Project #1
CS 2210 – Fall 2021
Sam DeCook

I. Requirements: Restate the problem specification, and any detailed requirements

Make a GUI that uses the Factorial and GCD classes made in Project 1. Have each pop up a new dialog with proper buttons. Include error handling for negative numbers (factorial) and also for non-integer input.

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.

I made the GUI and then worked out how to get text to and from the text boxes. After that I worked on error handling

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 is exploited?

I don't think there are any potential security vulnerabilities.

IV. Implementation: Outline any interesting implementation details.

There isn't much of interest, this was more of a "getting started" project.

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 possibilities of errors that you were testing for.

I tested negative numbers for factorial and string inputs for both and the correct error message popped up.

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

The program is working properly.

VII. Lessons Learned: List any lessons learned. What might you have done differently if you were going to attack this again.

Make sure to have your other classes in the same package. That was the only way I got the code to work.