CSCE 1030 – Homework 5

Due: 11:59 PM on Monday, November 10, 2014 CST

NOTE: You are given extra time to do this homework assignment because of your exam, and because it is more complicated and difficult than the previous ones. If you don't start working on this assignment right away, you will not complete it successfully.

Problem Statement

You will create a simple database of grades. Your program will allow storing of data in the database, retrieving data, and calculating individual and group statistics.

1. You will define arrays to store the grade information of UNT students. One array will hold the EUID as an integer (e.g. 1034845), one array will hold the homework average of a student (e.g. 84.93), one array will hold the grade of exam 1 (e.g. 84), and another one the grade of exam 2 (e.g. 92). The grades of a student will be in the same locations in the corresponding arrays. So, a student whose EUID is stored in location i of the EUID array will have his or her homework average, exam 1 grade, and exam 2 grade stored in location i of the appropriate arrays.

2. You will write a function that will read data from a file to populate your grade database. Since you don't know how to read from files yet, you will use C commands to read from the keyboard (e.g. get(), gets() or scanf()) and use the Linux command "<" to get input. A command might look like "./a.out<inputFile" (More of this in class!). The input file will look like:
   198390 45.6 39 78 1938440 50.03 78 34
   ETC
   The end of the data will be indicated by a -1 in the place of the EUID. You may assume that there will be no more than 200 students in the data file. An example data file can be copied from `/home/ct0179/hw5InputExample`

3. Next you will prompt the user for the information they want to retrieve from the database. The options will be:
   1. get complete grade data of a student – this must be followed by prompting for the student EUID
   2. get current grade average and letter grade of a student - this must be followed by prompting for the student EUID – Compute grade as homework*0.5+exam1*0.25+exam2*0.25 – Letter grade assigned in straight scale as shown in class syllabus. To compute the letter grade you should round off to the next highest grade (e.g. 79.50 or higher is a B; 79.49999 and lower is a C).
   3. get class average for a particular assignment (homework, exam 1 or exam 2) – this must be followed by a prompt asking the use to type in "homework," "exam 1" or "exam 2" which will be handled as strings
   4. all numbers output will be limited to 2 digits after the decimal point (if there is one)

5. exit the program

Example session for this part (program output in BLUE, user input in RED):

```
Please enter what information you want out of the database:
1.  student grade data
2.  student grade average
3.  class average for an assignment
4.  exit
> 1
Please enter student EUID: 1893839
EUID: 1893839
Homework: 84.34
Exam 1: 84
Exam 2: 76

Please enter what information you want out of the database:
1.  student grade data
2.  student grade average
3.  class average for an assignment
4.  exit
> 2
Please enter student EUID: 1893839
EUID: 1893839
Grade average: 81.56
Letter grade: B

Please enter what information you want out of the database:
1.  student grade data
2.  student grade average
3.  class average for an assignment
4.  exit
> 3
Please enter the assignment: homewrj
Sorry, I do not understand this word

Please enter what information you want out of the database:
1.  student grade data
2.  student grade average
3.  class average for an assignment
4.  exit
> 3
```

CSCE 1030 – Homework 5

Due: 11:59 PM on Monday, November 10, 2014 CST

```
Please enter the assignment: exam 1
Exam 1 class average: 68.98
```

To complete this program you will need to use arrays, strings, and functions. You will have to put together almost everything you have learned so far in the class.

Additional requirements for this homework:

1. You will write a function to display your personal information (department and course number, program number, your name, your EUID, and your e-mail address).
2. You will write a function to print out the menu.
3. As already mentioned, you will use a function (a SINGLE function) to read in the data into the database arrays
4. You will use a function to get the information of a particular student out of the database.
5. You will use a **void** function to compute the average grade of a student
6. You will use a function to compute the letter grade of a student
7. You will use a **void** function (a SINGLE function) to compute the average of any assignment.

These is the minimum number of functions you will use. You may, of course, use more.

Good luck!

Design:

On a piece of paper, write down in English the sequence of steps you will perform to solve the problem. Pretend this is a “recipe” for someone else to follow. Refine your “recipe” until it is clear. Be sure to include the steps for prompting for input, etc. This document should contain the sequence of steps used.

Type these steps into a document (Word, txt, PDF, etc.). Also be sure to include your algorithm steps as comments in your code file. Do this before you start coding as completing it afterwards does not help you in learning the design process!

Implementation:

CSCE 1030 – Homework 5

Due: 11:59 PM on Monday, November 10, 2014 CST

Now that you have a working design, your next step is to translate these steps into C code. Use the algorithm development techniques discussed in class to implement your solution to the problem above. Add your C code a little at a time, and compile and test as you go.

Remember to add your comments to your code to explain your program. Do this before/during programming instead of waiting until the end. At a minimum, you should comment the program header (e.g., name, class, date, brief description of the program, etc.), all variables (i.e., what they are used for), function headers (see below), and specific "blocks" of code. For example, use comments to describe the inputs, the formulas used, and any other important steps in your code.

Your program will be graded based largely upon whether it works correctly on a CSE Department machine, so you should make sure your program compiles and runs on a CSE machine.

Your program will also be graded based upon your programming style. At the very least, your program should include:

- A consistent indentation style as recommended in the textbook and in class;
- Meaningful variable names;
- A program header comment section that includes: your name, e-mail address, and a brief description of the program.
- Function headers as described below.

Documentation:

When you have completed your C program, write a short report (2–3 paragraphs) describing:

1. what the objectives were,
2. what you did to solve the problem,
3. the status of the program. Does it work properly for all test cases? Are there any known problems?
4. You will also include an example of your program working. To do so you will need to study the Linux command "script." Do not do a screen capture.
5. Some calculations or verification to show that your examples of your program working are correct. In other words, show that your program calculates grades, averages, etc. correctly, by comparing the output of your program to a few (a few!) hand calculations.

Save this report in a separate file to be submitted electronically. You should also include any specific instructions required to compile or execute your code, such as linking a specific library (e.g., "-lm" for the math library, or "–std=c99", etc.).

Testing:

Test your program to check that it operates as desired with a variety of inputs, including any boundary conditions. Compare the answers your code gives with the ones you get from hand-calculations.

Homework Submission:

In this class, we will be using electronic homework submission to make sure that all students hand their programming projects (and labs) on time. You will submit your program source file to the class website through the "**Homework 5**" drop box by the due date and time.

Note that this project must be done individually. The program will be checked using a code plagiarism tool against other solutions, so please ensure that all work submitted is your own.

Note that the dates on your electronic submission will be used to verify that you met the due date above. All homework up to 24 hours late will receive a 50% grade penalty. Later submissions will receive zero credit, so hand in your best effort on the due date.

Summary:

- You will design a solution to the problem.
- You will implement it on the CSE machines using C. You will make sure to use good style, good variable names, indentation, etc. You will compile, run, and test your code.
- You will write a brief report describing what your code does and how well it works.
- You will submit electronically your C code, your design, and your brief report.

General Guidelines (for ALL of your programming assignments):

- Your program's output should initially display the department and course number, program number, your name, your EUID, and your e-mail address.
- Use meaningful variable names.
- Use appropriate indentation.
- Use comments, including a program header. Example program header:

```
/*
Author: Jane Doe (Jane.Doe@my.unt.edu)
Date: 9/18/2014
Purpose: This program reads in three numbers and computes their
average
*/
```

- Add a header to each function. Example function header:

```
/*
Function: deposit
Parameters: a double representing the account balance,
and a double representing the deposit amount
Return: a double, the account balance after the deposit
Description: This function computes the account balance after a
deposit
*/
```