LaSalle College
Montréal

Project C++: 30%

Course Identification

Name of programs – Codes:	COMPUTER SCIENCE TECHNOLOGY (420.A0)
Course title:	**STRUCTURED PROGRAMMING**
Course number:	420-P16-AS
Group:	7148
Teacher's name:	Maftei Mihai
Duration:	Before 2016-12-02 15:30
Semester:	Fall 2016

Standard of the Evaluated Competencies

Statement of the evaluated competency – Code

To use a structured programming language – 016S

Evaluated elements of the competency

1. Adapt algorithms to the constraints of the programming language.
2. Translate the algorithms into a programming language.
3. Compile the program.
4. Test the program.

This project is calculated on 100 points distributed as follows:

Question 1: Designing the algorithm / flowchart	For a total of 10 points
Question 2: Translating algorithm into a C++ program	For a total of 10 points
Question 3-1 : Create a structure	For a total of 5 points
Question 3-2 : Run and display the output	For a total of 5 points

Development of a C++ application using arrays, structures, functions

Question 4 : Display the student management menu	For a total of 5 points
Question 5 : Input student information using structure	For a total of 10 points
Question 6 : Search for a student by student number	For a total of 20 points
Question 7 : Display the student list	For a total of 15 points
Question 8 : Exit the application	For a total of 5 points
Question 9: Document of application project	For a total of 15 points
	TOTAL : 100 points

Display the student management menu

Instructions

Read each of the following questions carefully, and then write your answers to the theory questions in the spaces provided. For the programming question, you have to use Visual Studio 2012 and C++.

SECTION I: Algorithm/code in C++ (30 points)

Question 1 (10 points)

Create the algorithm (pseudo-code) that will ask the user to choose what kind of conversion he/she likes to do (pounds to kilograms or kilograms to pounds).

There are **2.2046** pounds in a kilogram, **1000** grams in a kilogram, and **16** ounces in a pound.

If the first option was selected, enter a weight in pounds and ounces and outputs the equivalent weight in kilograms and grams or to enter 0 if wants to find out the max, the min and the average of the weights the user enter as far.

If the second option was selected, enter a weight in kilograms and grams and outputs the equivalent weight in pounds and ounces or to enter 0 if wants to find out the max, the min and the average of the weights the user enter as far.

Include a loop that lets the user repeat this computation for new input values until the user says he or she wants to end the program (the 3th option of the main menu).

Question 2 (10 points)

Translate the algorithm (in question 1) into the C++ Language. Use functions for the subtasks. (Implements the function prototype from question 1)

Question 3 (10 points)

1 Given the following data structure, declare a structure named **Courses** with the appropriate fields (**5 points**).

Course number	Title	Hours per week	Session
420-P16-AS	Structured Programming	6	Fall 2016

2 Write a code program in C++ that allows entering all the courses data that you have for current session at compile-time using the structure from previous point. Present the output you obtain in the documentation of the project. (**5 points**)

SECTION II: Programing in C++ (70 points)

Student Management Application

Create an application that keeps track of students, knowing that a **student** is defined by: a number, a first name, a last name, an email and a phone number where first name and last name are defined as a structure **Person** and phone number is a structure **Phone** with the following properties:

1- international code : 1digit
2- country code : 2 digits
3- regional code : 3 digit
4- home code : 7 digits

Question 4 (5 points): Display the student management menu that returns the chosen option by the user.
Question 5 (10 points): Create the students (add yourself as student).
Question 6 (20 points): Search for a student by his number.
Question 7 (15 points): Display the students list.
Question 8 (**5 points**): Exit the application.

The menu is as follows:

Student Management Application

1. Create and add a student.
2. Search for a student by number
3. Display the student list
4 Exit the application

Enter your choice: _

Include the execution output (as a print screen or as a comment into the documentation)

Question 9 (**15 points**): Documentation of the project with algorithms, flowcharts, print screens, comments of the application(s).

Identify yourself, save all your files, and upload your work before the time limit by LEA of Omnivox.

Thank you