HW #6 (Student Grades)

- The objective of this assignment is to gain experience with base and derived classes, **virtual functions**, and using applications of **polymorphism**. Also, to gain further practice with **file I/O**.
- You are expect to do the work in C++ (or Java if you prefer).
- 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.

HW #6 (2)

- 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):

1. Student’s first name (you may assume 20 characters or less).

HW #6 (3)

2. Student's last name (you may assume 20 characters or less).
3. Which course the student is in (English, History, or Math).

- 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 each final grade:
- **English** – Attendance = 10%, Project = 30%, Midterm = 30%, Final Exam = 30%.

HW #6 (4)

- **History** – Term Paper = 25%, Midterm = 35%, Final Exam = 40%.
- **Math** – There are 5 quizzes, to be averaged into one Quiz Average (which can be a decimal number). Final grade is computed as follows:

  Quiz Average = 15%, Test 1 = 25%, Test 2 = 25%, Final Exam = 35%

- Write a main program (in a separate file) that does the following (in this order):

HW #6 (5)

a) Ask the user for input and output file names. This is the only input and output that should be done from keyboard and to the screen. All other input and output will be to and from files.

b) Read the student data from the input file and store it using an array of appropriate type. You should use just one array for all students, not a separate array for each subject (i.e., this will be a heterogeneous list). You will **need to allocate this list dynamically**, since the size is stored in the input file. (Note that this also means each list item will need to be created dynamically).

HW #6 (6)

Each student's data should be stored in a separate object. (Any dynamically allocated space should be cleaned up appropriately with ***delete*** when you are finished with it.)

- **Hint:** Remember that a heterogeneous list is implemented using an array of **pointers** to the base class type. And as stated above, this must be created dynamically in this situation, i.e., you will need to use the ***new*** operator. If you declare your array like this:

  *Student *list[size];*

  then it is **WRONG**.

HW #6 (7)

c) Print a summary report to the output file, as specified below. You will 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

- 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 a list 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*.

HW #6 (8)

- Note that a name could include spaces – the comma will delineate勾畫 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** – Attendance, Project, Midterm, Final Exam.
- **History** – Term Paper, Midterm, Final Exam.
- **Math** – Quiz 1, Quiz 2, Quiz 3, Quiz 4, Quiz 5, Test 1, Test 2, Final Exam.

HW #6 (9)

- Output File – The output file that you 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 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.

HW #6 (10)

Sample run:

- Screen input and output: (keyboard input is underline)

```
Please enter the name of the input file.
Filename: test.txt
Please enter the name of the output file.
Filename: outfile.txt
Processing Complete.
```

HW #6 (11)

- **Sample input file**:

```
6
Bunny, Bugs
Math 90 86 80 95 100 99 96 93
Schmuckatelli, Joe
History 88 75 90
Dipwart, Marvin
English 95 76 72 88
Crack Corn, Jimmy
Math 44 58 23 76 50 59 77 68
Kirk, James T.
English 40 100 68 88
Lewinsky, Monica
History 60 72 78
```

HW #6 (12)

- **Corresponding output file:**

```
Student Grade Summary
---------------------------------

ENGLISH CLASS

Student               Final      Final      Letter
Name                  Exam       Avg        Grade
-------------------------------------------------------------------------------
Marvin Dipwart        88         80.30      B
James T. Kirk         88         80.80      B

HISTORY CLASS

Student               Final      Final      Letter
Name                  Exam       Avg        Grade
-------------------------------------------------------------------------------
Joe Schmuckatelli     90         84.25      B
Monica Lewinsky       78         71.40      C

MATH CLASS

Student               Final      Final      Letter
Name                  Exam       Avg        Grade
-------------------------------------------------------------------------------
Bugs Bunny            93         94.83      A
Jimmy Crack Corn      68         65.33      D
```