

CSCI 1301 - Lab 09

Clément Aubert

February 1, 2018

Deadlines: This lab needs to be completed before taking Lab 10, and is critical to complete Project #2.

Dependencies: Part I is about finishing Lab 8, and it is recommended that you complete this part before starting Part II.

Part 0 - Get Your Quiz Back

If time allows, I'll grade your quiz and make personal feedback during the lab. If not, you'll have it back next Tuesday, and are encouraged to meet with me if something isn't clear.

Part I - Finishing Lab 8

Go back to Lab 8 and make sure you properly “enriched” `Rectangle.cs`. Are the headers correct, i.e., those studied in class? Make sure you implemented them correctly, and test them. Once you completed that, implement the `MultiplyRectangle` method we saw in class, whose header is `public void MultiplyRectangle(int factor)`.

Part II - Writing Your Own Class

In this exercise, you will create your own first class instead of using and expanding one that was written for you. The idea is to take inspiration from the class you already know (`Rectangle`) to create a new class, called `PreciseRectangle`, that will manipulate rectangles whose width and length are floating-point values, instead of integers (as in `Rectangle`).

Conception

Draw the UML diagram of this class: it should have two attributes, of type `double`, and five methods: two setters, two getters (i.e., one for each attribute), and one method to compute the area of a precise rectangle.

Implementation

To implement your method in VS, I explain two methods below: you can edit the pre-existing project, or start “fresh”. I recommend that you pick the one you feel the most comfortable with, and then try the other one.

Edit the Pre-Existing Project

- a. Re-download the Rectangle project, extract it in a folder, open it with VS.
- b. Re-name the project to “PreciseRectangle”, rename the “Rectangle.cs” file to “PreciseRectangle.cs”
- c. In the “PreciseRectangle.cs” file, replace `class Rectangle` with `class PreciseRectangle`.
- d. Comment out the body of the Main method in “Program.cs”.
- e. Your program should compile as it is, but you have to edit `PreciseRectangle.cs` to now store the width and the length with double, and to propagate this change accordingly. What should be the return type of `GetWidth`, for instance?
- f. Declare and manipulate rectangles with floating-point values for the width and the length in the Main method, and make sure they behave as expected (can you compute the area, for instance?).

Starting From Scratch

- a. Create a new project in VS, name it “PreciseRectangle”.
- b. In the Solution Explorer, right-click on “PreciseRectangle”, then on “Add...” and select “Class”. Then, select “Class”, write “PreciseRectangle.cs” as the name of the file, and click on “Add”.
- c. You are now suppose to have two .cs files opened and displayed in the Solution Explorer: “Program.cs” and “PreciseRectangle.cs”.
- d. Implement the `PreciseRectangle` class according to your UML diagram.
- e. Declare and manipulate rectangles with floating-point values for the width and the length in the Main method, and make sure they behave as expected (can you compute the area, for instance?).

Part III (Optional) - Pushing Further

The following are two independent tasks, to widen your understanding of this class, and to prepare you for the next labs.

- a. Class diagram (the one we will be using) are just a special case of UML diagram. Have a look at https://en.wikipedia.org/wiki/Unified_Modeling_Language#Diagrams: in which category are class diagram, behaviour, or structure diagram? Have a look at https://en.wikipedia.org/wiki/Activity_diagram and try to read the example of activity diagram for a guided brainstorming process.
- b. Now that you know more about naming convention, have a look at <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/naming-guidelines>, and particularly at
 - <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/general-naming-conventions>

- <https://docs.microsoft.com/en-us/dotnet/standard/design-guidelines/capitalization-conventions>