Lab 5

Due May 24, 11:59 PM

Overview

In this lab, we will be exploring various ways C++ represents data structures. You will accomplish the following:

- Gain experience with declaring, initializing, and using C-style structures.
- Gain experience with declaring, initializing, and using C++ classes and objects.

We will work through two different ways to implement a small library to help work with circles. Exercise 1 will focus on a C-style approach using structures, and Exercise 2 will focus on a C++ approach using classes.

For this lab, it may be useful to review the following sections:

- Data structures
- Other data types
- Classes (I)

Exercise 1

Download the template `struct_template.cpp` from Canvas. For this exercise, you will finish the implementation of the Circle struct and helper functions.

The structure `struct circle_st` should have three fields: `x`, `y`, and `radius`. All fields should be of type `float`.

You must implement the following functions:

- `printCircle` – Print the circle structure in an easy to read format:

```
Center: (0, 0)
Radius: 1
```

- `area` – Compute and return the area for the circle.
- `circumference` – Compute and return the circumference for the circle.
- `concentric` – Returns `true` if `c1` and `c2` are concentric (they have the same center), otherwise returns `false`.

Sample Output

```
Center: (0, 0)
Radius: 1
C1 Circumference      6.28
C1 Area               3.14

Center: (2.50, 2.50)
Radius: 1.00
C2 Circumference      6.28
C2 Area               3.14

Center: (0.00, 0.00)
Radius: 10.00
C3 Circumference     62.83
C3 Area             314.16

C1 and C2 are concentric? false
C1 and C3 are concentric? true
C2 and C3 are concentric? false
```

Exercise 2

Download the template `class_template.cpp` from Canvas. For this exercise, you will finish the implementation of the Circle class and helper methods.

The Circle class must have three members: `x`, `y`, and `radius`. All members should be of type `float`.

You must implement the following:

- `Circle::Circle` constructor – Implement the constructor for the class. The constructor should ensure that the `radius` is not below 0.
- `Circle::print` – Print the circle in the same format as Exercise 1.

- `Circle::area` – Compute and return the area of the circle object.
- `Circle::circumference` – Compute and return the circumference of the circle object.
- `Circle::concentric` – Returns `true` if the circle object is concentric with circle object `c`, otherwise returns `false`. Use the same definition for concentric as Exercise 1.

Sample Output

```
Center: (0, 0)
Radius: 1
C1 Circumference:      6.28
C1 Area:               3.14

Center: (2.50, 2.50)
Radius: 1.00
C2 Circumference:      6.28
C2 Area:               3.14

Center: (0.00, 0.00)
Radius: 10.00
C3 Circumference:     62.83
C3 Area:             314.16

C1 and C2 are concentric? false
C1 and C3 are concentric? true
C2 and C3 are concentric? false
```

Submission

Share your REPL with your TA by copying the shareable link and submitting it to Canvas.

To get your shareable link, click the large **Share** button near the top of the page, then copy the URL next to **Share Link** and submit that to Canvas.

Rubric

Exercise 1	6
Exercise 2	3
Comments	1
Total	10