ITECH 7603

Advanced Programming
School of Science, Information Technology and Engineering

University of Ballarat
Learn to succeed

ASSIGNMENT 2

TEACHING PERIOD 2013/07

0.Introduction. In this assignment you are required to develop a program that allows the creation, viewing and storage of information about ITECH7603 Students. In order to complete the task you have to investigate how to use the following STL(Standard Template Library) containers - vector, set, and map.
The information about ITECH7603 students consists of: first name, surname, and marks for Lab Test, Assignment1, Assignment2, and Final Exam. This information should be stored in the binary file "ITECH7603Students.bin". The program should also calculate and display the marks and grade for each student.

The following describes how to calculate the marks and grade:

Lab Test Mark – is an integer from 0 to 10;
Assignment1 Mark – is an integer from 0 to 20;
Assignment2 Mark – is an integer from 0 to 20;
Final exam Mark – is an integer from 0 to 50;

A total mark is a sum of the previous four marks, so it is an integer from 0 to 100.

A student passes the course if and only if
(Assignment1 Mark + Assignment2 Mark) >= 20 and (Lab Test Mark + Final Exam Mark) >= 30

There are 5 grades, which are calculated according to the following rules:

F – if the Total Mark is less then 40;
MF – if (40 <= Total Mark < 50) or (Total Mark >= 50 but student didn't pass);
P – if (50 <= Total Mark < 60) and student passed;
C – if (60 <= Total Mark < 70) and student passed;
D – if (70 <= Total Mark < 80) and student passed;
HD – if Total Mark >= 80;

1. Class Student (or ITECH7603Student).

Develop a Student class that has the following header file:

```
#ifndef STUDENT_H
#define STUDENT_H
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

class Student{
public:
Student();
Student(string name,string surname,int a1, int a2, int test, int exam);

string getName()const;
string getSurname()const;
int getAssignment1Mark()const;
int getAssignment2Mark()const;
int getLabTestMark()const;
int getExamMark()const;

void setAssignment1Mark(int);
void setAssignment2Mark(int);
void setLabTestMark(int);
void setExamMark(int);

bool passed()const;
string getGrade()const;
friend ostream& operator<<(ostream& stream, const Student &);
bool operator<(const Student &)const;

private:
string name;
string surname;
int assignment1Mark;
int assignment2Mark;
int labTestMark;
int examMark;
};
#endif
```

You are required to implement the class, i.e. to develop the implementation file "Student.cpp" that conforms to the following specifications:

(1) The data fields have self explanatory identifiers, and their meaning should be clear from the Introduction.

(2) The "set" functions allow the user to set the assessment task marks within the specified margins, e.g. setExamMark ensures that the examMark is in the appropriate range – from 0 to 50.

[0.25 ×4 = 1 mark]

(3) The function getGrade() returns the student's grade, calculated as specified in the Introduction. The rest of the "get" functions simply return the values of the corresponding fields.

[1 mark]

(4) The function passed() returns true if `this` student , has passed the course, and returns false otherwise (see the Introduction).

[1 mark]

(5) `friend ostream& operator<<(ostream& stream, const Student &)` – the function overloads << operator, which allows a Student object information to be output in the following format:

```
Underwood Scott:
      Assignment 1      8
      Assignment 2     16
      Lab Test          0
      Exam             34
      Grade             P
```

[2 marks]

(6) `bool operator<(const Student &student)const` – the function overloads < operator, which allows to compare student's full names lexicographically.

[2 marks]

[Total for Task 1: 7 marks]

2. Class ITECH7603Class.

The class has the following header file. Objects of this *class* represent ITECH7603 teaching classes (student groups).

```
#ifndef ITECH7603CLASS_H
#define ITECH7603CLASS_H
#include "Student.h"
#include <set>
#include <map>
#include <fstream>

class ITECH7603Class{
public:
      ITECH7603Class();
      ITECH7603Class(int dummy);
      ITECH7603Class(set<Student>*);
      ~ITECH7603Class();
      void addStudent(Student*);
      map<string,Student>* getGroup();
      void save();
private:
      map<string,Student>* group;
};
#endif
```

You are required to implement the class, i.e. to develop the implementation file "ITECH7603Class.cpp" that conforms to the following specifications:

(1) The only field of the class is a pointer to a "`map<string,Student>`" object representing a teaching group(or class), which is a map (an associative STL container) that contains the pairs:

Key: a student's full name (i.e. name + " " + surname)

Value: corresponding Student object.

In this assignment you should assume that there are no different students with identical full names in ITECH7603 teaching classes.

(2) `void addStudent(Student* studentPtr)`- the function adds the pair

```
Key: (*studentPtr) student's full name
Value: (*studentPtr)
```

To the (*group) map;

Please note that instead of actual objects you are supposed to use pointers as function parameters.

[1 mark]

(3) `void save()`- the function writes the Student objects stored as values in the (*group) map to the "ITECH7603Students.bin" file. "ITECH7603Students.bin" is a binary file, and you are required to write the objects to the file using something like

```
out.write(reinterpret_cast<char*>(s), sizeof(Student));
```

[2 marks]

(4) Constructor ITECH7603Class() and the destructor:

A default constructor initializes dynamically the group field.

The destructor deletes the group pointer.

