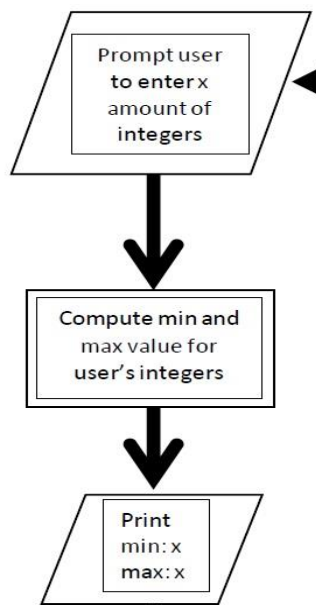


Understanding: What did you learn about the problem as you went? Why or how did you learn it?

As I went through the assignment, I learned that the problem of coding the assignment was a lot bigger than I imagined it would be during the planning phase. Even though this was a relatively small program, to me, it felt like a pretty large and complex program to design.

During the planning phase, I wanted to break down the problem as much as possible, in hopes that it would make coding much easier. However, in retrospect, I found that this caused me to narrow my focus a little *too* much. As a result, I wasn't thinking big enough and this made things a lot harder during the coding phase.

One example I can think of is that while I was designing my flow chart, during this part of the flow chart, I had not even considered *how* the program would actually compute the min and max values for the user's integers. Now that the program is completed, I know that one way to have the program do so is through the use of Relational Operators. Though I am not certain of how this affected my grade for this assignment, or how big of a problem it would be in a workplace scenario, it is my opinion that this stands as a concrete example of how simplifying the problem too much (or too quickly) caused me to narrow my focus and forget a critical part of the process: this program has to know how to compare the integers to one another in order to determine which is the smallest and which is the largest.



If I were to redesign this now, I would include steps that demonstrate the following:

- Firstly, the program must accept the first integer and set that integer as both the min and max number.
- If the user only wants to input 1 integer, that integer should be displayed as both the min and max, and the program should end.
- If the user wants to enter any additional integers, then the program should allow them to do so, comparing each of those integers to the current min and max, and setting the new values as necessary.
- Once the user is finished entering numbers, the final calculations are done and displayed before the program is terminated.

Had I understood this prior to starting assignment 3a, I have no doubt that I would have had a much easier time completing the assignment.

Testing plan: 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?

Negative integers is the first thing that comes to mind. The first time I got my program to compile with it actually asking the user for integers, accepting those integers, and then start comparing them to one another, I quickly noticed that if I entered any negative integers, they simply did not compute. My program would not display any numbers less than 0. This puzzled me for the longest time. Eventually, I was able to determine that I was assigning the min and max values the number 0 and that was causing numbers to never be read as less than zero. Therefore, no number below 0 would ever be considered. I probably would not have had this issue if I had properly considered how the program would handle the user's input in the design phase.

Design: What was missing or needed to be altered from your initial design, and why?

I touched on this a bit earlier and I go in to a little more detail about this later on in this reflection, but overall my design process was delayed and much harder than it needed to be. Ultimately, I think this was because I spent so much time worrying about *how* this program would actually work (with code I hadn't become familiar with until I actually read through the entirety of the required reading), rather than breaking down the larger problem into smaller steps, without worrying about the code I'd need. I looked at the assignments before I started the reading and this was probably one of my biggest pitfalls during this assignment.

I also noticed that a few of my classmates shared some of my concerns towards the assumptions about the user. Although it has been clearly stated that we should assume the user won't try to enter anything that will violate the specifications. I guess I thought this was counterproductive to testing the programs I write. When I am testing my programs or thinking about how they'll be tested, I am usually worrying about trying to think of ways the user will break the program, when instead I should be making sure that legal inputs are functioning the way they should.

Implementation: 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?

I would have to say that I spent the most time trying to properly loop in the program. Initially, I had trouble trying to decide which loop to use. I ended up writing two programs, one that used the while loop and another that used the do loop. I kept making changes to each of them hoping that I would eventually learn something in the process that would actually help me complete either of them for submission. In the end, I used the while loop with a counter. The last unsuccessful build I was working with only had a problem with the counter.

```
int min;
int max; // Maximum number inputted
int counter = 1; // Counter

cout << "How many integers would you like to enter?" << endl;
cin >> numIntegers;
```

Originally, I had the counter set to 1 and this was causing an issue where the program was not asking for the correct number of integers. For example, if the user wanted to enter 4 integers, the program would ask for 5. I corrected this by changing "int counter = 1;" to "int counter = 2;"

The textbook was ultimately the best source of information for any troubles I had, but a close second would be discussions on Piazza and going back and forth with classmates on Slack.

One classmate in particular was particularly skilled at asking me questions in a way that got me to understand exactly what problem I was having in my program and how to fix it. I know not what sorcery she weaved into those questions that changed the way I was thinking and resulted in my brain knowing exactly what I needed to do to correct my mistakes, but I am indeed indebted to her. This is officially a shoutout to her.

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

Finish the assigned reading first, before attempting any part of the activity, even the planning phase. This may have been a result of feeling pressured to finish before the deadline, but initially, I thought I didn't need to read chapters 4 and 5 before starting the planning assignment. In fact, I recall a discussion with a fellow classmate on Slack, just a few days ago:



Oct 9th at 1:04 AM
in #usw161

Can you guys let me know for project plan 3a do i have to know how to program in order to design plan? Since i was not able to start my assignment on time i wanted to know if I can just design and make plan or it's necessary that i need to read chapter 4 and 5, know how to program 3a and then know how to make plan. I am little confused about setting up plan so.. (since i couldn't start study early this time i am asking very basic question 😞)



Tristan Santiago 4 days ago

I felt the same way before I started working on the design and forced myself to read through both chapters before finally getting to work on the plan itself. However, I think that it wasn't actually necessary. It's just my opinion, of course, and others might disagree. In retrospect, I think I was asking myself the wrong questions prior to starting. I was initially really worried about having no idea how to actually code the program before starting the design and that actually made me put off starting on the design. Looking back now, I feel like if I had just read the instructions for assignment 3.a first and took more time to understand what the assignment was actually asking to do, I probably still could've designed the plan without reading the chapters first.



Tristan Santiago 4 days ago

I won't say that reading the chapters first didn't help me design the plan (because it definitely did, as I was really stuck on how to the program would actually work before being introduced to loops), but I think that understanding exactly what the program needs to do is a little more important than knowing exactly what kind of code needs to be used to actually make it work.

My thought process was all over the place during this assignment. I went from thinking I did not need to read the chapters, to deciding that I probably should force myself to, and back to "maybe I didn't need to after all." I probably could have avoided all of this confusion if I had either:

A) Been more decisive in the beginning, or
B) if I just did the assigned reading before even looking at the assignment.

All of the back and forth really just made things worse for me. I constantly felt the pressure of the deadline and would make the mistake of thinking a single piece of information from the text was all I needed to complete the assignment, when in actuality, it was only a small part of a much bigger whole. That whole would only come together once I finally understood what the program needed to do and what I needed to use to make the program perform those required tasks.