**Lab 8 Shape**

`

**Interface:** Write an interface called Shape. All objects of classes which implement Shape will have the following methods:

* area() and perimeter()
* showArea() and showPerimeter() which returns the calculations as a string Hint: Do not repeat the calculations in these methods
* showDescript() which returns the name of the shape as a string

Remember to add an interface to your Netbeans project, in the same way that you’ve been adding classes.

**Rectangle Class**: Write a Rectangle class which implements Shape with any necessary constructors, accessors, mutators, and additional methods. What instance variables should Rectangle have?

**Square Class:** Write a Square class that is derived from Rectangle. Deal with the fact that a square has a length and width with are the same. How can you make this work without requiring a potential user to enter the same value twice? (There will be no user input in this lab, but there may be in a future lab. And it’s always good to be thinking ahead to what a user might do.)

**Triangle Class:** Write an abstract triangle class that implements Shape. (Hint: Which method should be made abstract?)

**Right Triangle Class:** Write a right triangle class that is derived from Triangle. Include a two parameter constructor that takes the legs of the triangle as input and calculates the hypotenuse.

**Isosceles Triangle Class:** Derived from the Triangle class. Write appropriate constructors and methods.

**Equilateral Triangle Class**: Derived from the Isosceles class. Write appropriate constructors and methods.

**Scalene Triangle Class:** Derived from the Triangle class. Write appropriate constructors and methods.

Hint: Heron’s formula will probably be useful in Triangle.

**Client Code:** Write a program called ShapeApp in which at least one object of each Shape type is stored in an array. Sort the array in ascending order by the area of the shape. Use the sorting algorithm of your choice.