EEE102 C++ Programming and Software Engineering II

Assessment 3

Assessment Number	3
Contribution to Overall Marks	15%
Submission Deadline	Monday, 3-April-2017, 23:55

How the work should be submitted?

SOFT COPY ONLY !

(MUST be submitted through ICE so that the TAs can run your programs during marking.)
Make sure your name and ID are printed on the cover page of your report.

Assessment Overview

This assessment aims at testing some basic concepts of C++ programming and initiates the routine of code development using the software development process (**SDP**), namely the five main steps of the software development process:

1. Problem statement: formulate the problem.
2. Analysis: determine the inputs, outputs, variables, etc
3. Design: define the list of steps (the algorithm) needed to solve the problem.
4. Implementation: the C code has to be submitted as a separate file. Just indicate here the name of the file.
5. Testing: explain how you have tested and verified your C program.

You will need to apply this methodology to each one of the following simple exercises.

What should be submitted?

A short ***report*** (up to a few pages of texts plus C++ source codes) detailing for all the questions of the assignment. The answer for each question should follow the SDP method:

a) SDP steps 1 to 3. (30% of the total marks for that question)
b) SDP step 4 (implementation): your C++ source code including the comments. (50%)
c) SDP step 5 (testing): you will explain how you have tested the correctness of your C++ program and will include some sample runs of your C++ Programs. (20%). **Testing result must be shown by screenshot.**

The report in Microsoft Word format **(.DOCX file)** and **C++ source code (with comments)**, for all questions should also be zipped into ***a single file***. (It is a good practice to include comments in your code stating the aim of the program, what are the inputs, what are the outputs, which algorithm is used, who is the author and so on.)

EXERCISE 1 (5 POINTS OUT OF 15)

Define a class called **`Month`** that is an abstract data type for a month. Your class will have one member variable of type **`enum`** to represent a month (January, February, and so forth). Include all the following methods:

1. A constructor to set the month using the first three letters in the name of the month as three arguments;
2. A constructor to set the month using an integer as an argument (1 for January, 2 for February, and so forth);
3. A default constructor;
4. An input function that reads the month as an integer;
5. An input function that reads the month as the first three letters in the name of the month;
6. An output function that outputs the month as the whole name of the month (January, February, and so forth) ;
7. and a member function that returns the next month as a value of type **`Month`**.

Test your class appropriately.

EXERCISE 2 (10 POINTS OUT OF 15)

Design a new class to represent a date like follows:

JAN - 25

The class **Date** contains two data members, denoting the 3-letter abbreviation of the month and the date.

Part 1: Fundamental requirements:

1. Write the constructors, destructor and output function members of this class;

Part 2: Using "operator overloading" to realise the following tasks:

2. Define the assignment operator "=" to assign the value of one Date object to the other;
3. Define the comparator ">" "<" and "==" to compare two dates;
4. Define the unary sign "++" to add one day to a date object:

```
Eg:  Date dToday(MAR, 20);        // value of dToday is Mar-20
     ++dToday;                    // value of dToday changed to Mar-21
```