Assignment 4 on Session4: Abstraction

Problem Statement1:

Create an abstract class Figure with following properties and functions:

Properties: double dim1;

Methods: abstract void findArea(); abstract void findPerimeter();

Create three subclasses Circle, Rectangle and Triangle that extends Figure class and define both the methods. Write a program that will find the area and perimeter of 3 Figures and print the details for all.

Solution:

```
public class FindAreaPerimeter
{
  public static void main(String[] args)
  {
    // Circle
    double radius = 5;
    Figure Circle = new Circle(radius);
    System.out.println("Circle radius: " + radius
        + "\nResulting Area: " + Circle.findarea()
        + "\nResulting Perimeter: " + Circle.findperimeter() + "\n");
    // Rectangle
    double width = 5, length = 7;
    Figure rectangle = new Rectangle(width, length);
    System.out.println("Rectangle width: " + width + " and length: " + length + "\nResulting area: " +
rectangle.findarea()
        + "\nResulting perimeter: " + rectangle.findperimeter() + "\n");
    // Triangle
    double a = 5, b = 3, c = 4;
    Figure triangle = new Triangle(a,b,c);
    System.out.println("Triangle sides lengths: " + a + ", " + b + ", " + c
```

```
            + "\nResulting Area: " + triangle.findarea()
            + "\nResulting Perimeter: " + triangle.findperimeter() + "\n");
  }
}
abstract class Figure
{
  public abstract double findarea();
  public abstract double findperimeter();
}


class Circle extends Figure {
    private final double radius;
    final double pi = Math.PI;

    public Circle() {
        this(1);
    }
    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public double findarea() {

        return pi * Math.pow(radius, 2);
    }
```

```
    public double findperimeter() {

        return 2 * pi * radius;
    }
}

class Rectangle extends Figure {
    private final double width, length;

    public Rectangle() {
        this(1,1);
    }
    public Rectangle(double width, double length) {
        this.width = width;
        this.length = length;
    }

    @Override
    public double findarea() {

        return width * length;
    }

    @Override
    public double findperimeter() {

        return 2 * (width + length);
    }
```

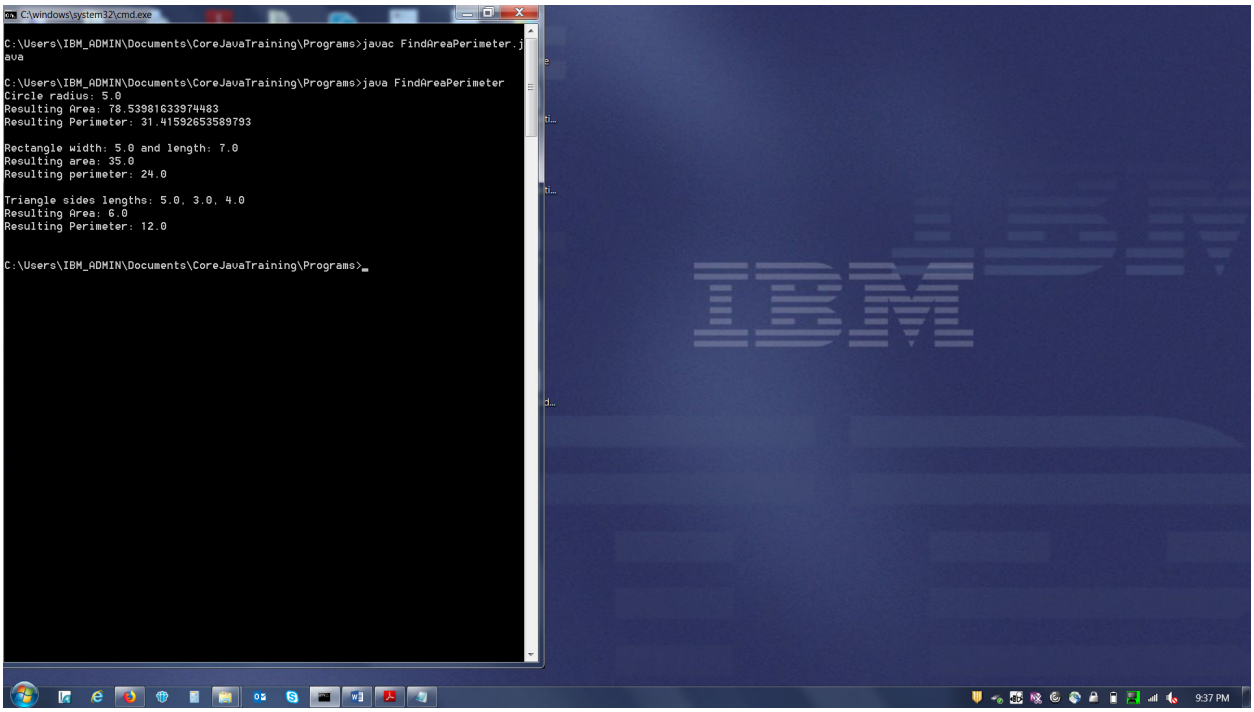
```
}

class Triangle extends Figure {
    private final double a, b, c;

    public Triangle() {
        this(1,1,1);
    }
    public Triangle(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    @Override
    public double findarea() {

        double s = (a + b + c) / 2;
        return Math.sqrt(s * (s - a) * (s - b) * (s - c));
    }

    @Override
    public double findperimeter() {

        return a + b + c;
    }
}
```

ASSIGNMENT 4

Explanation of the code:

This code deals with finding area and perimeter of different shapes such as Circle, Rectangle and Triangle.

