WEEK-4

ROLL NO:230701235

1) 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 = πr²

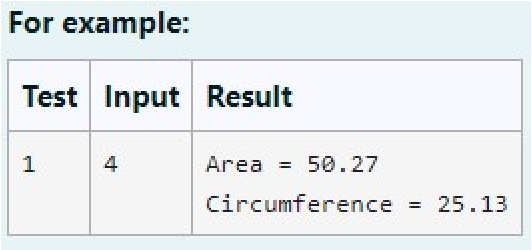
Circumference = 2πr

Input:

2

Output:

Area = 12.57
Circumference = 12.57



For example:

Test	Input	Result

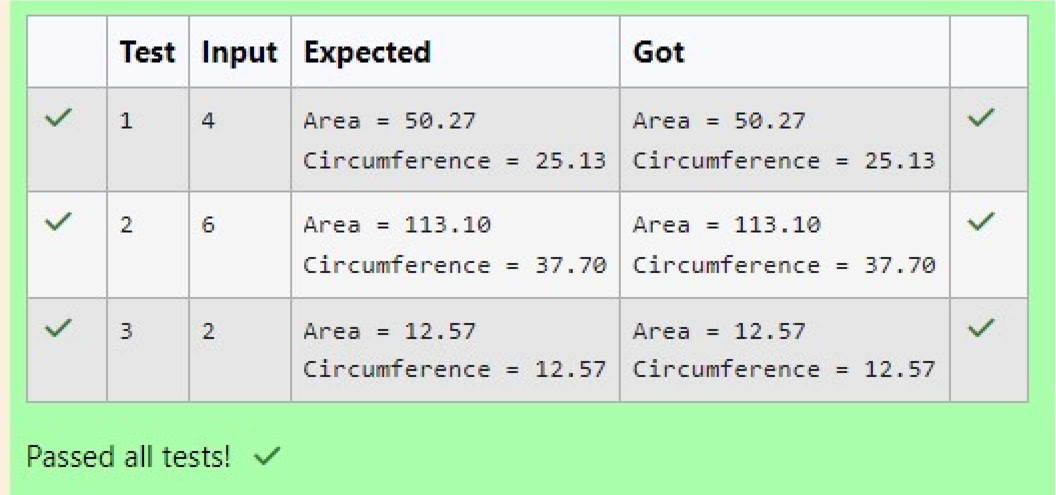
1	4	Area = 50.27
		Circumference = 25.13

CODE:

```
import java.io.*; import

java.util.*; 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;
   }
   public double getRadius()    {
return radius;
      // return the radius



   }
   public double calculateArea()  { // complete the below statement
return Math.PI*(Math.pow(radius,2));


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

```
    }
}
```

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! ✓

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

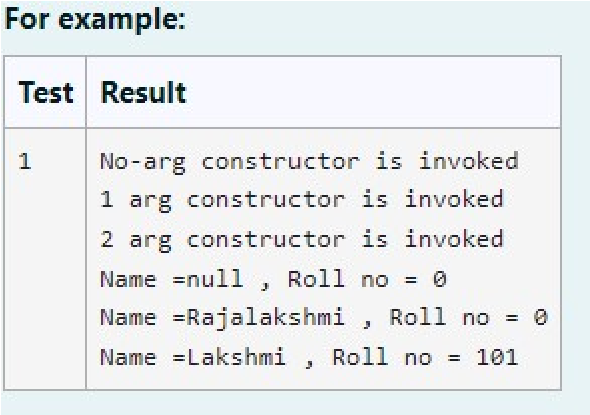
Student()

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:

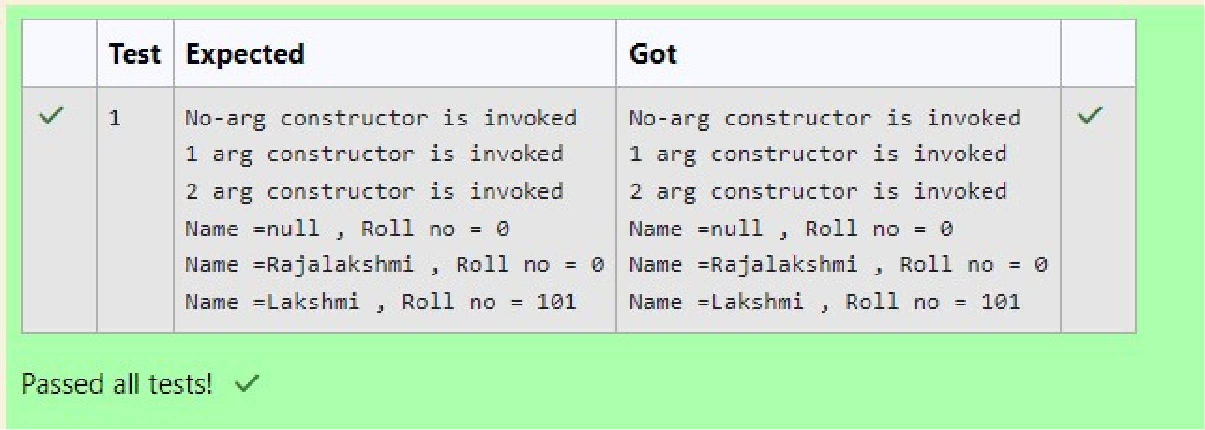
Test	Result

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;     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 [] a){
      Student stu1=new Student();
      Student stu2=new Student("Rajalakshmi");
Student stu3=new Student("Lakshmi",101);
stu1.display();        stu2.display();
stu3.display();


      }
}
```

