CS1336

Homework #6-1

Assigned April 11, due April 18 at 11:59 PM

This homework will allow you to practice reading from a file, menu interface, and loops. You have the option to submit the basic version, or the extra credit version. It is recommended you implement and have the basic version working first before you attempt the extra credit version.

HW6-1- Basic Version (You can earn up to 100 points out of 100)

Write a program that reads GPA scores from a file called "studentData.txt" and displays a menu asking the user to choose a, b, or c. In addition, the user also has to option to quit by choosing d. As long as the user does not choose to quit, the program loops over displaying the menu. Your program must also do input validation and print an error message if the user types anything other than a, b, c, or d, then display the menu again and ask the user to reenter the user's choice.

a) The number of scores in the file
b) The sum of all the scores in the file
c) The average of all scores in the file
d) Quit

If the user chooses a), b), or c), your program should print the result on the screen (cout).

"studentData.txt" file can be downloaded from eLearning. Use that file to test your code, but your code must work for any file which contains GPA scores, where the number of GPA scores is arbitrary.

1. Additional requirements – Make sure you meet all the requirements to avoid losing points

1. To complete the assignment, you are **required to use only what has been taught in class**. If you have prior programming experience, refrain from using more advanced C++ constructs, so all the homework programs can be graded on a consistent basis.
2. Make sure you follow the requirements in the "Homework Notes Function Headers".
3. You are allowed to hard code the file name in your program.
4. You are required to implement the following functions
   - `getUserChoice`: displays the menu, prompts the user to enter the choice, performs input validation, displays the menu again if the input is invalid. Returns the user's choice if it is valid
   - `calcNumber`: open the file (checks for file open failure and exit the program if there is failure), read the data, calculate the number of values in the file, then close the file, and returns the result.
   - `calcSum`: open the file (checks for file open failure and exit the program if there is failure), read the data, calculate the sum of values in the file, then close the file, and returns the result.

- `calcAvg`: open the file (checks for file open failure and exit the program if there is failure), read the data, calculate the average of values in the file, then close the file, and returns the result
- Your `main()` function implements the following pseudocode

    call getUserChoice
    depending on the user's choice, call the appropriate calculation function, or quit
    print the result on the screen
    loop while user's choice is not Quit

2. Implementation Notes

If you open the file only at the start of your program and close it only at the end, you will get incorrect results. **Make sure you close the file and reopen it at every new calculation, as indicated above in the functions**. Doing so will allow your program to read from the beginning of the file.

HW6-1 - Extra Credit (You can earn up to 115 points out of 100)

In the extra credit version, your program reads from a more complicated file, which contains student records. Each record is a student's name, followed by that student's GPA.

An example of file having 5 student records is shown below, and you can download the file "studentDataXtraCredit.txt" from eLearning. Use that file to test your code, but the file is only an example, and your code must work for an arbitrary number of records in the file. You can assume the student's name has no whitespace in it.

Kai 3.8

Alice 3.9

Seppo 3.2

Ken 3.0

Anna 3.8

The program displays a menu asking the user to choose a, b, c, or d. In addition, the user also has to option to quit by choosing e. As long as the user does not choose to quit, the program loops over displaying the menu. Your program must also do input validation and print an error message if the user types anything other than a, b, c, d, or e, then display the menu again and ask the user to reenter the user's choice.
a) The number of scores in the file
b) The sum of all the scores in the file
c) The average of all scores in the file
d) Highest score in the file, along with the name of the student who achieved that score
e) Quit

If the user chooses a), b), c), or d), your program should print the result on the screen (cout).

3. Additional requirements – Make sure you meet all the requirements to avoid losing points

5. To complete the assignment, you are **required to use only what has been taught in class**. If you have prior programming experience, refrain from using more advanced C++ constructs, so all the homework programs can be graded on a consistent basis.
6. Make sure you follow the requirements in the "Homework Notes Function Headers".
7. You are allowed to hard code the file name in your program.
8. You are required to implement the following functions
   - `getUserChoice`: displays the menu, prompts the user to enter the choice, performs input validation, displays the menu again if the input is invalid. Returns the user's choice if it is valid
   - `calcNumber`: open the file (checks for file open failure and exit the program if there is failure), read the data, calculate the number of values in the file, then close the file, and returns the result.
   - `calcSum`: open the file (checks for file open failure and exit the program if there is failure), read the data, calculate the sum of values in the file, then close the file, and returns the result.
   - `calcAvg`: open the file (checks for file open failure and exit the program if there is failure), read the data, calculate the average of values in the file, then close the file, and returns the result
   - `findHighest`: open the file (checks for file open failure and exit the program if there is failure), read the data, finds the highest GPA and the corresponding student's name in the file, then close the file, and returns the result
   - Your `main()` function implements the following pseudocode

     call getUserChoice
     depending on the user's choice, call the appropriate calculation function, or quit
     print the result on the screen
     loop while user's choice is not Quit

4. Implementation Hints

If you open the file only at the start of your program and close it only at the end, you will get incorrect results. **Make sure you close the file and reopen it at every new calculation, as indicated above in the functions**. Doing so will allow your program to read from the beginning of the file.