Result flow & Screen shot:



Problem Statement2:

Declare an integer array of size 10. Initialize using for loop with 1 to 10, and print all even numbers from an array.

Solution:

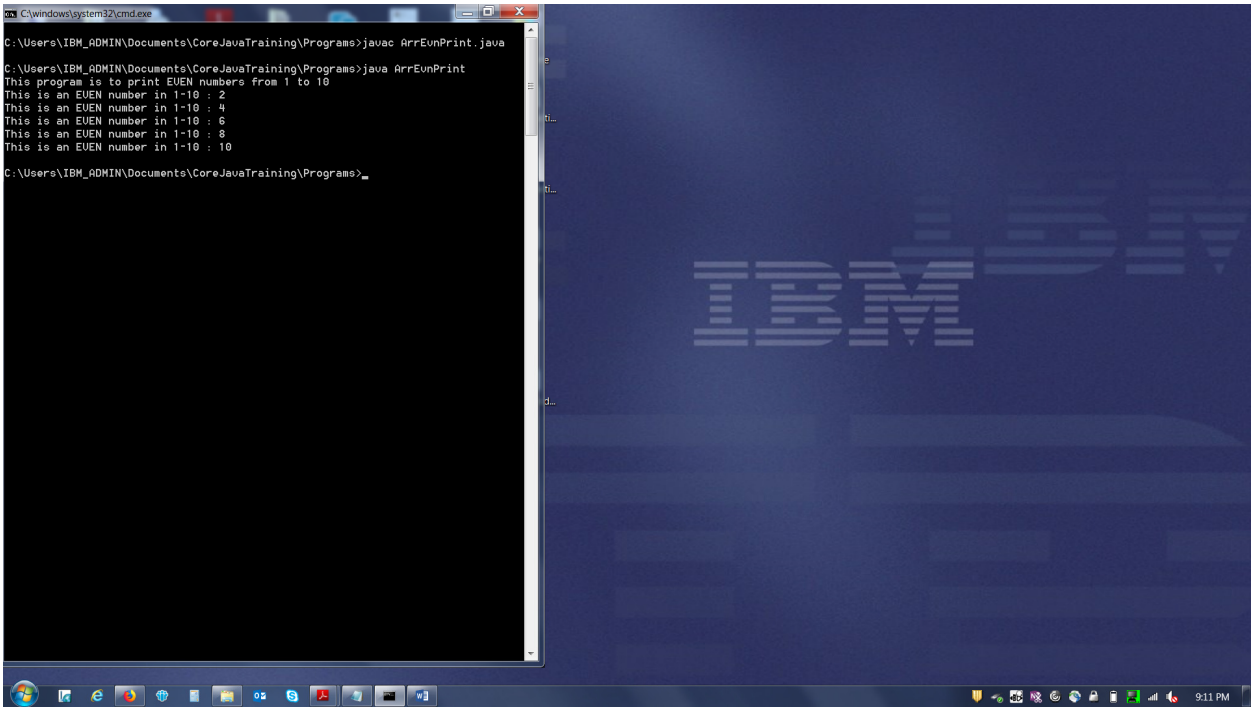
```
class ArrEvnPrint
{
  public static void main(String a[])
  {
  System.out.println("This program is to print EVEN numbers from 1 to 10");
  int[] arrnum=new int[10];
  for(int i=0; i<10; i++)
  {
  arrnum[i]=i+1;
  if(arrnum[i] % 2 == 0)
    System.out.println("This is an EVEN number in 1-10 : "+arrnum[i]);
  }
  }
}
```

Explanation of the code:

This code assigns integer values from 1 to 10 in an array. Then inside the FOR loop the values are assigned into each array and modulus value calculated by using % operator. If the remainder is 0 then its an even number and the number will be printed.

Result flow and screen shot:

ASSIGNMENT 4



Problem Statement3:

Write a program to generate a user-defined exception called NegativeAgeException if the user inputs negative value for age.

Solution:

```
import java.util.Scanner;


public class NegAgeExcep
{
  public static void main(String args[])
  {
    Scanner sc=new Scanner(System.in); //To get keyboard input
    System.out.println("Enter value for age: ");
    int input=sc.nextInt();
    if (input < 0)
    try
    {
      throw new NegativeAgeException("Please Enter Positive Integers");
    }
    catch (NegativeAgeException e) // TODO Auto-generated catch block
    {
      System.out.println(e.getMessage());
    }
    else
    System.out.println("Entered Number is a Positive Integer");
  }
}
class NegativeAgeException extends Exception
{
```

```
 private String message;
 public NegativeAgeException()
 {
  // TODO Auto-generated constructor stub
 }
 public NegativeAgeException(String message)
 {
  this.message= message;
  // TODO Auto-generated constructor stub
 }
 public String getMessage()
 {
  return message;
 }
 public void setMessage(String message)
 {
  this.message = message;
 }
}
```

Explanation of the code:

This program get input from user and validate whether it's a positive or a negative number.

Result flow & Screen shot:

ASSIGNMENT 4