[0.5×2 = 1 mark]

(5) Constructor `ITECH7603Class(int dummy)`:

This constructor does not require any parameter. However, to distinguish it from the default constructor you should use some dummy argument (say of `int` type). This constructor opens the "ITECH7603Students.bin" file, if such file exists, and reads information from the file into the (*group) map using addStudent() function. If the file "ITECH7603Students.bin" does not exist, the constructor behaves as the default constructor.

[2 marks]

(6) Constructor `ITECH7603Class(set<Student>* students)`:

This constructor takes a set of Student objects (in fact, a pointer to the set) as an argument and adds students from the set to the (*group) map, i.e. initializes the group field.

[2 marks]

[Total for Task 2: 8 marks]

3. TestAssignment2.cpp

This is a test program that tests the Student and ITECH7603Class classes.

In this assignment you are provided with three input text files associated with this program:

- firstNamesBoys.txt – contains first names for boys
- firstNamesGirls.txt – contains first names for girls
- lastNames.txt - contains last names

Each of these input files contains one name per line. Hence "firstNamesBoys.txt" will contain one boy's name per line, and "lastNames.txt" contains one surname per line. These input files will be used to create a random set of Student objects to test your application.

The test program should conform to the following specifications:

(1) There are three three global variables declared in the program:

```
vector<string>* names = new vector<string>();
vector<string>* surnames = new vector<string>();
set<Student>* students = new set<Student>();
```

(2) The test program should define a function with the following prototype:

```
void readInputFiles();
```

This function does the following:

- Opens and reads firstNamesBoys.txt. Each name should be read into a string and added to the vector<string>* names.
- Opens and reads firstNamesGirls.txt. Each name should be read into a string and added to the vector<string>* names.
- Opens and reads lastNames.txt. Each name should be read into a string and added to the vector<string>* surnames.

[2 marks]

(3) The test program should define a function with the following prototype:

```
void createRandomStudentSet(int n);
```

This function creates Student objects choosing values for their fields randomly:

- Value for first name is chosen randomly from the vector (*names).
- Value for surname is chosen randomly from the vector (*surnames).
- Values for the rest of fields are chosen randomly from the corresponding ranges, e.g. value for the examMark is a random value from 0 to 50.

If thus created Student object has passed the course it is added to the (*students) set (an associative container from STL). You should continue adding random Student objects to the set until the total number of Student objects in the set is *n*.

[2 marks]

(4) The main function performs the following tasks:

- Invokes consecutively functions readInputFiles(), createRandomStudentSet(12).
- Then creates a ITECH7603Class object (or a pointer to the object) using the constructor ITECH7603Class(students), and saves it (to the file "ITECH7603Students.bin").
- Creates another ITECH7603Class object using the constructor with a "dummy" parameter and prints information of the Student objects from the (*group) map field of the class.

[1 mark]

[Total for Task 3: 5 marks]

The following is a sample output of the test program (number of students is 5):

```
Adams Tiana:
    Assignment 1    17
    Assignment 2     5
    Lab Test         4
    Exam            34
    Grade            C

Barley Tia:
    Assignment 1     9
    Assignment 2    19
    Lab Test         5
    Exam            41
    Grade            D

Cameron Trent:
    Assignment 1    19
    Assignment 2    10
    Lab Test         4
    Exam            41
    Grade            D

Cannon Isabel:
    Assignment 1    17
    Assignment 2    19
    Lab Test         8
    Exam            25
    Grade            C

Davis Chelsea:
    Assignment 1     3
    Assignment 2    19
    Lab Test         6
    Exam            40
    Grade            C
```

Note that the student names are printed in lexicographical order.

ITECH 7603

Advanced Programming
School of Science, Information Technology and Engineering

University of Ballarat
Learn to succeed

Allocated Marks: See Course Description

Due Date: See Course Description
Please refer to the Course Description for information relating to late assignments and special consideration.

Plagiarism:
Please refer to the Course Description for information regarding Plagiarism.

Assignment Submission:
Assignments must be submitted by the due date and your assignment should be completed according to the General Guidelines for Presentation of Academic Work (http://www.ballarat.edu.au/aasp/student/learning_support/generalguide/).
The following criteria will be used when marking of your assignment:

- successful compilation
- successful completion the required tasks
- adherence to the guidelines provided
- quality of code that adheres to the programming standards for the Course; including:
    - comments and documentation
    - code layout
    - meaningful variable names

You are required to provide the following documentation, contained in an appropriate folder:

- a statement of what has been completed and acknowledgement of the names of **all** people (including other students and people outside of the university) who have assisted you and details on what parts of the assignment that they have assisted you with
- a printed copy of your code (this may be included as an Appendix). Please include this because we can provide more feedback this way

In addition to submitting a printed copy of your written report into your tutor's assignment box, you should also submit the following using Moodle:

- an electronic copy of your code. Zip up the project itself and submit with the name <surname><idNo>.zip
- a copy of your report (surnameStudentIDAssign1.doc) including program code.

ITECH 7603

Advanced Programming
School of Science, Information Technology and Engineering

University of Ballarat
Learn to succeed

Student ID: _____

Student Name: _____

Marks awarded/deducted based upon considerations outlined on previous page

Tasks	Max marks	Awarded/Deducted
1.2	1	
1.3	1	
1.4	1	
1.5	2	
1.6	2	
2.2	1	
2.3	2	
2.4	1	
2.5	2	
2.6	2	
3.2	2	
3.3	2	
3.4	1	
Total	**20(-2)**	