Lab 2 Circles

Purpose

The purpose of this lab is for you to practice material covered during Weeks 4 and 5, and Lab 1 Cars.

Problem specification

The `Circle` class represents circles. A circle has a color (e.g. "red", "blue", "yellow"), and a radius. Separate `Circle` methods must return the color, calculate and return the circumference, and calculate and return the area.

The `CircleTester` class uses the `Circle` class, gets user input to create a new circle and does calculations on it.

The `Circle` class

Here is a partial pseudocode outline of `Circle`:

```
constants
    set PI to 3.1415926

instance variables
    as specified above

methods
    constructor(radius parameter, color parameter)
        initialize all instance variables

    design other methods, parameters and return types as specified above
    (IMPORTANT NOTE: none of the Circle methods print anything out.
    Instead, all of them return a value back to the method calls in CircleTester)
    (algorithm for circumference is $2\pi r$)
    (algorithm for area is $\pi rr$)
```

The `CircleTester` class

Put all `CircleTester` code into the `main()` method. Declare in `main()` any local variables that are needed. Here is a pseudocode outline:

```
instance variables
    none
```

methods

- `main()`
    - prompt user for and read from the keyboard a radius
    - prompt user for and read from the keyboard a color (hint: use `next()`)
    - create a new `Circle` object with this radius and color
    - print the color of the circle. Call a `Circle` method as you do this
    - print the circumference of the circle (to 2 decimal places) Call a `Circle` method as you do this
    - print the area of the circle (to 2 decimal places) Call a `Circle` method as you do this

So a run of your program would look something like:

```
Enter radius: 10.5
Enter color: red

For the red circle
Circumference is: 65.97
Area is: 346.36
```

Hints

You will be writing two new classes called `Circle` and `CircleTester`.

To help you with this lab, make sure you first download and study carefully the Week 4 and 5 example programs.

Then follow the usual development process (covered in the Week 4 lecture):

- on a piece of paper, begin the design of your new classes

- identify instance variables and data types

- identify methods, design algorithms, think about parameters and return values

- the `Circle` constructor has two parameters. When writing the `Circle` constructor method, remember to say the data type for each of the two formal parameters. In `CircleTester`, to call this constructor, list the two actual parameter values

    – as an example, if in the class `Foo`, the constructor has these two params:

    ```
    public Foo(int size, String taste)
    {
      // code to initialize all Foo instance variables
      // goes here
    ```

```
        }
```

- then in another class, to call this constructor:

  ```
  Foo obj = new Foo(123, "yukky");
  ```

- also, you can use variables in the method call. For example, say that `int x` has the value `123` and `String s` has value `"yukky"`, could say:

  ```
  Foo obj = new Foo(x, s);
  ```

- also, in Blackboard, Course Documents, Week 5 folder, Example programs, study the ‘Two parameters’ example program. This shows the `Savings` class constructor, that demonstrates writing and calling a constructor that takes 2 parameters

- now use BlueJ to write and test your methods as you go…
- (create a new project in BlueJ, then click **New Class…** to start writing your program)
- write a method, create a new object and test the new method
- finish one method before starting the next
- finally, have `CircleTester` run your `Circle` class. Check that the output in the Terminal Window is correct, then use **Options | Save to file…** to save your output file as `output.txt`

Required
Failure to meet these requirements will cost you points:

- `PI` must be declared as a constant in the `Circle` class
- every method you write must have a comment
- when your program is tested and correct, you must run it for a “blue” circle of radius 5.5 and save these results as your `output.txt` output file
- circumference and area must be output to 2 decimal places
- automatically and routinely use all the other components of simplicity and clarity, as listed in Blackboard, Course Information, “How labs are graded”

Lab 2 submission

- deadline for this lab is 1 week, by end of Sunday 9/25

- zip your BlueJ project plus `output.txt` output file and email to me at awsmith@palomar.edu

  – you will lose points if you do not include a file named `output.txt` containing the output of your program

- your email Subject line must say ‘CSCI 114 Lab 2’ followed by your full name, so that it filters to the correct email folder for grading

  – you will lose points if you format your email Subject incorrectly

  – e.g. my email Subject would be:

  ```
  CSCI 114 Lab 2 Anthony W. Smith
  ```

- this is a graded lab, so a reminder that you may not copy code from other people

- reminder that late labs will be penalized 3 points per week or part of week late