# Object-Oriented Programming I Fall 2016

**CIS 9310**

## **Programming Assignment 4**

For this assignment you are to begin by implementing the class **Rectangle**. Objects of his class, as implied by its name, are intended to model rectangles, where a rectangle is characterized by a length and width, each of which is a **Length** class object. (This Length class is the one implemented in the previous assignment.) The **Rectangle** class should include the following set of functions:

Constructors:

**Rectangle();**

Default constructor initializing an object to a width and length of 0;

**Rectangle(Length ln, Length w);**

Initialization constructor initializing an object to a length of **ln** and a width of **w**.

Access Functions:

**Length getLength();**

**Length getWidth();**

**double getArea();**

This function returns the area of a rectangle in square feet expressed as a doouble.

**bool EqualAreaTo(Rectangle);**

**bool GreaterAreaThan(Rectangle);**

**bool LessAreaThan(Rectangle);**

Modifier Functions:

**void setLength(Length ln);**

**void setWidth(Length w);**

Input/Output Functions:

**void Read();**

This function prompts for (separately) both the length and width of a **Rectangle** object.

**void Write();**

This function when sent to a **Rectangle** object having length **ln** and width **w**, outputs the line

**Length ln, Width w**

where **ln** and **w** are expressed as **Length** class values in the form *f’i”.*

Using the **Rectangle** class you are to implement an application that reads and then outputs in sorted order a sequence of up to 100 **Rectangle** values in ascending area order. You can decide on how you prompt the user to enter the values. For example, you repeatedly ask the user if there is another value to be entered, and when the user responds in the affirmative, prompt the user to enter the value; or you can ask the user upfront how many values are to be input, and then prompt the user that many times to enter a value.

Your implementation should include, except as specified, the following code segment:

int main()

{

//here you should define the data needed to complete this

//function

ReadData(rList, rCount);

SortData(rlist, rCount);

WriteData(rlist, rCount);

Return 0;

}

The ReadData() function retrieves from the user a list of rectangle values that are passed back in the array specified by the first parameter. The second parameter is used to pass back to the calling function, in this case main(), the number of rectangle values contained in the array specified by the first parameter.

The SortData() function sorts by area the rectangle values contained in the array passed in by the first parameter. On return, this parameter should contain the sorted rectangle values. The second parameter is used to specify to the SortData() function the number of values contained in the first parameter.

The WriteData() function outputs, one per line, the elements in the array specified by the first parameter. The second parameter specifies the number of elements in the array that are to be output, starting from the first array element.

Your program must include the use of the 3 function calls as shown above, and they perform only the specified tasks. These statements may not be altered in any way. However your main() function may, and should, include additional statements to define the data to be used that is consistent with the given function calls.

You will also have to include the declaration/definitions of each of the functions called in main();

Due Date: December 1, 2016