

Lab Report 02

Problem

Calculate the final course grade from the grades listed in a file.

Proposed Solution

Use a buffer (for efficiency) to read lines from a file. Store percentage in a floating point number (Percentage here is percentage points of FINAL grade, not mean score for an assignment). If there is (or could be) more than one than one grade for the category, create a state variable and add the current percentage to the state variable (allowing one pass through the file instead of two). Add the percentages.

Tests and Results

Calculating grades for file perfectGrades.txt...

Your EXAM_1 average is 100.

Your EXAM_2 average is 100.

Your LAB_EXAM_1 average is 100.

Your LAB_EXAM_2 average is 100.

Your FINAL average is 100.

Your homework average is 100.00.

Your lab average is 100.00.

Your extra credit total is 10.00.

Your raw grade is 110.0%.

A.

Calculating grades for file grades.txt...

Your EXAM_1 average is 65.

Joshua Nelson
CSCE 146

Your EXAM_2 average is 65.

Your LAB_EXAM_1 average is 100.

Your LAB_EXAM_2 average is 100.

Your FINAL average is 80.

Your homework average is 90.00.

Your lab average is 92.50.

Your extra credit total is 4.00.

Your raw grade is 90.5%.

A.

Enter a file to read, or 'quit' to quit.

nonexistent.txt

Calculating grades for file nonexistent.txt...

Could not open file `nonexistent.txt' for reading. Please ensure it exists and you have the proper permissions.

Enter a file to read, or 'quit' to quit.

quit

Problems Encountered

Very many problems.

- Firstly, the number of homeworks and labs is unknown, meaning that in order to store them in a static array, the file must be read two times. I used state variables to work around this.
- Secondly, there are very many conditions for grading; I found dropping the lowest grade the most difficult to express logically. I made the enum class CATEGORY instantiable to avoid magic numbers for grade percentages.

- Thirdly, there are a variable number of extra credit problems, which *do not* have a fixed percentage of grade; these must be evaluated solely as a percentage points of the final grade for the class.
- Fourthly, although the percentages *are* integer percentages, they give errors when evaluated as integers instead of floats (e.g. $(\text{homeworkSum} - \text{minHomework}) / (\text{numHomeworks} - 1)$ gives wrong results when homeworkSum is an integer). I had to redo much of the design after realizing this.

Minor problems:

- Java does not have a convenient way to express 'delimiter'.join(string); this was very frustrating as you can see in my JavaDocs for makeValidCategories.
- This project was extremely stateful, much more than previous projects, and I had trouble finding classes to design. As a result the *entire* project is done from static methods, with only final static data members used from the class. This is certainly not the ideal way to do the project, but I'm not sure what I could have done instead.
- Finally, my IDE decided it would be a good idea to create a `src` package to place my java file in, which meant that `javac file.java && java file` failed miserably.

Conclusions and Discussion

This was a difficult lab, due not to reading from a file, but the number of variables and state to keep track of. I was proud of researching how to instantiate enums in Java, which removed magic numbers almost altogether. I could have improved by creating class variables and delegating to methods; however, this would have required a two-pass system because the number of homework assignments was unknown.

Additional Questions

1. When you open a stream for both reading and writing files, what must you always remember to do at the end?

Close the file descriptor to avoid memory leaks.

2. Define file format or protocol.

Joshua Nelson
CSCE 146

A file format is a standardized way of storing information. Standardized formats allow different programs with wildly different internal designs to read the file and interpret it in the same way. Common formats include HTML, XML, Office Open XML, and CSV/TSV (Comma/Tab seperated values). In this lab, the format was

...

Category

Data

Data

...

Category

Data

...

...