CS424/524 Programming Languages Spring 2022 Programming Assignment #1

Due: See Canvas for Due Dates (40 points)

Problem Statement

You are to develop a program in the Go programming language to read Student data from an input file into a list/collection. You are to store the students in a collection in such a way that when they are displayed to the screen, they are in sorted order by lastname (then firstname to break name ordering ties).

Each student will comprise a set of three lines of data in the data file in the following format:

firstname lastname
list of integer test grades
list of integer homework grades

You may assume there are no errors in the data. I will not leave data out or put non-numeric data in numeric fields. You must read until the end of the file. There will be no blank lines at the end of the input data.

You will additionally prompt the user for the weight to be applied to the test average in your overall computation. (Homeworks use the remaining weight %).

You will compute an overall class test average, an overall class homework average and an individual weighted average for each student in the list.

Summary of Operation

- Prompt the user for the input file name. DO NOT hardcode file names into your program.
- Prompt the user for the weighted % amount to apply to the tests portion of the grade.
- Prompt the user for the number of homeworks and tests.
- Open input file
- Read each student and add them to your data set
- Keep track of the number of items in the list
- Write a report summary line to the screen.
- Write each item from the list formatted to the screen, along with any other output required by the assignment
- If a student has fewer homework grades or test grades than the maximum found for any one student, add a note to the report to indicate there may be missing grades.

TURN IN:

Submit a copy of your program file to CANVAS. Make sure your name and the name of the file is in the comments at the top of each program file. Also include the environment in which you tested your program (GoLand, linux command line at UAH, etc).

NOTE:

I am not teaching the Go language in class. It is up to you to give yourself enough time to explore and learn the language in order to complete this assignment.

The Go language tools can be downloaded from https://golang.org/

There is a tutorial for learning the basics of Go at: https://tour.golang.org/

Other Requirements

- Your program must be well-commented. Comment all variables, functions and remember to have a section of comments at the top of your program that includes your name, date, course section and a description of what your program does.
- Use good variable names.
- Think about data objects and what they contain. Even though GO is not fully object-oriented (as in inheritance) we should be using good design practices.
- Use good and consistent naming conventions for data members.
- Use proper code indentation to make sure your program is easy to read and understand.

Sample Execution (user input shown in yellow)

```
Welcome to the gradebook calculator test program.  I am going to
read students from an input data file.  You will tell me the name of
your input file.


Enter the name of your input file:  studentinput.txt
Enter the % amount to weight test in overall avg: 45.0

Tests will be weighted 45.0%, Homework weighted 55.0%

How many homework assignments are there? 5
How many test grades are there? 3

. . .   ← note remaining report data will display here (see samples below)


End of Program 1
```

Sample Input File and Corresponding Output Data
sample input text file

```
Beth Allen
100 100 100
75 75 85 85 95
Joe Davis
50 85 96
50 50 85 85 99
Reza Albedin
88 88 88
45 45 99 99
```

Note – there should be NO blank lines after the last line of data in an input file.

Sample output (formatted so grades line up)

```
GRADE REPORT --- 3 STUDENTS FOUND IN FILE
TEST WEIGHT: 45.0%
HOMEWORK WEIGHT: 55.0%
OVERALL AVERAGE is 81.7

    STUDENT NAME     :     TESTS      HOMEWORKS    AVG
-------------------------------------------------------
       Albedin, Reza :    88.0 (3)     72.0 (4)    79.2    ** may be missing a homework **
         Allen, Beth :   100.0 (3)     83.0 (5)    90.7
          Davis, Joe :    77.0 (3)     73.8 (5)    75.2
```

General Grading Rubric (out of 40 points)

I provide this as a general guideline for how each area of a program is to be rated for grading purposes.

Performance Element	5 - Excellent	4 – Very good	3 - Good	2 - Limited	1 - Inadequate
Specifications ***(20 points)***	Program runs & meets all specifications	Runs, gets correct answers, output displayed correctly, meets most other specifications	Runs, gets correct answers, output is not displayed correctly or other specifications not met	Runs, get some correct results, does not meet most specifications	Program does not run, runs but gets no results or mostly wrong results
Readability ***(5 points)***	Code is well organized and easy to follow	Most of the code is well organized & easy to read	Parts of the code are easy to read but the organization is not good	Code is readable if the reader knows what it is supposed to be doing	Code is poorly organized and very difficult to read
Documentation (i.e. comments) ***(5 points)***	Documentation is clearly written & explains what the program as a whole should do; comment blocks introduce each function	Documentation is brief but helpful, includes header information for functions	Documentation consists of embedded comments and some header information for functions	Little or no documentation other than obvious embedded comments	Little or no documentation
Software Architecture ***(10 points)***	Displays excellent information hiding and modularity	Displays good information hiding and modularity	Displays some information hiding and modularity	Displays some information hiding or modularity but not both	Little or no structure