CSC212 Project 1 Report

Joseph Yanez, Matthew Boekamp, and Sahil Chadha

The task for this project was to create a gradebook for the CSC212 course. The gradebook must contain functions to return specific grades of specific assignments, total category grades, total overall grades, both weighted and current, and allow amendments to the ".txt" file which the grades will be entered from.

Before starting any programming, the group entered a "pregame" phase, where we discussed how exactly to tackle the prompt. Most of this planning was conducted through Zoom meetings, where we not only planned with pseudocode, but shared screens to properly convey our idea before committing and final alterations. The pseudocode consisted mostly of our initial ideas, where we laid out the skeleton of our program, including the header file, full of variables and methods, as well as general ideas for said methods. Instead of writing out the pseudocode separately, we directly edited the pseudocode and filled in the program directly, eliminating unnecessary confusion and making sure that the group was on the same page.

The first methods written were those which grabbed the grades from the .txt file, as well as the methods that calculated the grades. This way, any future changes can be made directly with little to no effect on these methods, as the calculations would remain largely unchanged. To grab and hold onto the information from the text file, we used Structs to store the information by category, by creating a loop and changing the category when different words were recognized. For example, when the program saw the word "Labs" in the .txt file, a "Labs" category was created where the lab grades were stored until the word "Assignments" was seen, where a new assignments category was created to store the assignment grades. This process would repeat until all grades had been stored into their respective categories, and ready for manipulation from the user.

Some of the other methods consisted of actually outputting the desired information. Through the use of vectors and pointers, the program can successfully locate and output any individual grade of any category, by searching through the inputted category for the specific assignment, which is taken in as a common line argument, and outputting it onto the console. This same idea is seen through the total grade methods, where instead of searching for individual grades, these methods instead go to the respective categories, add up the grades of that category, and return them for the user to see. The same can be said for the total overall grade, where all grades from the text file are summed, and may either be outputted as a grade or weighed by dividing by the total number of current points, depending on user input.

The last methods left to discuss are those which make direct alterations to the text file. So far, all of our methods have only taken from the file, and output desired information based on user input from the command line. These methods, however, write

directly onto the text file, meaning that any grades that may have been inputted incorrectly can be amended to provide accurate grading information. For example, if the text file had an incorrect lab grade, the main function would first ask the user for the name of the category, the name of the assignment that they want to change, and then the new grade, and use all of these as inputs to call the changeGrade method. This method first loops through the locates the grade that the user wants to change inside the vector, sets the grade to the new grade, and then writes it to the text file after the program has ended, to ensure that no information is lost or overwritten when using the change methods.

Although only lightly mentioned before, the main function is the control center of the program. It takes in all of the information from the text file and calls the correct methods to ensure that only the desired data is being outputted. This is mainly done through a Menu system, where the user is asked to select from a number of options, depending on what they want the program to do. The menu is used solely through command line input, and is equipped with an exit feature, error messages for improper inputs, and the correct function calls for the desired data. Furthermore, some of the menu options need additional options, such as looking for individual assignments, which also take in command line input to call different methods. All of the inputs for the menu systems are whole integers, with each number representing a different option.

Overall, the program has all of the necessary gradebook capabilities, including the ability to fetch specific grades, calculate totals based on category, weight, and total class grade, and can change grades and write them back into the text file. The program could be altered to create a more universal gradebook by asking the user for categories, number of assignments per category, and the grade weights for the assignments in the future, however, for a CSC212 gradebook this will more than suffice, and accurately return all grade values that the user will ever deem necessary.