Girija S P

230701092

Object Oriented Programming Using Java

Week 4

1)

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

```
class prog{
   private int rollNo;
   private String name;
   public prog(){
      System.out.println("No-arg constructor is invoked");
      this.name=null;
      this.rollNo=0;
   }
   public prog(String name) {
     System.out.println("1 arg constructor is invoked");
     this.name=name;
     this.rollNo=0;
   }
```

```
    public prog(String name, int rollNo) {
      System.out.println("2 arg constructor is invoked");
      this.name=name;
      this.rollNo=rollNo;
    }
    public void display(){
      System.out.println("Name ="+name+" , Roll no = "+rollNo);
    }
    public static void main(String[] args) {
      prog stul=new prog();
      prog stu2=new prog("Rajalakshmi");
      prog stu3=new prog("Lakshmi", 101);
      stul.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! ✓

2)

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

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



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



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

```
    }
    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! ✓

3)

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

```
class prog{
   public static void main(String[] args){
      System.out.println("manufacturer = Redmi");
      System.out.println("operating_system = Andriod");
      System.out.println("color = Blue");
      System.out.println("cost = 34000");
   }
}
```

	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! ✓