

Project 1 Report

Garrett Nix

1-23-2026

1 Introduction

While calculating a single grade by hand may seem trivial to compute, in scale this often becomes very tedious and time consuming. Then there is a need for having a method suitable for creating an efficient method for computation. This project aims to amend this issue by providing a potential solution.

2 Purpose

As stated, this project aims to more effectively scale the grading process for the course CSCI271 by creating a program that does this in an efficient and robust manner. This allows for many data points to be processed in rapidity, and makes the workload of grading less cumbersome at the end of the semester.

3 Implementation

Let A be the average score of assignments. Also let E be a combined scoring of all test averages T , midterm score M , and final exam score F divided by the scaling factor 70%. That is

$$E = \frac{0.4F + 0.2M + 0.1T}{70}.$$

Using E , calculate the final grade G using the piecewise function

$$G = \begin{cases} E & E < 60 \\ (1 - W)E + WA & 60 \leq E < 80 \\ 0.4F + 0.2M + 0.1T + 0.3A & E \geq 80 \end{cases}$$

Where $W = 0.3(E - 60)/20$. Given a CSV file as a command-line argument, the goal is to have the file (that a professor has been updating as the semester has progressed) passed into the program, and efficiently output the name of the student, with their final numeric grade along with it.

4 Discussion

Most of the pitfalls encountered during this are knowledge-based. In specific, attempting to make the program robust, it was desired to create a function that took a header of column names from the csv and sort the columns to match the example CSV provided. This would involve what is presumed to be some form of linked list, as well as a sorting algorithm. One of these concepts are known, but given the time constraints of the project, this is a pitfall that I would like to amend later in the course once sorting algorithms have been discussed.