

StudentGrades Assignment (CA)

Due: Friday, December 30, 2022 at 11:59 PM

Objective

This assignment will provide practical experience on OOP .

Score:

30%

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 base class to store common data and three derived 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 or protected. Any access to class data from outside should be done through public member functions. The base class should allocate storage for the following data (and only this 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)

2. Each class should have a function that will compute 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 file) that does the following (in order):

- Ask the user for input and output file names. This is the only input and output that should be done from the keyboard and to the screen. All other input and output will be to and from files.
- Read the student data from the input file and store it using an array of appropriate type. You should use just one array (i.e. a heterogeneous list) for all students, and not a separate array for each subject. You will need to allocate this list dynamically since the size is stored in the input file. Since the list is dynamic, the list items will also need to be

created dynamically. Each student's data should be stored in a separate object, and dynamically allocated space should be cleaned up appropriately with delete when you are finished with it.

- Print a summary report to the output file as specified below. You'll need to use the function that computes the final average when you do this since the final averages will be included in this summary report.

File Formats

1. Input File (see attached file)

The first line of the input file will contain the number of students listed in the file. This will test you how big a list you need. After the first lines, 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 delineate last name from first name. The second line will contain the subject ("English", "History", or "Math") followed by 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

2. Output File (see attached 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 (i.e. 90-100 A, 80-89 B, etc). 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.

Submitting

ZIP your source files and README files into a simple zip file. Turn-in on Google classroom assignment. Make sure the submitted files are named as *yourname_regnum*, and that you do not include any extra files (e.g. object files or executables).