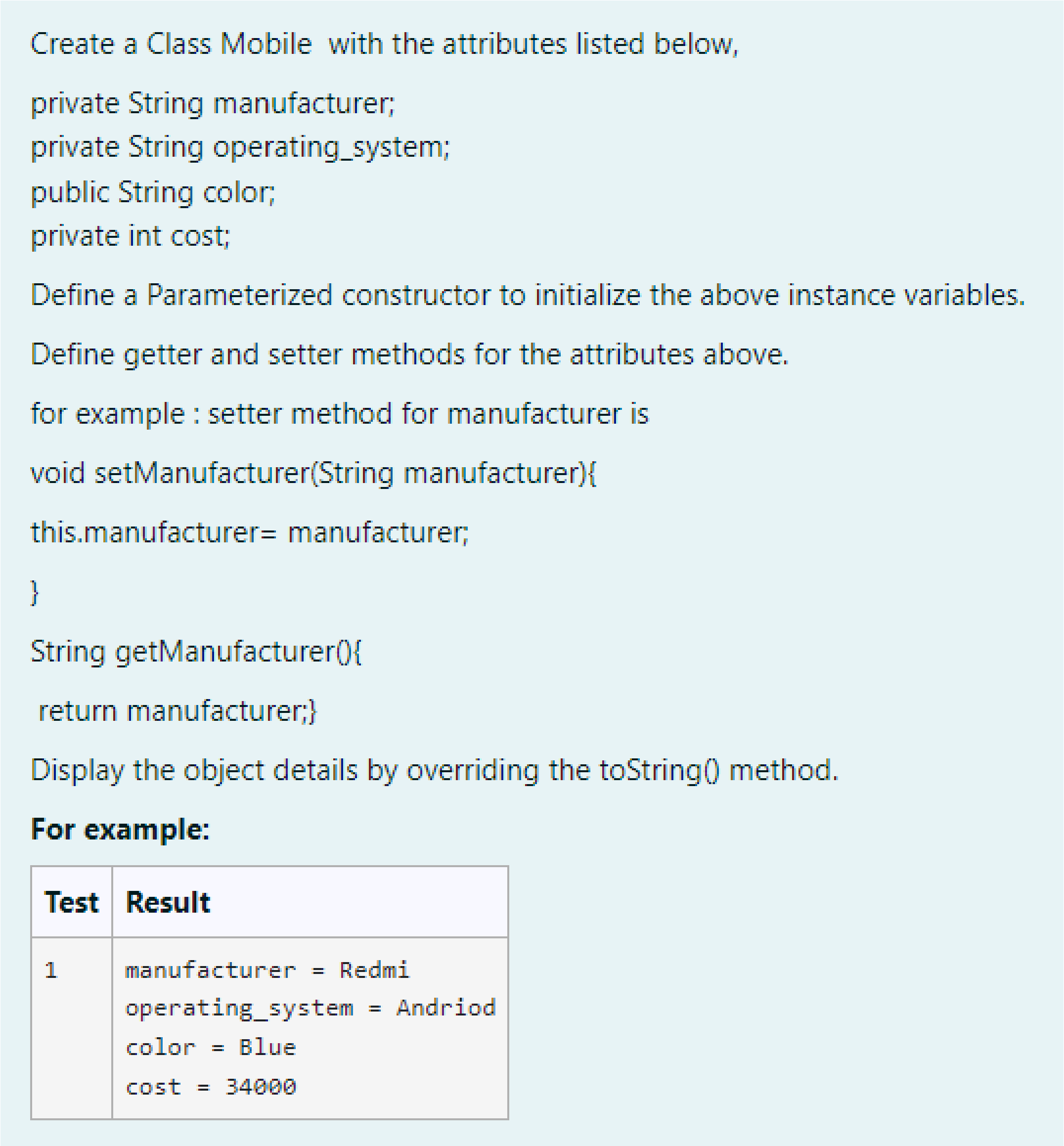
OUTPUT:



	Test	Expected	Got	

✓	1	No-arg constructor is invoked
		1 arg constructor is invoked
		2 arg constructor is invoked
		Name =null , Roll no = 0
		Name =Rajalakshmi , Roll no = 0
		Name =Lakshmi , Roll no = 101
			No-arg constructor is invoked
			1 arg constructor is invoked
			2 arg constructor is invoked
			Name =null , Roll no = 0
			Name =Rajalakshmi , Roll no = 0
			Name =Lakshmi , Roll no = 101
				✓

Passed all tests! ✓



Create a Class Mobile with the attributes listed below,

private String manufacturer;
private String operating_system;
public String color;
private int cost;

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

void setManufacturer(String manufacturer){

this.manufacturer= manufacturer;

}

String getManufacturer(){

return manufacturer;}

Display the object details by overriding the toString() method.

For example:

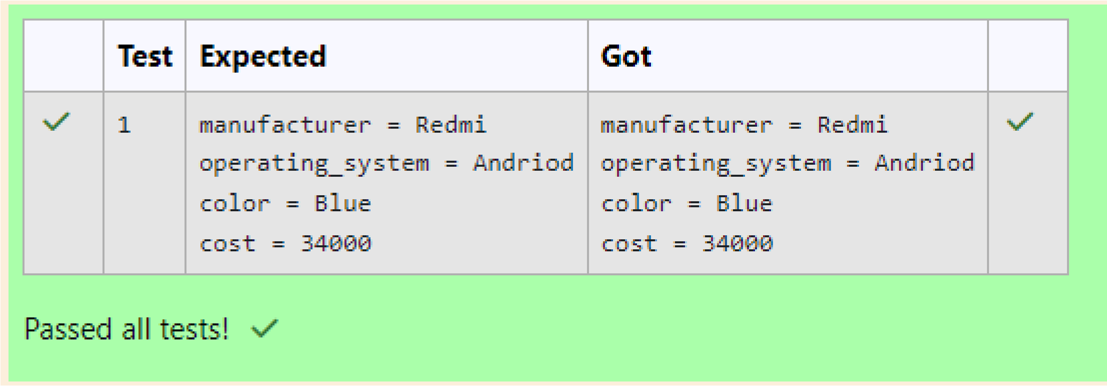
Test	Result

1	manufacturer = Redmi
	operating_system = Andriod
	color = Blue
	cost = 34000

public class Mobile {

```
    private String manufacturer;
    private String operatingSystem;
    private int cost;
    private String color;

    // Parameterized constructor to initialize the instance variables
    public Mobile(String manufacturer, String operatingSystem, int cost, String color) {
        this.manufacturer = manufacturer;
        this.operatingSystem = operatingSystem;
        this.cost = cost;
        this.color = color;
    }

    // Override toString() method to display object details
    @Override
    public String toString() {
        return "manufacturer = " + manufacturer + "\noperating_system = " +
operatingSystem + "\ncolor = " + color + "\ncost = " + cost;
    }

    public static void main(String[] args) {
        // Creating an object of Mobile class with the specified details
        Mobile mobile = new Mobile("Redmi", "Andriod", 34000, "Blue");

        // Displaying the details of the mobile
        System.out.println(mobile.toString());
    }
}
```

OUTPUT:



	Test	Expected	Got	

✓	1	manufacturer = Redmi
		operating_system = Andriod
		color = Blue
		cost = 34000
			manufacturer = Redmi
			operating_system = Andriod
			color = Blue
			cost = 34000
				✓

Passed all tests! ✓