Status	Finished
Started	Friday, 4 October 2024, 8:39 PM
Completed	Friday, 4 October 2024, 9:17 PM
Duration	37 mins 58 secs

Question **1**

Correct

Marked out of 5.00

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

Answer: (penalty regime: 0 %)

Reset answer

```
import java.io.*;
import java.util.Scanner;
class Circle
{
    private double radius;
    public Circle(double radius){
        this.radius=radius;
        // set the instance variable radius


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


    }
    public double getRadius()    {
        return radius;
        // return the radius


    }
    public double calculateArea()  {
        return Math.PI*radius*radius;// complete the below statement


    }
    public double calculateCircumference()    {
        // complete the statement
       return 2*Math.PI*radius;
    }
}
class prog{
    public static void main(String[] args)  {
        int r;
        Scanner sc= new Scanner(System.in);
        r=sc.nextInt();
        Circle c= new Circle(r);
        System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
        System.out.println("Circumference = "+String.format("%.2f",c.calculateCircumference()));
        // invoke the calculatecircumference method


    }
}

```

	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! ✓

Question **2**

Correct

Marked out of 5.00

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

Answer: (penalty regime: 0 %)

```
public class student{
    private String name;
    private int rollno;
    public student(){
        this.name=null;
        this.rollno=0;
        System.out.println("No-arg constructor is invoked");
    }
    public student(String name){
        this.name=name;
        this.rollno=0;
        System.out.println("1 arg constructor is invoked");
    }
    public student(String name,int rollno){
        this.name=name;
        this.rollno=rollno;
        System.out.println("2 arg constructor is invoked");
    }
    public void display(){
        System.out.println("Name ="+ name + " , Roll no = " + rollno);

    }
    public static void main(String[]args){
        student stu1=new student();
        student stu2=new student("Rajalakshmi");
        student stu3=new student("Lakshmi", 101);
        stu1.display();
        stu2.display();
        stu3.display();
    }
}
```

	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! ✓

Question **3**

Correct

Marked out of 5.00

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

Answer: (penalty regime: 0 %)

```
public class Mobile{
    private String manufacturer;
    private String operating_system;
    public String color;
    private int cost;
    public Mobile(String manufacturer, String operating_system, String color,int cost){
        this.manufacturer=manufacturer;
        this.operating_system=operating_system;
        this.color=color;
        this.cost=cost;
    }
    public void setmanufacturer(String manufacturer){
        this.manufacturer=manufacturer;

    }
    public String getManufacturer(){
        return manufacturer;
    }
    public void setOperatingSystem(String operating_system){
        this.operating_system=operating_system;
    }
    public String getOperatingSystem(){
        return operating_system;
    }
    public void setColor(String color){
        this.color=color;
    }
    public String getcolor(){
        return color;
    }
    public void setcost(int cost){
        this.cost=cost;
    }
    public int getCost(){
        return cost;
    }
    public String toString(){
        return "manufacturer = " + manufacturer + '\n' +"operating_system = "+ operating_system+ '\n' +"color = '
    }
```

```
    public static void main(String[] args){
        Mobile mo=new Mobile("Redmi", "Andriod", "Blue", 34000);
        System.out.println(mo);
    }
}
```

	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! ✓

◄ Lab-04-MCQ

Jump to...

Number of Primes in a specified range ►