EE441 HOMEWORK 1

EE441 Classroom Interface

Due: November 20, 2021 , 23:55

***You may ask HW-1 related questions to** *Kamil SERT - ksert@metu.edu.tr*

Submission

- Use **Code::Blocks IDE** and choose **GNU GCC Compiler** while creating your project. Name your Project as "**eXXXXXXX_HW1**", where X's are your **7-digit student ID number**. Send the whole Project folder compressed in a rar or zip file. Name your submission as "**eXXXXXXX_ee441_hw1.rar**". You will **not** get full credit if you fail to submit your project folder as required.
- The homework must be written in **C++** (**not** in C or any other language).
- Your C++ program should follow object oriented principles, including proper class and method usage and should be correctly structured including private and public components. Your work will be graded on its correctness, efficiency, clarity and readability as a whole.
- You should insert **comments** in your source code at appropriate places without including any unnecessary detail. Comments will be graded.
- **Late submissions** are welcome, but are penalized according to the following policy:
  - 1 day late submission : HW will be evaluated out of **70**.
  - 2 days late submission : HW will be evaluated out of **50**.
  - 3 days late submission : HW will be evaluated out of **30**.
  - Later submissions : HW will NOT be evaluated.
- You should **not** call any external programs in your code.
- **Check** what you upload. Do not send corrupted, wrong files or unnecessary files.
- Programs giving compilation errors will not be evaluated!
- The homework is to be prepared **individually**. Group work is **not** allowed. Your code will be **checked** for cheating.
- The design should be your original work. However, if you partially make use of a code from the Web, give proper **reference** to the related website in your comments. Uncited use is unacceptable.
- **METU honor code is essential**. Do **not** share your code. Any kind of involvement in cheating will result in a **zero** grade, for **both** providers and receivers.

Introduction

In this homework, you are asked to implement a user interface and the corresponding database of students for 'EE441' course. The database entries will be instances of a custom-made *Student* class. The database itself will be an instance of a *Course* class. *Student* and *Course* classes and the user interface for the 'EE441' course will be designed and implemented in this homework.

Student Class

Student class consists of the following;

- Private members:
  - ID
  - name
  - surname
  - quiz_scores (an array for 1st, 2nd, 3rd and 4th quiz scores)
  - hw_scores (an array for 1st, 2nd and 3rd homework scores)
  - final_score (final exam score)
- Public methods:
  - Constructor
  - Getter and setter methods for all members (e.g. getID, setQuizScores, etc.)
  - overallCourseScore (calculates overall score of a student considering quiz, homework and final exam scores and weights)

Course Class

Course class consists of the following;

- Private members:
  - students[MAX_SIZE] (an array for Student instances, MAX_SIZE is 10)
  - num (the current number of students in EE441 course)
  - exam weights
    - weightQ (weight for the quizzes, default 30%)
    - weightHW (weight for the homeworks, default 30%)
    - weightF (weight for the final exam, default 40%)
- Public methods:
  - Constructor
  - getNum (returns how many students are in the course)
  - addNewStudent (adds new student to the course)
  - updateWeights (updates weights)
  - getStudentInfo (returns a Student information given an ID)
  - calculateAverage (calculates course averages for quiz, homework, final exam and overall course scores)

User Interface

The program should ask the user for his/her input by offering only the following options as a menu:

1. Add student
2. Search a student by ID
3. Change a student's score
4. Update weights
5. Show classroom average
6. Show students with overall score above a threshold
7. Show students with overall score below a threshold
8. Exit

For user interface you will implement only the following functions whose inputs are indicated in brackets (*it is not allowed to change inputs or the order of the inputs*):

- ***addNewStudent(ID, name, surname)*** : Adds new student to the course. All the scores will be initialized as '0'.
- ***showStudent(ID)*** : Shows all information of a student given his/her ID.
- ***changeStudentScores(ID, quiz_scores, hw_scores, final_score)*** : Changes student's scores given his/her ID.
- ***changeWeights(weightQ, weightHW, weightF)*** : Changes all exam weights.
- ***showAverage()*** : Shows classroom average for each exam and overall score.
- ***showAbove(threshold)*** : Lists all students' information above a certain overall score threshold.
- ***showBelow(threshold)*** : Lists all students' information below a certain overall score threshold.

Remarks

- Your program should check for illogical (e.g. negative grade) and invalid (e.g. invalid ID) cases, and give appropriate warnings.
- If necessary, you can implement additional methods and functions (but **not** class data members)

Example

```
EE441 Classroom Interface
Choose your option:
1) Add a student
2) Search a student by ID
3) Change a student's score
4) Update weights
5) Show class average
6) Show students with overall score above a threshold
7) Show students with overall score below a threshold
8) Exit

Enter your option: 1
Enter student's ID, name, surname and scores (quizzes, homeworks, final)
123 Alice Evans 60 55 40 85 70 75 80 76

Enter your option: 1
Enter student's ID, name, surname and scores (quizzes, homeworks, final)
265 Bob Crane 90 85 70 85 100 95 88 93

Enter your option: 1
Enter student's ID, name, surname and scores (quizzes, homeworks, final)
1602 Charlie Doe 45 35 60 25 50 54 48 54

Enter your option: 2
Enter student ID
123
123  Alice   Evans   60  55  40  85  70  75  80  76

Enter your option: 2
Enter student ID
553
Invalid ID

Enter your option: 5
Quizzes-Avg: 61,25
Homeworks-Avg: 73,33
Final-Avg: 74,33

Enter your option: 7
Enter threshold: 55
1602 Charlie Doe Avg:49,18

Enter your option: 8
Program closed.



...Program finished with exit code 0
Press ENTER to exit console.
```