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.

//Implement a Circle, Triangle, and Rectangle class which extend the class Shape.

//Shape.java

public abstract class Shape { // Shape class created

public abstract double area(); //Area method created

public abstract double perimeter(); // perimeter method created

}

//Rectangle.java

public class Rectangle extends Shape { // Rectangle class extended from Shape

private final double width, length; //rectangle sides

public Rectangle() {

this(1,1); // referring to current class object

}

public Rectangle(double width, double length) { // rectangle method created by passing width and length

this.width = width; // Width referring to currnet class object

this.length = length; //length referring to current class object

}

// Overriding

public double area() {

// A = w \* l

return width \* length;

}

// Overriding

public double perimeter() {

// P = 2(w + l)

return 2 \* (width + length);

}

}

//Circle.java

public class Circle extends Shape {

private final double radius;

final double pi = Math.PI;

public Circle() {

this(1); // referring to current class object

}

public Circle(double radius) {

this.radius = radius; //referring to current class object

}

// Overriding method area

public double area() {

// A = p r^2

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

}

public double perimeter() {

// P = 2pr

return 2 \* pi \* radius;

}

}

// Triangle.java

public class Triangle extends Shape {

private final double a, b, c; // Triangle sides

public Triangle() {

this(1,1,1); // referring to current class object

}

public Triangle(double a, double b, double c) {

this.a = a; //referring to current class object

this.b = b; // referring to current class object

this.c = c; // referring to current class object

}

// Overriding

public double area() {

// Heron's formula:

// A = SquareRoot(s \* (s - a) \* (s - b) \* (s - c))

// where s = (a + b + c) / 2, or 1/2 of the perimeter of the triangle

double s = (a + b + c) / 2;

return Math.sqrt(s \* (s - a) \* (s - b) \* (s - c));

}

// Overriding perimeter method

public double perimeter() {

// P = a + b + c

return a + b + c;

}

}

//TestShape.java

public class TestShape { // TestShape class created

public static void main(String[] args) {

// Rectangle test

double width = 5, length = 7;

Shape rectangle = new Rectangle(width, length); // passing width and legnth into rectangle method

System.out.println("Rectangle width: " + width + " and length: " + length

+ "\nResulting area: " + rectangle.area()

+ "\nResulting perimeter: " + rectangle.perimeter() + "\n");

// Circle test

double radius = 5;

Shape circle = new Circle(radius); // Constructor created

System.out.println("Circle radius: " + radius

+ "\nResulting Area: " + circle.area()

+ "\nResulting Perimeter: " + circle.perimeter() + "\n");

// Triangle test

double a = 5, b = 3, c = 4;

Shape triangle = new Triangle(a,b,c);

System.out.println("Triangle sides lengths: " + a + ", " + b + ", " + c

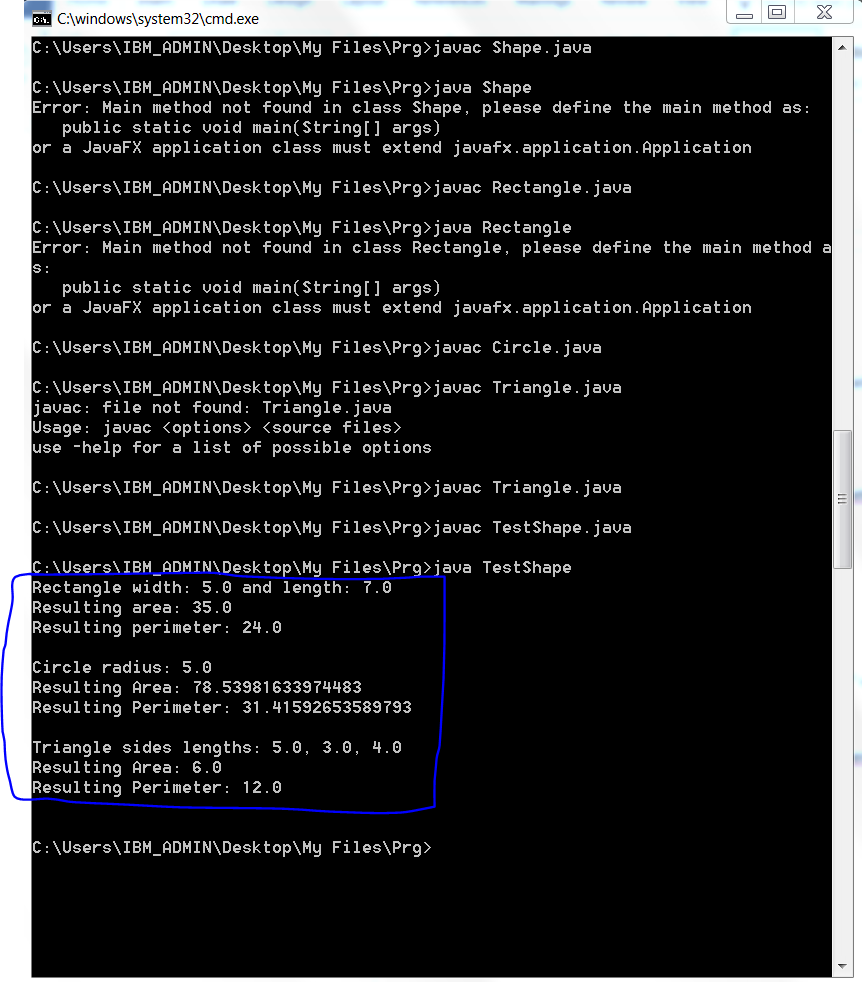
+ "\nResulting Area: " + triangle.area()

+ "\nResulting Perimeter: " + triangle.perimeter() + "\n");

}

}

Output:-



2) Declare an integer array of size 10. Initialize using for loop with 1 to 10, and print

all even numbers from an array.

class ArrayDemo

{

public static void main(String[] args)

{

int arr[]={1,2,3,4,5,6,7,8,9,10}; // Array initialized

for(int i=0;i<10;i++) // for loop to validate each value from array

{

if(arr[i]%2==0) // if condition to verify even numbers from above array

{

System.out.println("Even number from an array:" +arr[i]); // to print even numbers from array.

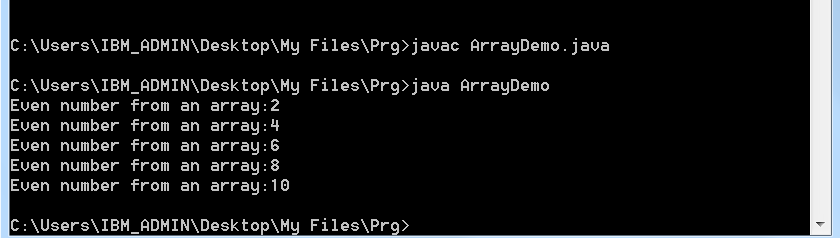
}

}

}

}

Output:-



3) Write a program to generate a user-defined exception called

NegativeAgeException if the user inputs negative value for age.

import java.util.Scanner;

class AgeException extends Exception { // class extended from Exception class

public AgeException(String str) { // Created a method with AgeException

System.out.println(str);

}

}

public class AgeExcDemo { // Class AgeExcDemo created

public static void main(String[] args) {

Scanner s = new Scanner(System.in); // scanner constructor to accept input value and read

System.out.print("Enter ur age :: ");

int age = s.nextInt(); // input value

try {

if(age < 18) // if condition to validate age

throw new AgeException("Invalid age");

else

System.out.println("Valid age");

}

catch (AgeException a) { // Catch block to throw error message(catch exceptions) if we got any

System.out.println(a);

}

}

}

Output:-

