

CSC 313 Final Project Report

Your Name

Your Team

Month Day, Year

Project GitHub Repo

Please include a link to your team project GitHub repo here. If the repository isn't public, make sure your instructor has read access to it.

Project Summary

I found the lifecycle of this project to be far more fulfilling than any of the other projects I've had heretofore. It was so much more complete in my mind. Our other projects, even when they were accomplished in groups or with collaboration, still felt detached and rather trivial. The completion of which was usually more of a relief so I would have time to work on other things. While I can't say that I'm sad to see the completion of this project, or at least its conclusion, I feel like we have actually made something of a mark. Future classes that come through, if they should happen to receive the same projects, have something of a bar to compare to. While I'm rather certain that these programs will soon be relegated to dusty digital archives, for the time being I feel quite pleased with what we have accomplished.

The initial design phase was a little rocky but after we really got into the project and the existing code, we understood what needed to be done, and prepared a pretty good design.

The programming phase was a little challenging for the team. We had somewhat differing views on how the project should proceed about halfway through the weeks of coding, namely on specific features to drop that ultimately were cut out of the program due to difficulties in implementation. Something of a result of one portion being allowed to continue instead of altering midway through, is that the GUI class seems rather monolithic due to lack of time to partition it better.

Even with some setbacks, I am still quite pleased with the project. In retrospect, I appreciate the experiences this provided, both in working in teams, and in realizing the benefits of good object oriented code (even though it's possible we didn't fully implement all the best design principles we could have).

Time Breakdown

Week	Total Hours (Rounded)
June 1-7	4
June 8-14	12
June 15-21	7
June 22-28	11
June 29 - July 5	15
July 6-12	21
July 13-19	15
Total	85

Self-Evaluation of the Project

Working Code

Although you may not be able to find runtime exceptions, this is because we edited our program close to the deadline in order to remove some features that were insufficiently robust in their implementation. Does it work though? Well, it certainly runs, and I think it runs quite well at that. In terms of test cases, we did put a few of them in, but definitely could have been better with them.

Requirements

There was one explicit requirement not met, namely the requirement to be able to set initial pressure. However, I believe that that is the only project requirement not met, and we actually completed the stretch requirements.

Design Principles

We constantly focused on making our code object-oriented. As explained in the previous section, our end result was on track to become what we modeled. At the time of submission, the GUI class was rather large and unwieldy. Some of the other classes though are much more object oriented. A specific example of an OO change made was in the CSVWriter class. Initially, both functions in the class created their file writer independently. Seeing the duplication of code, we changed it so that they simply called another function to create a valid file writer for them. As it stands, the code is modular, and we followed the principles we discussed in class, as far as we can tell.

Code Style

I believe that we tried to maintain the java coding standards and had good success in doing so. I didn't find the code to be painful to look through.

Documentation

We have javadoc headers for most of our methods, and have some comments in a few key parts, but could have been more diligent in our documentation. We started out pretty good, but got a little less committed as time went along.

Version Control

I maintained the habit of committing my changes to GitHub frequently so as to reduce the amount of repeated work by myself and by my team. I also tried to not be verbose in my log entries and to just simply state what was being done in that change.

Project Evaluation

Criterion	Performance Level
How well does the project work?	Good (90%)
How well does the project fulfill its requirements?	Between Good and Mostly Adequate (85%)
How well does the code embody good design principles and patterns?	Mostly Adequate (80%)
How beautiful is the code?	Exceptional (100%)
How well documented is the code?	Mostly Adequate (80%)
How well was version control used?	Exception (100%)

Grade Calculation

According to the numbers above and their weights given in the rubric, the total grade comes to 88.75%. While we may not have had ideal procedures in coding or a perfect product, we worked hard, and made a neat program. We all agree that we performed rather well, and I think we are deserving of this grade.

Team Member Contribution

Name of Teammate	Participation (0-3)	Responsibility (0-3)	Value Added (0-4)	Total Points (0-10)
Myself	3	3	3	9
Teammate 1	3	3	4	10
Teammate 2	3	3	4	10

I was very pleased with the quality of my teammates. While there was some initial misunderstandings during design and studying for our presentation, once we got through them it was very pleasant. Most of the misunderstandings were merely saying the same thing in different ways. Teammate 1 put in a lot of work on the GUI elements, and Teammate 2 worked hard on the graph and table. I did more work in the service package, bringing the classes all together, and providing debugging and consulting for my teammates. They worked hard on the project, and were excellent teammates. There was a stretch for about week, where I was really busy with a project for another class, so I think I could have pulled my weight a little more, but otherwise, I think we did great!