

CS 12 Grades Calculator

The Task

You will design a set of classes for storing student information along with a main program that will read student information from a file, store the data, compute final grades, and then print a summary report to an output file.

Program Details

1. Design a set of classes that store student grade information. There should be one super class `Student` to store common data and three child classes that divide the set of students into three categories: English students, History students, and Math students. All data stored in these classes should be private. Any access to class data from outside should be done through accessor methods. The parent class should allocate storage for the following data:

- student's first name (assume 20 characters or less)
- student's last name (assume 20 characters or less)
- Which course the student is in (English, History, or Math) - *hint: get the constructor of the child class to assign this value*

2. The parent class should have a method that computes the final grade. Each child class should override this method and return the student's final average based on the stored grades. All grades are based on a 100 point scale. Here are the grades that need storing for each subject along with the breakdown for computing the final grade:

- English
 - Term Paper = 25%
 - Midterm = 35%
 - Final Exam = 40%
- History
 - Attendance = 10%
 - Project = 30%
 - Midterm = 30%
 - Final Exam = 30%
- Math
 - Quiz Average = 15%
There are a total of 5 quizzes averaged together (result can be a decimal number)
 - Test 1 = 25%
 - Test 2 = 25%
 - Final Exam = 35%

3. Write a main program in a separate java file that does the following (in order):

- Read the data from in input file called `input.txt`. Write to an output file called `output.txt`.
- Read the student data from the input file and store it using an array (or `ArrayList`) of appropriate type. You should use just one array for all students, and not a separate array for each subject. You will need to allocate this list dynamically (after the program begins) since the size is stored in the input file. Each student's data should be stored in a separate object.
- Print a summary report to the output file as specified below. You'll need to use the method that computes the final average when you do this since the final averages will be included in this summary report.

File Formats

Input File

The first line of the input file will contain the number of students listed in the file. This will tell you how big an array you need. After the first line, every set of two lines constitutes a student entry. The first line of a student entry is the name in the format `lastName, firstName`. Note that a name could include spaces; that the comma will separate last name from first name. The second line will contain the subject (“English”, “History”, or “Math”) followed by a space, then a list of grades (all integers) all separated by spaces. The order of the grades for each class type is as follows:

English Term paper Midterm Final Exam
History Attendance Project Midterm Final Exam
Math Quiz 1 Quiz 2 Quiz 3 Quiz 4 Quiz 5 Test 1 Test 2 Final Exam

Output File

The output file that you print should list each student’s name (`firstName lastName` – no extra punctuation between), Final Exam grade, final average (printed to 2 decimal places), and letter grade based on a 10 point scale (see below). Output should be separated by subject, with an appropriate heading before each section, and each student’s information listed on a separate line in an organized fashion. Data must line up appropriately in columns when multiple lines are printed in the file. At the bottom of the output file, print a grade distribution (i.e. how many As, Bs, Cs, etc) of all students.

10-point scale

A: 90-100

B: 80-89

C: 70-79

D: 60-69

F: 0-59

Within each subject in the output file, list the students in alphabetic order sorted by name. This sort needs to be true alphabetical and not just [lexicographical sort](#). To do this, make the `Student` class sortable by subject, then last name, then first name.

Sample Input File

```
3
Smith, Suzy
English 60 89 75
Heng-Suz, George
History 99 87 80 86
Sol, Fred
Math 87 76 45 76 34 78 89 77
```

Sample Output File (for sample display only, grades not accurately calculated)

***** English *****

Name	Final Exam	Final Average	Letter Grade
Suzy Smith	75%	76.15%	C

***** History *****

Name	Final Exam	Final Average	Letter Grade
George Heng-Suz	86%	85.8%	B

***** Math *****

Name	Final Exam	Final Average	Letter Grade
Fred Sol	77%	78.24%	C

***** Grade Distribution *****

A - 0 B - 1 C - 2 D - 0 F - 0