Project #1
CS 2210 – Fall 2021
Sam DeCook

I. Requirements: Create two classes that can compute the factorial and greatest common denominator (GCD). Do all error checking in the classes (so they could theoretically become library code). For the factorial, return a double and do not multiply by one. For the GCD, compute it recursively, using the definition GCD(x, y) = GCD(y, x mod y).

II. Design: First, I wrote out the code to perform the operations. Then, I wrote the error checking, and finally, I designed the main class to take in and parse input and then print out the output.

III. Security Analysis: I don't think there are many vulnerabilities. We are only manipulating variables that the program itself creates, not working with anything that could affect other areas of the computer.

IV. Implementation: The recursive portion of the GCD code was very similar to the code you write to switch two variables. When reassigning y, you set it to a (which is x) mod y. With factorial, you can set the output to 1, and then start the loop at i = 2, which makes the for loop more efficient.

V. Testing: For Factorial, I tested the common numbers going up and found it return 0 after 33. For GCD I went through some test cases and everything checked out.

VI. Summary/Conclusion: Everything worked properly, except for the huge factorial computations. However, I don't think that takes away from this program.

VII. Lessons Learned: I forgot to add a package inclusion at the top of my files and I forgot that you need to have `Integer.` before `parseInt`. However, these are my first few weeks back in Java, so some rustiness is to be expected.

My code compiled and ran properly and produced the expected output.