Assignment 3a Reflection

Byron Kooima

CS/161

January 25, 2017

Tim Alcon

Assignment 3a Reflection

The week 3 problem solving process was very useful in laying the ground work for the actual coding problem. As a software test engineer, I was very pleased to see an early module in this CS161 class focused on creating a software development plan. It is essential in any software program to have a good method of converting systems engineering requirements into functioning code.

I noticed very early in the process that I needed to start with an initial value from the user. The most useful part of breaking down the problem was focusing on how I would enter the correct number of integers from the user. It was a struggle at first to see how I could grab the first integer from the user and still contain that in a loop. I looked at the various examples of “for loops”, “while loops” and “do while” loops to come up with the best solution.

I spent a significant amount of time at the beginning using pseudo code to lay out the problem. This significantly reduced the number of errors in my test plan. However, the test plan did force me to think about the situation when a user enters a single integer and how my input loop would handle that condition. Initially, I had everything inside my “for loop” and my single integer input wouldn’t produce a min and max. After executing the entire suite of tests, I did recognize that I didn’t test the limits of integer inputs. However, we were directed that we could assume the user would use valid integers as inputs.

My design was fairly complete as well, but I noticed after coding the “for loop” for integer inputs, the “if statements” for capturing min and max could be improved. After reading some more details on when to use “if”, “if, else” and “if, elseif”, I modified my min max syntax. I knew that if a user entered an integer, it could be either smaller than the current min or larger

than the current max, but could never be both. There was no need for two completely separate “if statements”. Instead, I could use “if, elseif” statements for min and max.

I initially encountered issues with my loop and counter. I knew that min and max had to be initialized to the first integer passed in by the user. However, my “for loop” needed something to compare min and max to in order to determine if min and max had to be updated. When I started my “for loop” counter at 1, it would prompt the user for too many inputs since I already had initialized the values before starting the loop. I found the example from the “Odds and ends from Chapter 5” in the Supplementary Content useful for initializing my loop. I also used my design flowchart to break down the issue and when/how to initialize min/max and my loop counter.

The best part of this assignment was the real world application of breaking down requirements into usable code snippets. The example of painting a house was a great analogy of how to break down a software problem. The main improvement for my coding in the future stems from the use of pseudo code. I am typically a visual learner and flow charts present the big picture. However, in this assignment, when I started brain storming with pseudo code, I was able to progress very quickly. From our assignments, we can typically glean the information needed to get the why and what software needs to do, but the test plan and design are essential in working through the how.