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! ✓

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){

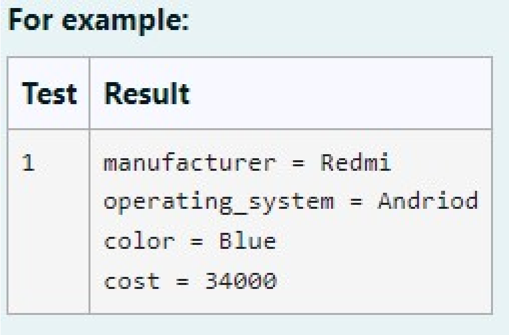
this.manufacturer= manufacturer;

}

String getManufacturer(){ return

manufacturer;}

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



For example:

Test	Result

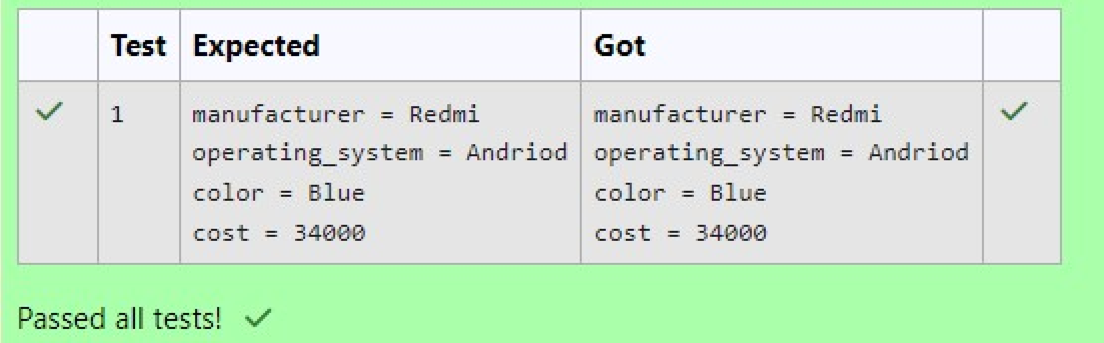
1	manufacturer = Redmi
	operating_system = Andriod
	color = Blue
	cost = 34000

CODE:

```
public class Mobile{     private String manufacture;     private String operating_system;
private String color;     private int cost;     public Mobile (String manufacture ,String
operating_system,String color,int cost){         this.manufacture=manufacture;
this.operating_system=operating_system;         this.color=color;         this.cost=cost;
   }
   public String getmanufacture(String manufacture){
return manufacture;
   }
   public void setmanufacture(String manufacture){         this.manufacture=manufacture;

   }
   public String getoperating_system(String operating_system){
return operating_system;
   }
   public void setoperating_system(String operating_system){
this.operating_system=operating_system;
   }
   public String getcolor(String color){
return color;
   }
   public void setcolor(String color){
this.color=color;
   }
   public int getcost(int cost){
return cost;
   }
   public void setcost(int cost){
this.cost=cost;
```

```
    }
    public String tostring(){        return "manufacturer =
"+manufacture+"\noperating_system = "+operating_system+"\ncolor
= "+color+"\ncost = "+cost;
    }
    public static void main(String [] args){
        Mobile mymobile= new Mobile("Redmi","Andriod","Blue",34000);
        System.out.println(mymobile.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! ✓