Homework 1a: Plain Text Databases

Due: 10:00 AM, August 29, 2014

Part 1 (25 points): Reading & displaying data

The file `gradebook.txt` is your database for this assignment and stores its data in plain text. The database contains a set of students defined by: 1) their first name; 2) their last name; and 3) their GPA. The information within it is stored in the following format with an entry for each individual student on its own line:

```
firstName lastName GPA
```

The first two attributes (first and last names) are stored as a string and the GPA attribute is represented as a floating point number. For clarity, the entirety of `gradebook.txt` is listed here:

```
Chris Knight 3.8
Mitch Taylor 3.6
Lazlo Hollyfeld 4.0
Paul Stephens 3.2
David Lightman 2.8
Stephen Falken 3.7
Joshua Falken 3.2
Dade Murphy 3.3
Kate Libby 3.9
```

You will create your own copy of this and name it `gradebook.txt`. Your program should read this file and store its contents in RAM. Please don't try to handle parsing first or last names containing spaces or punctuations of any kind. Only handle simple cases such as those that you see here. Once you have loaded all of the data, implement and call a `printData` method which takes as its input the collection of data you have just loaded.

The `printData` method should print out each tuple (data item) in the collection you passed it on its own line in the following format:

```
firstName lastName GPA
```

In other words, the output should look identical to the format of `gradebook.txt`.

Part 2 (10 points): Computing the average GPA

Next, create a new method called `computeAverageGPA` which takes a collection of student data as its input and returns a floating point number which represents the average GPA.

Pass the set of student information, taken from `gradebook.txt`, to the `computeAverageGPA` method and print the computed average to the screen.

Part 3 (15 points): Display the students with the top K GPAs

Create a method called `getTopK` which takes as its input a collection of student data and an integer k. This method should return a sorted collection containing the k students with the highest GPAs. The returned collection should be sorted in descending order such that the highest GPA is first.

Call `getTopK` by passing in the set of students from `gradebook.txt` and a value of 3 for k. Print the collection returned by `getTopK` by passing it to `printData`.

Extra Notes: Output formatting

It's not necessary for your program to produce any fancy output. In fact, an example of correct output (running Part 1, Part 2, and then Part 3 in sequence) could like the following:

```
Chris Knight 3.8
Mitch Taylor 3.6
Lazlo Hollyfeld 4.0
Paul Stephens 3.2
David Lightman 2.8
Stephen Falken 3.7
Joshua Falken 3.2
Dade Murphy 3.3
Kate Libby 3.9
3.5
Lazlo Hollyfeld 4.0
Kate Libby 3.9
Chris Knight 3.8
```

Also, don't worry about the formatting of floating point numbers on the output (4 instead of 4.0 is just fine).

Submission: Turning in the assignment

All assignments for this course will be turned in on the CS-Data server.

To turn in your assignment, place your work in your personal folder on the CS-Data server under the directory: `cmsc321\hw_01a`. This means you should immediately add the "cmsc321" folder to your personal student folder if it is not already there. For each subsequent homework assignment that you will turn in, you will put the files for that homework assignment in their own folder within your cmsc321 folder (hw_01b, hw_02, hw_03, and so on...).

By turning in *this* assignment you should have the following files in your homework folder.

```
\\CS-Data\Students\your_user_name\cmsc321\hw_01a\hw_01a.{cpp or java}
\\CS-Data\Students\your_user_name\cmsc321\hw_01a\gradebook.txt
```