Advanced Object Oriented Programming – Inheritance,

Polymorphism, Abstract Classes

1. Write a set of classes that could be used to do simple calculations on geometric objects. The set should include the following classes
   1. Shape, containing abstract methods area and perimeter  
      abstract class Shape{

abstract in area();

abstract int perimeter();

}

* 1. Rectangle, a subclass of Shape with instance fields length and width, and containing a constructor (with parameters to set the instance fields) along with definitions for the methods area and perimeter  
     class Rectangle extends Shape{

protected int length;

protected int width;

}

* 1. Triangle, a subclass of Shape with three instance fields for the length of each of the triangle’s sides, and containing a constructor (with parameters to set the instance fields) along with definitions for the methods area and perimeter.  
     class Triangle extends Shape

{

private int side1;

private int side2;

private int side3;

//using heron's formula, calculates the area of a Triangle

public int area ()

{

int semiPerimeter;

semiPerimeter = (side1+side2+side3)/2;

return (int)Math.sqrt(semiPerimeter\*(semiPerimeter - side1)\*(semiPerimeter - side2)\*(semiPerimeter - side3));

}

//returns the perimeter using the formula, perimeter = sum of sides (side1 + side2 + side3)

public int perimeter()

{

return side1+side2+side3;

}

}

To determine the area of your triangle, you may want to look up “Heron’s Formula”.

* 1. Square, a subclass of Rectangle, containing a constructor (with a parameter for the length of a side) and methods that invoke the methods of the Rectangle class to determine the perimeter and area of a square. The constructor should use the super constructor but ensure that both the length and width are set with the side length provided.

class Square extends Rectangle

{

//constructor that ensures the

public Square(int length)

{

super(length,length);

}

}