LaSalle College
Montréal

Course Number: 420-P16-AS
Course Title: Structured Programming
Teacher: Quang Hoang Cao
Session: Autumn 2015
Group: 7150

Final Project

Evaluation Weight: 30 % (out of the total mark for the course)
Due Date : 3 December, 2015 (Demo and Submission)

Objective

This final project helps you to *master the competency required for the course*. The project focuses on analyzing, designing, implementing and testing an application using a structured programming language C++.

Competency-code: To use a structured programming language (016S).

Elements of the competency

1. To analyze the situation (016W.1).
2. To develop the algorithm (016W.2).
3. To translate the algorithms into a programming language 016S.3).
4. To compile the program (016S.4).
5. To test the program (016S.5).

Relevant performance criteria

- Detection of compilation errors.
- Correction of compilation errors.
- Effective use of the environment’s operating system and debugging features.
- Development of tests to verify that the program is functioning properly.
- Correct interpretation of results.
- Appropriate debugging of the program according to the algorithm.

Critical thinking.
Autonomy, initiative.
Time management.
Analytic and synthetic.

Case Study

Design and implement an application using C++. The application will be used by

Teacher: Quang Hoang Cao, for the following courses he is teaching in this session (Autumn 2015).

Course Number	Course Title	Group
420-P16-AS	Structured Programming	7148
420-P16-AS	Structured Programming	7150
420-P16-AS	Introduction à la programmation structurée	7151
420-P34-AS	Advanced Object Programming	7256

The following Evaluation Chart is applied to all the courses:

Evaluation Chart

Components	**Weight**
Project	30%
Midterm Exam	30%
Final Exam (Evaluation)	40%
Total	100%

Requirement: The teacher must login with the following valid user name and password:

Username: 5257
Password: quanghoang

The application allows the teacher to perform the following operations:

1. List all the courses he/she is teaching
2. Enter students' grades for a given course
3. Search a student's grades by Student ID/First Name/Last Name/First Name and Last Name
3. Sort the student list by Student ID
4. List all the students' grades for a given course
5. Quit the application

Option 1: Enter a student's grades

The teacher has to enter the following data:

- Student ID (7-digit number, unique value)
- First Name
- Last Name
- Project Grade
- Midterm Exam Grade
- Final Exam Grade
- Password

To keep it simple, username is Student ID.

Note: A grade entered must be between 0.0 and 100.0 included.

Option 2: Search a student's grades by Student ID / Firstname/Lastname/ Firstname and Lastname

Example : *Search by Student ID*

The teacher enters a Student ID; the following output will be displayed:

```
                    Course Number: 420-P16-AS
                Course Title: Structured Programming
                    Instructor: Quang Hoang Cao
                       Session: Autumn 2015
                           Group: 7148

                    ASSESSEMENT SUMMARY

Student ID   Student Name          Project    Midterm    Final Exam    Final Result
=======      ==========            =====      ======     ========      =========

1234567      Mary, Brown             80.0       80.0          80.0            80
```

Option 3: Sort the student list by Student ID in ascending order

Option 4: List all the students’ grades

List all the students together with grades in the following format:

```
                    Course Number: 420-P16-AS
                Course Title: Structured Programming
                    Teacher: Quang Hoang Cao
                      Session: Autumn 2015

                     ASSESSMENT SUMMARY

Student ID  Student Name        Project   Midterm   Final Exam   Final Result
=======     ==========          =====     ======    ========     =========

1234567     Mary, Brown         80.0       80.0       80.0            80
1234568      John, Brown         90.0      90.0       90.0            90

..........................................................................
..........................................................................
    Number of students who passed the course:
    Number of students who failed the course:
```

Important Note: *The final grade must be rounded, for example 76.3 -→ 76;*
65.6 →66

Option 5: Quit the application

The teacher selects this option to exit the application. Confirmation message must be shown to the teacher.

Student: The application also allows the students to perform the following operations:

1. List all the courses he/she is taking
2. View the grade for a given evaluation component (Midterm Exam, Final Project or final Exam) related to a course he/she is taking in this session (Autumn 2015)
3. List all the grades for a given course
4. Quit the application

Requirement: Each student must login with *valid username and password* created by the teacher.

Source Code Requirements:

1. Short description of the program.
2. Your name and due date.
3. Useful comments.
4. Using structures.
5. Using an array of structure.
6. Using functions.

   For each function, you have to write down the following:

   Purpose: A short description of the function
   Parameter List (if required): Input or output
   Value returned (if required): Specify the type.
   Where to use the function: Specify clearly the function will be used in another function or in the function main ().

7. Using pointers

   Examples

   - Pointer to a structure
   - Pointer to an array
   - Pointer to an array of structure
   - Pointers as input parameters of a function

8. Data validation

Project Marking Scheme

Certainly, your application must satisfy all the functional requirements specified in the project. Pay attention to the following evaluation criteria.

Element(s) of the competency	Evaluation Criteria	Requirements	Weight
1,2,3,4 and 5	1. Data validation	Be sure no invalid data has been entered into the application	**10**
1,2,3,4 and 5	2. User Interface	▪ Easy navigation ▪ Clear instructions ▪ No spelling mistakes ▪ Elegant error messages ▪ Complete implementation of the project's requirements ▪ Hints: Follow the techniques I presented in class	**15**
1,2,3,4 and 5	3. Source Code (See the Source Code Requirements)	▪ Useful comments ▪ Naming Conventions: Consistent ▪ Application of the concepts of structured programming discussed in the class ▪ Organization of the source code into 3 files : - Declaration file (Header file .h) - Implementation file (.cpp) - Application file (.cpp) ▪ Complete implementation of the project's requirements ▪ Hints: Follow the techniques I presented in class	**55**
Critical thinking. Autonomy, initiative. Time management. Analytic and synthetic.	4.Project Documentation and Demo	▪ Final Project in Word Format (Hints: Read the document posted in LEA)	**20**
	Total		**100**

Important Notes and Recommendations

Penalty

- ❖ If the program gets errors (Syntax or Run-time) when running, **20 points** will be deducted from the total points.
- ❖ Logical errors will result in **0 point** for the code part related.
- ❖ Late submission and absent from the project demo automatically results in **zero (0)** for the final project.

Recommendations

- ❖ For any question concerning the final project, do not hesitate to contact me via MIO or come to see me in the teacher's room (3209) during my office hours.
- ❖ You should have a good plan to carry out the final project.

Language Evaluation

Written language

The teacher is responsible for identifying spelling and grammar errors and for deducting the corresponding number of marks for any given summative evaluation. Below is the % – based on language requirements – that can be deducted from the grade of each summative evaluation:

• Penalty of up to 5%.