EEE102 C++ Programming and Software Engineering II

Assessment 5

Assessment Number	5
Contribution to Overall Marks	15%
Submission Deadline	Sunday, 07-May-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 (15 POINTS OUT OF 15)

You are asked to program a card game called Blackjack, also known as Twenty-one. Blackjack is a comparing card game between a player and dealer (In a real game, multiple players are allowed but only one player is considered in this assignment). One may win a game in one of the following 3 ways:

1. Get 21 points on the player's first two cards (called a "blackjack"), without a dealer blackjack
2. Reach a final score higher than the dealer without exceeding 21 or
3. Let the dealer draw additional cards until his or her hand exceeds 21.

An ace can be counted as 1 point or 11 points. Face cards (kings, queens, and jacks) are counted as ten points. All other cards are counted as the numeric value shown on the card. After receiving their initial two cards, the player can place his initial bet. In any given round, the player has the following options

1. Hit: Take another card from the dealer
2. Stand: Take no more cards.

The player’s decision also includes whether or not the player wants to increase the bet and how much he wants to increase if yes. The dealer must hit until the cards total 17 or more points. Scoring higher than 21 (called "busting" or "going bust") results in a loss. A player may win by having any final score equal to or less than 21 if the dealer busts. You may write an algorithm for the dealer to make a decision or simply make the decision for the dealer when it is the dealer’s turn to make the decision.

To play this game, every player must set up an account first with positive balance say the starting deposit is 100. This means that you have to set up a data base (a file) which records the players’ information, for example, name, gender and account balance etc.. The player’s account balance will be increased by twice of the size of the bet if he wins. Otherwise, the player losses his bet and he is not allowed to continue his game if the balance becomes zero. You should design and implement at least 3 classes (objects).