Project 1 – Classes and Top-down Design

Cameron J. Pastorius
COP3014
University of West Florida

September 16, 2019

Project 1

Aim

The aim of this project is to make a program that focuses on using classes. These classes will be able to help us create, modify, and access objects easily and efficiently. The actual computations being preformed by the program are fairly simple, its more about the neat and orderly implementation of the classes that this project is focused on.

Classes

This project had 3 classes total, a "Triangle", a "Circle", and a "Rectangle" class.

The Circle class had 4 private member variables (all doubles), Radius, Area, Diameter, and Circumference. The Triangle class also had 2 constructors and 5 member functions (4 accessors and 1 mutator).

The Triangle class had 3 private member variables (all doubles), Base, Height, and Area. This class also featured 2 constructors and 5 member functions but had 3 accessors and 2 mutators.

The Rectangle class had 6 private member variables (5 doubles 1 bool), Length, Width, Area, Perimeter , Diagonal, and the bool- Square. This class also had 2 constructors but had 8 member functions (6 accessors and 2 mutators).

Method

My approach to this project was to first open the shapetest.cpp and see how it was laid out, to see the goal that i should aiming for when i am writing my classes. This was a brief step and no real work was done, just gathering information. The next step was to start writing the header files for my classes, this project made it clear that there would be 3 classes, Triangle, Circle, and Rectangle. I quickly wrote the .h files for each class as the member variables and functions i needed where in the project description. I than began work on the class .cpp files, and this was the time consuming part, making sure each one worked in conjunction with main. Lots of testing went into this step, and i also built main while i was building these class cpp files to test if everything was working. After thorough testing of what was a complete program my last step was to make a makefile and do a final test on the ssh server to make sure everything worked.

Conclusion

Program works beautifully, with 3 different classes all neatly in their own files. This style of programming while more confusing at first, makes for a much neater final product. The final amount of classes i ended up with was the same as the project description said i would have, and the layout of these classes turned out to be the same as product description said. Overall the project went very smoothly.