CS240 Data Structures and Algorithms I
Assignment 3 - Hashing

Due Date

Sunday 11:59pm, November 29, 2015

Score

25

Questions and Directions

This is a project for you to get some practice with the hashing technique.

Your program is going to process score files where each line is either blank (in which case it should be ignored) or it has a name and a score on it. The name can be multiple words with any amount of white space between them. You should convert all names to strings with just one space between each word. The last word on each line is a score, which is a floating point number (**do error check if the last word is not a number**). Two sample score files ex1.txt and ex2.txt (in the folder **ex**) are shown as follows:

```
Marc Bulger          85.5
Daunte Culpepper       119
Donovan McNabb          100.6
Brett Favre         88.8
Aaron Brooks          90.5
Peyton Manning          121.2
```

```
Marc Bulger          95.5
Daunte Culpepper       129
Donovan McNabb          110.6
Brett Favre         98.8
Peyton Manning          111.2
```

Your program takes a directory from the command line that contains a list of score files. It then reads each score in every file, and for each name, it computes the average score for that name. In other words, a name can have multiple entries in a score file, and different score files can have different scores with the same name. For example, if I call your program with a directory that contains the above two files as the command line argument, the program will keep track of the names:

- Marc Bulger, with an average score of 90.5
- Daunte Culpepper, with an average score of 124
- Donovan McNabb, with an average score of 105.6
- Brett Favre, with an average score of 93.8

- Aaron Brooks, with an average score of 90.5 (Note, Aaron Brooks is only in the file **ex1.txt**)
- Peyton Manning, with an average score of 116.2

Once your program runs, it should ask the user to enter a name, and it gives **the number of scores** plus **the average score** for that name. If the name wasn't specified in any of the score files, then it says that the name isn't found. It repeats this until the user closes standard input, and your program exits.

```
> java Project3 ex
Enter name: Peyton Manning
  Peyton Manning: Avg: 116.20   # Scores: 2
Enter name: Donovan McNabb
  Donovan McNabb: Avg: 105.60   # Scores: 2
Enter name: Donald Duck
  Donald Duck not found
```

You should use the hash table with linked list collision handling to keep track of the entries. You can reuse the hash table we have implemented during the class. Feel free to design your own Key and Value for this application. For instance, you can use the name (String) as the Key, and a class that contains the total score and number of entries as the Value.

Your program will be tested with the following input files in a directory called: **hashingData**

```
> java Project3 hashingData
Enter name: Philadelphia
  Philadelphia: Avg: 24.125   # Scores: 16
Enter name: Blah
  Blah not found
Enter name: Green Bay
  Green Bay: Avg: 26.500   # Scores: 16
Enter name: Tennessee
  Tennessee: Avg: 21.500   # Scores: 16
Enter name:
...
```

How to read from command line arguments and how to read files from a directory.

Testdir.java: is able to process multiple files from a directory. You run this program with "java Testdir hashingData", where hashingData is the directory that contains all your testing data.

In this project, please develop your program based on Testdir.java. Feel free to reuse the code.

Submission

1. Your final assignment should contain the following Java files
    a. Classes needed for your Hash Table or other data structure implementation
    b. Project3.java - this class could be based on Testdir.java, but it includes the extra logic to record all the scores, calcualte the average and count the number of scores. It also asks for user input and print out the result. We will be running "java Project3 hashingData" to test your program.
2. You do not submit any documents or papers. However, for anything you want to clarify or explain, please make Java comments directly in your program. Well-commented programs are highly recommended.
3. Zip all of the above files and name it as yourname_a3.zip.
4. Submit in Blackboard.

Grading Guide

- 80%: Program correctness and completeness. If your program does not compile, you get a zero here.
- 20%: Your code. Programming style, comments, formats. Make your code clean.

Questions and Help

Please email me yusun@cpp.edu or see me during office hours if you have questions.