**HOMEWORK OOP1**

Write a superclass called Shape, which contains:

* Two instance variables color (String) and filled (boolean).
* Two constructors: a no-arg (no-argument) constructor that initializes the color to "green" and filled to true, and a constructor that initializes the color and filled to the given values.
* Getter and setter for all the instance variables. By convention, the getter for a boolean variable xxx is called isXXX() (instead of getXxx() for all the other types).
* A toString() method that returns "A Shape with color of xxx and filled/Not filled".

Write a test program to test all the methods defined in Shape.

Write two subclasses of Shape called Circle and Rectangle.

The Circle class contains:

* An instance variable radius (double).
* Three constructors: a no-arg (no-argument) constructor that initializes the radius to 1.0, a constructor that initializes only the radius field and a constructor that initializes the radius, color and filled fields.
* Getter and setter for the instance variable radius.
* Methods getArea() and getPerimeter().
* Override the toString() method inherited, to return "A Circle with radius=xxx, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

The Rectangle class contains:

* Two instance variables width (double) and length (double).
* Three constructors: the no-arg constructor that initializes the width and length to 1.0, a constructor that initializes the width and lenght fields and a constructor that initializes the width, length, color and filled fields.
* Getter and setter for all the instance variables.
* Methods getArea() and getPerimeter().
* Override the toString() method inherited, to return "A Rectangle with width=xxx and length=zzz, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

Write a class called Square, as a subclass of Rectangle.

* Three constructors: a no-arg constructor that initializes the side of the square to 1.0, a constructor that initializes the side of the square to the given value and a constructor that initializes side, color and filled to the given values.
* Getter and setter for the side of the square
* Override the toString() method to return "A Square with side=xxx, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.
* Do you need to override the getArea() and getPerimeter()? If you override the methods please leave comments in the methods why you chose to override them, otherwise leave a comment in class why they shouldn't be overriden.