CSC 110: Introduction to Computer Programming – HW 3

Possible Points: 25 points

Due Date: Please see our Canvas class site for this info

General:

This homework has the following goals:

- Give you practice with graphical programming and with using the graphical library
- Introduce you to concepts of object-oriented programming
- To give you practice creating objects with values determined from user input
- Introduce the concept of sequences, as used in Python
- Give you practice manipulating strings and help you understand how strings are represented in the computer – one of the great things about Python is that it has many methods that facilitate string manipulation

Assignment:

1. (10 points) Problem 1 gives you practice creating and manipulating graphical objects.

    a. (7 points) Write a program **target1.py** as described in Programming Exercise 2 on page 126 of the textbook (2nd edition page 118).

    b. (3 points) Modify your program from part (a) above to make it interactive. Allow the user to specify the *diameter of the outermost circle of the target*. You may get this value from the user in a similar manner as the *principal* and *apr* were obtained in the **futval_graph2.py** program on page 105 of the textbook (2nd edition page 101). You will need to have the graphics window adjust its size to accommodate the archery target that will be created within it. Save your new program as **target2.py**.

    Hint: You will ask the user for the diameter of the archery target. How is this related to the radius of the inner circle? The larger circles' radii can be expressed as multiples of this value.

    Submit your responses to Problem 1 as two separate modules (**target1.py** and **target2.py**).

2. (15 points) Problem 2 gives you practice with interactive graphics that use mouse clicks rather than the console to get user input. It also gives your practice using coordinate transformation to make placing text in the window easier.

    a. (10 points) Write a program **line1.py** as described in Programming Exercise 8 on page 127 of the textbook (2nd edition page 119).

b. (5 points) Modify your program from part (a) so you have a second program named **line2.py**. This program will have the midpoint labeled, offset slightly from the actual location of the midpoint of the line.

Hint: To create labels that combine text and numerical values, you may find using the type conversion function str(<expr>) helpful. For more, see page 152 of the textbook (2nd edition page 146).

Submit your responses to Problem 2 as two separate modules (**line1.py** and **line2.py**).

Details:

As you have done on the earlier homeworks, add comments to your Python programs describing what a line or a block of code is meant to accomplish, including your name, and other important details as a title header at the beginning of the module.

Turn In:

Submit all of your python programs for this assignment as .py files, a total of 4 modules.