BLG252E - Object Oriented Programming

Homework-3

Assignment Date :	20.12.2016
Due Date :	03.01.2017 at 18:00

Write C++ codes of the two classes shown in the UML class diagrams below.
Notice that, there is no direct association between the classes.
(Member access symbols : - is private, + is public.)

Student
+ id_number : int - fname : string - lname : string - grades : **map <string , int>**
+ Student (int id_number, string fname, string lname) + add_grade (string course_code , int points) : void + print() : void

Course
+ course_code : string - grades : **map <int , int>**
+ Course (string course_code) + add_grade (int id_number , int points) : void + print() : void

STUDENT CLASS

- id_number, fname, lname are the student identification number, first name, and last name respectively.
- The ***grades*** member variable is a Standard Template Library (STL) **map.**
  The map template parameters are <string , int>.
  First field (string) is the Key field representing the course codes.
  Second field (int) is the Value field representing the corresponding points.
- Parameterized constructor should initialize the data members with given parameters.
- The add_grade function should add the given course code and the given points to the grades map.
- The print function should display the student id number, first name, and last name.
  In order to display the grades map, function should call the independent templated function Print_Grades which is described below.

COURSE CLASS

- The ***grades*** member variable is a Standard Template Library (STL) **map.**
  The map template parameters are <int , int>.
  First field (int) is the Key field representing the student id numbers.
  Second field (int) is the Value field representing the corresponding points.
- Parameterized constructor should initialize the course code member with given parameter.
- The add_grade function should add the given student id number and the given points to the grades map.
- The print function should display the course code.
  In order to display the grades map, function should call the independent templated function Print_Grades which is described below.

TEMPLATED FUNCTION

Write the independent (non-member) templated function whose prototype is given below.

template < typename T > void Print_Grades(map < T , int > grades);

The grades argument is a map, whose template parameters are <T , int>.
First field (T) is the Key field representing either the course codes or the student id numbers.
Second field (int) is the Value field representing the corresponding points.

By using a map iterator, function should loop thru the given map object and display contents of it on screen.
Function should also calculate and display the average of points in the grades map object.

MAIN PROGRAM

Write the main program to do followings.

- Define a vector of **Student** class by using the STL **vector**.
- Define a vector of **Course** class by using the STL **vector**.
- By using either fscanf or ifstream, read all records from the **DATA.TXT file** and build both two vectors. The data file contains student id number, first name, last name, course code, and points information. Program should call the add_grade functions of the Student and Course vectors.
- In two printing phases, loop thru the Student and Course vectors and call their print function.

EXAMPLE SCREEN OUTPUT

```
PHASE1 : PRINTING BY STUDENTS

STUDENT : 443369 Ahmet Turhan
Grades :
        BLG102  70
        EHB205  80
        MAT101  90
Average = 80
--------------------------------
STUDENT : 704326 Furkan Gunsel
Grades :
        EHB303  95
        MAT102  75
Average = 85
--------------------------------
STUDENT : 604283 Kemal Yakut
Grades :
        BLG102  50
        MAT101  60
        MAT102  80
Average = 63.3333
--------------------------------
STUDENT : 187194 Bulent Turker
Grades :
        EHB107  70
        EHB205  60
        EHB401  100
Average = 76.6667
--------------------------------
```

```
**********************************

PHASE2 : PRINTING BY COURSES

COURSE : MAT101
Grades :
        443369  90
        604283  60
Average = 75
---------------------------------
COURSE : BLG102
Grades :
        443369  70
        604283  50
Average = 60
---------------------------------
COURSE : EHB205
Grades :
        187194  60
        443369  80
Average = 70
---------------------------------
COURSE : MAT102
Grades :
        604283  80
        704326  75
Average = 77.5
---------------------------------
COURSE : EHB303
Grades :
        704326  95
Average = 95
---------------------------------
COURSE : EHB401
Grades :
        187194  100
Average = 100
---------------------------------
COURSE : EHB107
Grades :
        187194  70
Average = 70
---------------------------------

Press any key to continue . . .
```

IMPORTANT RULES ABOUT BLG252E HOMEWORKS

1) You must do the homeworks by yourself individually.

- Copying, collaboration, getting help is absolutely not permitted.
- A student should never give his homework to other students.
- All submitted student homework files will be compared by using an automatic detection software system (such as Moss, JPlag, etc).
- If significant amount of similarities are found between any files, it will be considered as cheating; and those homework grades will be zero.

2) You should submit your homework file to Ninova system only.

- Email submissions or late submission requests are not accepted.
- Ninova homework system closes itself automatically at the deadline time.
  Therefore you should not wait for homework submission until the last minutes.
- You should submit only a file with *.cpp extension to Ninova.
  Other types of files (such as c, txt, docx, zip, rar, etc.) are not accepted.
- If you make any changes in your homework file, you can re-submit it to Ninova within the deadline time.
  In that case, only the last submitted file is kept in the system by Ninova.

3) Homeworks will be graded by the course assistant and results will be announced at Ninova.

4) The following criteria will be considered when grading the homeworks.

Your program should ;

- be compilable with all standard compilers (Dev-C++, Linux, etc.) without any syntax errors.
- not include non-portable header files such as <conio.h> , <stdafx.h> , etc.
- work correctly, effectively, and display expected outputs.
- be written according to given specifications.
- have a consistent coding style (indentation, comment lines, valid variable names).
- contain the following information at the beginning of your source file.
  (otherwise 5 points will deducted from the homework grade).

```
/**************************
Student Number : 123456789
Student Name   : Aaa Bbb
Course         : BLG252E
CRN            : 12345
Term           : 2016-Fall
Homework       : #3
**************************/
```