CSC 340 Programming Assignment Rubric

The graders should be able to compile and run your program!

If the graders cannot compile and run your program, you will receive a maximum of 20 points on functionality, and a maximum of 20 points on style. These points are given at the grader's discretion.

Functionality - 50 points

The graders will test your program in a variety of ways. If your program passes all of their tests, you will receive full credit for functionality. The number of points docked per test will vary from assignment to assignment.

It is up to you to test your code! I expect you to try your program on different inputs before you submit it, to verify that it works correctly.

Your program will be tested on some inputs that are invalid. For example, if you are expected to ask the user for an integer, it is possible that I will enter a string instead of an integer when I run your program. Unless otherwise specified, you can handle this however you want, *as long as your program doesn't crash*. In general, your program should never crash.

Style - 50 points

You will be graded on a few different style categories:

Comments (10 points)

- Each file should have a comment at the top explaining the purpose of the file.
- Each method/function should have a comment at the top explaining its purpose.
- Inline comments should be used as appropriate - e.g. to explain a particularly dense patch of code.

Indentation, General Use of Whitespace (10 points)

- In general, make your code readable.
- Blocks within curly braces should be indented by one level. Generally, your code should line up.
- Spaces should be used as appropriate to separate variables from operators, parameters from each other, etc.

Variable and Function Naming (10 points)

- This one is pretty self-explanatory. Give variables and functions names that describe what they are. Don't make the names too long.

General Cleanliness of Implementation (10 points)

- Don't repeat yourself! In other words, don't write the same code twice. Find a way to unify common code.
- Don't allocate memory when you don't have to.
- Don't do something wildly inefficient (unless you are specifically instructed to).

Write-up (10 points)

- Describe your implementation. Some assignments may have specific questions that you are expected to answer. If there are no specific questions for an assignment, just provide a general description of how your program solves the problem given in the assignment.
- Explain how you compiled and ran your program.

Total - 100 points