

Project #: 2

Semester: Spring 2022

Name: Christopher LaFave

I. Requirements: Restate the problem specification and any detailed requirements in your own words.

Declare and create a working constructor, multiplication operator, and input operator for the ComplexNumber class.

Only certain initialization cases were required to be taken into account.

II. Design: How did you attack the problem? What choices did you make in your design, and why? Show class diagrams for more complex designs.

The multiplication one was the easiest by far. I just followed what wolfram said the algebra was and plugged it in.

The constructor was a little harder. I used strtod to find both the numbers in the string and assigned them to r and i respectively. This worked for both a complete complex number and just r. But I had other cases to check. I next implemented code to detect if the complex number had only an i part. After getting r, the program checks if there is an i after it. If there is, it gives i the number just put in r and turns r back to 0. This made the case of only an i part work. But my code had one more problem. It didn't work for either i or -i. So, I wrote cases into the no-r checker to see if it was either of those cases, and to map i accordingly.

The last section was the input operator. After realizing that I don't have to account for cases like "1 2", it was easy to implement. I just grab the input and give it to my constructor.

III. Security Analysis: State the potential security vulnerabilities of your design. How could these vulnerabilities be exploited by an adversary? What would be the impact if the vulnerability was exploited?

Currently, this program does not check for inputs that do not fit any constructors. There can probably be some memory manipulation done there.

IV. Implementation: Outline any interesting implementation details in your solution. Using strtod was very fun and learning the ins and outs of it was very fun and will be very useful for my future.

V. Testing: Explain how you tested your program, enumerating the tests if possible. Explain why your test set was sufficient to believe that the software is working properly, i.e., what were the range of errors for which you were testing.

I didn't do any testing outside of the regular zybook tests, but because of the nature of this assignment I don't think I need to.

VI. Summary/Conclusion: Present your results. Did it work properly? Are there any limitations? NOTE: If it is an analysis-type project, this section may be significantly longer than for a simple implementation-type project.

My project does work properly! The only limitations are ones allowed by the instructions.

VII. Lessons Learned: List any lessons learned. For example, what might you have done differently if you were going to solve this problem again?

It really helped when I got to class and was able to ask a friend for help. I need to stop being so self-sufficient and ask others for help more often.

Also, I am getting continually more comfortable with C-strings.