Daniel Gonzalez

COMPSCI -2

Professor Onwuka

9 May 2025

Gradebook Management System

For my final project for this class, I created a student gradebook management system using C++. The goal was to create a program that allows users to manage and store students' records, such as their names, ID numbers, and a list of their grades. The reason I decided to make a gradebook was that when I started to go back to school, I wanted to get good grades. My main goal for my first semester was to get all As, which I did. This project has three requirements for the final project, which include classes, file I/O, and pointers.

Key C++ Concepts Used:

1. Classes and Encapsulation: The student class encapsulates the data (full name, student ID, and grades) using methods like addGrade(), getAverage(), and toFileFormat().
2. File I/O Streams: I used ifstream and ofstream to save and load student data in a readable and portable format.
3. Vectors: Vector<float> was used to store multiple grades per student, allowing the user to add many grades each student has.
4. Error Handling: The program includes input validation to ensure the user enters a valid value.

Challenges Encountered:

One challenge I encountered was handling invalid user input. For example, when entering a letter instead of a numeric value when entering the grade caused the program to loop and break. The

way I resolved this was by using cin.fail(), which checks that the input is valid and ignores letter inputs. Another challenge was structuring the project using multiple files. I couldn't figure out how to use headers properly in my main() function. So I put it into one .cpp file. Going to work on some projects on my own to figure out how to use headers in my next project.

Conclusion/Future Work:

The project was fun to make on my own. I wanted to do this on my own because I wanted to see how much knowledge and creativity when it comes to programming. Learning how to use a repository system like GitHub was fun, and now I understand how it can be used in group programming projects. In the future, I would like to split the file into multiple files, like headers, to follow C++ best practices and design, add the ability to edit or delete existing student data without deleting the whole student, implement sorting, and searching algorithms to organize data and use more advanced data structures like maps or linked lists for faster access and flexibility. This project can easily be expanded into a full course management system for professors or schools.