Software Engineering Assignment SDD

Trenton MacNinch, Michael Lin, and Ying Gao

2024/11/22

Contents

1 Introduction

1.1 Date of Issue

2024/11/22

1.2 Authorship

Trenton MacNinch, Michael Lin, and Ying Gao

1.3 Summary

The program is used with the following syntax:

```
calcgrades [name file] [course file] [output file]
```

The application reads two input text files and outputs one file with specific formatting. The input files are given by the user, formatted as the following:

```
[name file]
Student ID, Student Name
100000001, John Hay
100000002. Mary Smith
…

[course file]
Student ID, Course Code, Test 1, Test 2, Test 3, Final exam
100000001, CP317, 75.3, 80.4, 60.3, 70.5
100000002, CP414, 80.2, 90.5, 50.4, 75.6
100000001, CP414, 60.5, 70.6, 80.6, 75.6
…
```

The output file is formatted as the following, with special notice given to student IDs being in ascending order:

```
[output file]
Student ID, Student Name, Course Code, Final grade
1000000001, John Hay, CP317, 66.7
1000000001, John Hay, CP414, 72.6
1000000002, Mary Smith, CP414, 74.8
…
```

2 Software Architecture

2.1 Overview

Given the requirements for our application (Object oriented features, compiled language, file handling, etc.), we have chosen to create a C++ program running on a Linux system. The overall system architecture then includes the following;

- 2 Input Files
- 1 Executable
- 1 Output File

3 Detailed Design

3.1 Overview

The application loads all student information into a map, then loads all course information into a vector of custom course objects, and then sorts the vector of course objects. The program will then proceed to iterate through all stored course information and attach student names based on IDs while placing it in the designated output file. If the program encounters an error, it will print the encountered error.

3.2 Student Information

The program opens the name file as a normal C++ file handle. The program then calls a function that generates a map. This function reads the info from a line in the name file, validates the given information, and adds the student info to the map if the information is acceptable.

3.3 Course Information

The program opens the course file as a file handle, and uses a function to generate a vector of course objects. The function reads the information from a line in the course file, validates the given course information, and rejects any invalid courses. The program then sorts the vector of course information by student ID to enable a more efficient outputting phase.

3.4 Creating Output

The program takes the information stored in the student id map and the course object vector, and combines it according to the output format while pushing it to the file handle for the output file. The program then closes.

3.5 Error Handling

Given all errors are likely to be from erroneous inputs (invalid id, course code, grades), the program validates all inputs to ensure only valid inputs are taken. The program will continue to run after printing an error message to console indicating what the error was. Due to the possibility of bonus marks/grades, only negative grades are detected as invalid. Additionally, all student names are treated as valid.