`findHighest`: To find the highest score, you can use essentially the same algorithm as provided on the lecture slides to find the highest value in an array. However, the algorithm needs to be adapted, because the score values are not readily available in an array, but will be read on the fly from the file. Instead of initializing `highest` to `numbers[0]`, initialize it to -1. `highest` will be compared with the score read from the file, instead of `numbers[count]`.

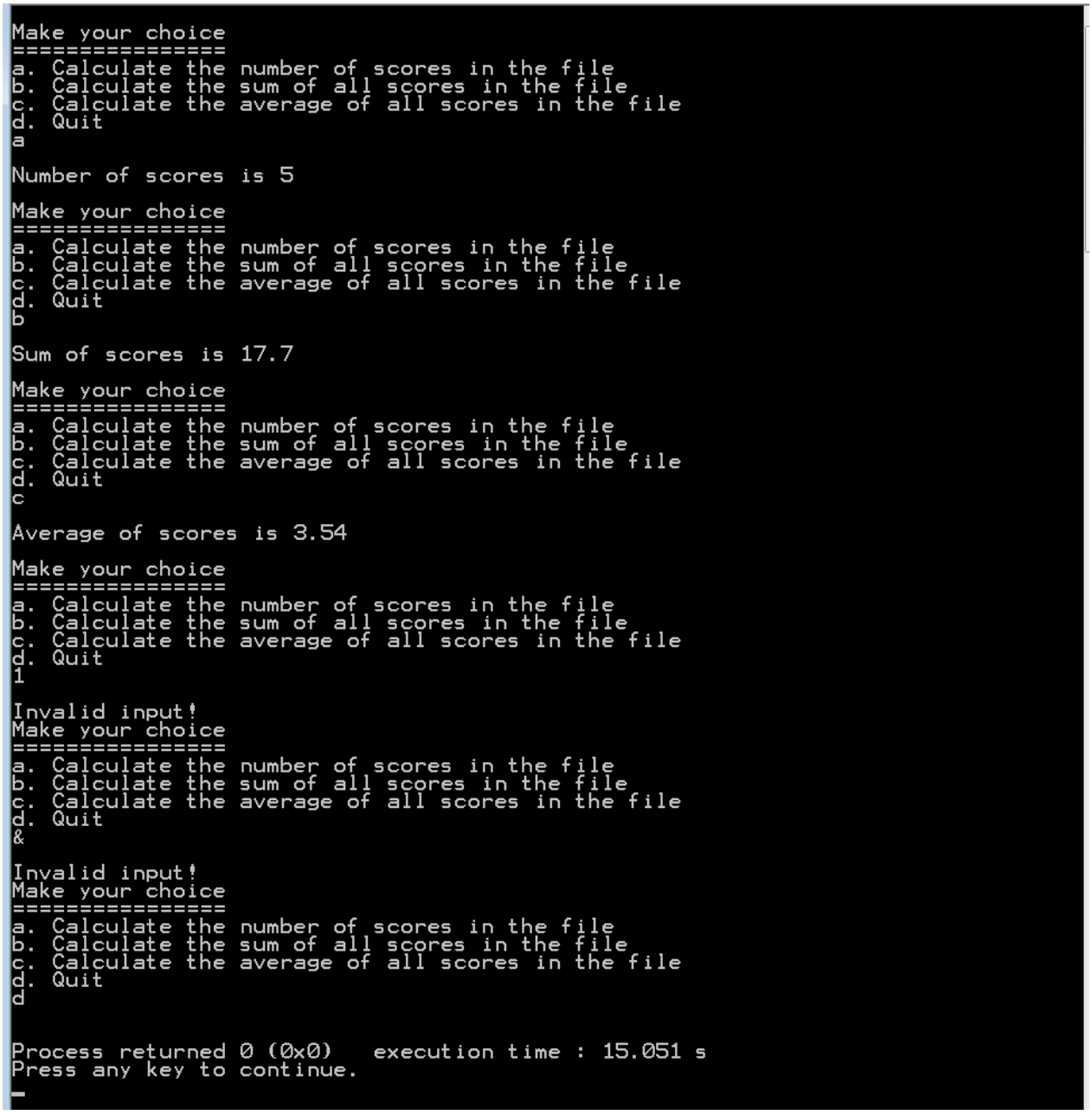
To enable `findHighest` to return two data items (highest score and student's name), you can use a `return` statement and a reference variable.

```
/* This function reads student records from a file
and finds the highest score, along with the name of the
```

```
student.
The name is returned as a string, and the highest score
is passed back as a reference variable
*/
string findHighest(double & score)
{
      // Function body
}
```

Example Output

1. Basic Version



```
Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Quit
a

Number of scores is 5

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Quit
b

Sum of scores is 17.7

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Quit
c

Average of scores is 3.54

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Quit
1

Invalid input!
Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Quit
&

Invalid input!
Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Quit
d


Process returned 0 (0x0)   execution time : 15.051 s
Press any key to continue.
```

2. Extra Credit Version

```
Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
a

Number of scores is 5

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
b

Sum of scores is 17.7

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
c

Average of scores is 3.54

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
d

Highest score is 3.9, achieved by Alice

Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
1

Invalid input!
```

```
Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
&

Invalid input!
Make your choice
================
a. Calculate the number of scores in the file
b. Calculate the sum of all scores in the file
c. Calculate the average of all scores in the file
d. Find highest score and student's name
e. Quit
e


Process returned 0 (0x0)   execution time : 17.854 s
Press any key to continue.
```