Assignment 2 Reflection **John Waczak**

CS 161 Date: April 25, 2018

Understanding

The biggest challenge for me was trying to figure out a way to solve the first problem without using some kind of array or using a string like an array to hold all of the integers. I think that writing the pseudo-code actually made this easier as I wasn't stuck thinking about C++ syntax which enabled me to realize that I really only needed to check if the maximum or minimum should update each time the user enters a number. It was also valuable having the project plan as a sort of rough outline that way when it came time to actually write the code, all I had to do was figure out to translate my work into the C++ language.

Testing Plan

My testing plan was overall pretty good except for the first problem in the min-max problem. I had originally had the idea that setting both the min and max to zero in the beginning would be fine but when I ran my test case for a single number input, I realized that I had to set both the min and max to the value entered if this was the first time through the loop. Otherwise a user could enter a single number like -1 and get that the result was max = 0, min = -1 even though they've only entered a single number.

Design

As I mentioned in the previous example, my test cases helped indicate a flaw in my min-max design. This made me go back and add some added functionality for the case of running through the for loop for the first time. Beyond that my overall design was fine. For the file reading and writing problem, I wrote my pseudo-code before we had been presented the material in class. This led me to think I would have to be careful and cast the strings read from the file to ints. After figuring out the actual C++ syntax for file i/o I simplified the design since we use the ">>" and "<<" operators which do the casting for you based on what type of variable you are trying to write.

Implementation

To solve the problems I previously mentioned for the min-max and fileAdder problems I reread my code closely. Then for the case of the file stuff I checked the book which in fact had multiple examples for how to handle file reading/writing as well as EOF. I also googled around a bit and used stack overflow to get enough of the syntax figured out in order to be able to search for the write terms in the text.

Improvement

To improve the process I think it would be valuable to think of the Test cases first and then write the pseudo-code second. I did the reverse order and having done this now I think it would be more useful to decide how you want your code to perform before trying to determine the logic.