# CSC 232 Lab 6 Spring 2018

## Submission requirements:

## All files are to be placed in a folder named as Lab6 in your folder on Trace Server

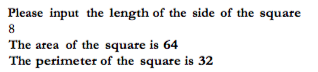
## Also put your folder name on Blackboard.

## The due date will be the midnight of next Friday. (March 9)

Part 1: Squares as a Class

Open program square.cpp from the Lab 6 folder.

Exercise 1: This program asks you to fill in the class declaration and client code based on the implementation of the member functions. Fill in the code so that the following input and output will be generated.



Exercise 2: Add two constructors and a destructor to the class and create the implementation of each. One constructor is the default constructor that sets the side to 1. The other constructor will allow the user to initialize the side at the definition of the object. The destructor does not have to do anything but reclaim memory space. Create an object called box1 that gives the value of 9 to the constructor at the definition. Add output statements so that the following is printed in addition to what is printed in Exercise 1.



Part 2: Circles as a Class

Open program circles.cpp from the Lab 6 folder.

Exercise 1: Alter the code so that setting the center of the circle is also done during the object definition. This means that the constructors will also take care of this initialization. Make the default center at point (0, 0) and keep the default radius as 1. Have sphere defined with initial values of 8 for the radius and (9, 10) for the center. How does this affect existing functions and code in the main function?

The following output should be produced:

**The radius of the circle is 8**

**The center of the circle is (9, 10)**

**The area of the circle is 200.96**

**The circumference of the circle if 50.24**

Exercise 2: There can be several constructors as long as they differ in number of parameters or data type. Alter the program so that the user can enter either just the radius, the radius and the center, or nothing at the time the object is defined. Whatever the user does Not include (radius or center) must be initialized somewhere. There is no setRadius function and there will no longer be a setCenter function. You can continue to assume that the default radius is 1 and the default center is (0, 0). Alter the client portion (main) of the program by defining an object sphere1, giving just the radius of 2 and the default center, and sphere2 by giving neither the radius nor the center (it uses all the default values). Be sure to print out the vital statistics for these new objects ( area and circumference).

In addition to the output in Exercise 1, the following output should be include:

**The radius of the circle is 2**

**The center of the circle is (0, 0)**

**The area of the circle is 12.56**

**The circumference of the circle is 12.56**

**The radius of the circle is 1**

**The center of the circle is (0, 0)**

**The area of the circle is 3.14**

**The circumference of the circle is 6.28**

Exercise 3: Alter the program you generated in Exercise 2 so that the user will be allowed to enter either nothing, just the radius, just the center, or both the center and radius at the time the object is defined. Add to the client portion of the code an object called sphere3 that, when defined, will have the center at (15, 16) and the default radius. Be sure to print out this new object’s vital statistics (area and circumference).

In addition to the output in Exercise 1 and 2, the following output should be printed.

**The radius of the circle is 1**

**The center of the circle is (15, 16)**

**The area of the circle is 3.14**

**The circumference of the circle is 6.28**

Exercise 4: Add a destructor to the code. It should print the message This concludes the Circles class for each object that is destroyed.

Part 3: Arrays as Data Members of Classes

Open program floatarray.cpp and temperatures.txt from the Lab 6 folder.

Exercise 1: Fill in the code so that the program reads in the data values from the temperature file and prints them to the screen with the following output:

**78.90**

**87.40**

**60.80**

**70.40**

**75.60**

Exercise 2: Add code (member function, call and function implementation) to print the average of the numbers to the screen so that the output will look like the output from Exercise 2 plus the following:

**The average temperature is 74.62**

Part 4: Arrays of Objects

Open program inventory.cpp and the data file inventory.dat from the Lab 6 folder.

Exercise 1: Complete the program by giving the code explained in the commands in bold. The data file is as follows:

**986 8**

**432 24**

**132 100**

**123 89**

**329 50**

**503 30**

**783 78**

**822 32**

**233 56**

**322 74**

the output should be as follows:

