

Emily Edmondson

[edmondem@oregonstate.edu](mailto:edmondem@oregonstate.edu)

Introduction to Computer Science 1

Project 3A Project Reflection

---

*I. What did you learn about the problem as you went? Why or how did you learn it?*

One of the bigger issues I encountered was realizing I needed to implement loops in order to account for all of the user's input without an obscene amount of variables. Furthermore, I don't know how to create new variables based on user-input that aren't already predefined in the program and so I knew that wasn't going to work. Thus, I needed a loop. Understanding I needed a loop was significantly different than actually implementing it. Implementing the loop required paying attention to what actually needed to be computed at each input as well as setting the boundaries to keep the loop finite so as to not have a program running continuously.

*II. What tests didn't work out the way you expected? What alterations did you have to make to your program due to failed tests? How could your planned tests have been more complete?*

In my pseudocode (and in my actual code once I initially wrote it) I had designed my loop thinking that my loop would only take the user-inputted integers and evaluate the input for min and max values. However, as I came to find out once I began to test my actual code, the cout statement in my code was nested within the loop, causing the program to output "Please enter X integers" however many times the user had requested integers to be input. This was annoying, and I had to alter my loop so that the cout statement was not nested within.

Additionally, in my pseudocode (and again in the first iteration of my actual code) I had set the "if" tests to check whether the input value was less than or equal to the minValue or greater than or equal to the maxValue. This works with inputs like [-3, 4, 5] where the output would be min: -3 and max: 5, but not for inputs like [7, 7, 7] where the output should be both min: 7 and max: 7. This was also a problem with my variable initialization. I had set both minValue = 0 and maxValue = 0.

This obviously had to be changed in the final version of the code, where I set both min**Value** and max**Value** = to the original user input and began my loop after that initial input. My tests could have covered that better.

*III. What problems did you encounter during implementation? How were you able to solve those problems? What outside sources (sites, books, or other materials) did you find helpful?*

During implementation a lot of the errors described in the second paragraph of the previous section were encountered. Additionally, I had trouble implementing a nested loop within my code, mostly due to improperly placed "{" and "}". This was easily resolved by consulting with the book. During most (albeit short) of my coursework for this class I have consulted the book, consulting only once [cplusplus.com](http://cplusplus.com) when I had trouble understanding a do...while... loop which, incidentally, is not implemented in Project 3a.

*IV. How can you generalize any parts of your problem solving experience in a way that might help you on future assignments?*

I think the use of pseudocode is especially useful for me. I initially tried to complete this project by using a flowchart and just kept running into mental blocks until I tried to implement pseudocode. Pseudocode gave me the flexibility to express my thoughts on how the program should be structured and developed without being bogged down at the time with the syntactic details necessary to ensure no compiling errors. Speaking generally, the use of pseudocode can only be beneficial for me in the future.

Additionally, based on my own experience in this project and some of the problem-solving steps given in the book (especially things like steps to developing programs, and the layouts of what happens in if statements in ch. 4, and what happens in loops in ch. 5) I think it will be important to layout in pseudocode exactly what data needs to be tested by code statements, what data needs to be iterated through and to what end, and what code iterations and code tests (by tests I mostly mean "if" statements) will actually compute. Having that laid out clearly beforehand can help prevent a lot of confusion.