EEE101 C Programming and Software Engineering 1 – ASSESSMENT 3

Assessment Number	3
Contribution to Overall Marks	35%
Issue Date	12/11/2018
Submission Deadline	17/12/2018 at 0900 (9am)

Assessment Overview

This assessment aims to test your ability to design a modular program which makes use of files and structures. Application of the software development process (**SDP**) is also under assessment; the five main steps of the software development process are:

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.

EXERCISE

In undertaking Assessment 2 you have developed a game of rock, paper, scissors played against the computer. An entertainment company would now like you to develop this software further to provide players of the game with their own secured account in which their game history is recorded and stored in a file. There are a number of requirements as described in the following section.

Program Requirements

- Your program should provide users with the ability to create an account. An account should be a structure type variable containing: a username, a password and a record of the game history.
- The game history should be stored in a two-dimensional array and contain the following details as a minimum:
  (i)   Number of rounds in a game
  (ii)   Number of player wins
  (ii)   Number of computer wins
  (iii)   Number of draws
  (iv)   Whether or not the game was overall a win draw or loss.
- All of the accounts should be stored in a data file and accessed by the program.
- Once a user is logged on to the game they should be able to:
  (i)   Start a new game
  (ii)   Review their game history
  (iii)   Clear their game history
  (v)   Logout
- Normally, the use of global variables is not allowed. Any use of global variables must be fully justified in your report..

Ideas

Note: the following are only to provide you with ideas of how to implement the required functionality. They do not represent the "best" or only ways to implement the functions.

All of the player accounts are structure variables and will be stored in a file. The login process can be achieved using a single structure variable, into which each account can be read from the file one by one, each time checking the username until you find the correct players account. The gaming operations can then be performed using that single structure variable. When the player finishes only this one structure needs to be written back to the file for saving.

Alternatively, you can create an array of structures in the program and read the whole file into the array, then search the array. The game can be played using the correct element of the array of structures. When the player finishes, the whole array can be written to the file.

Try to create your own functions to simplify the programming task.

Consider how your program should function when there is an invalid input.

To make your program more interesting for the user think about what kind of information they may like to access for example overall win percentages.

What should be submitted?

You should submit the followings:

1) Your C source code (.c) file and a short report (up to a few pages of text):

   a) SDP steps 1 to 3 in the report (Problem statement(10%) + Analysis(10%) + Algorithm Design(10%) + Report Quality(10%)) (40%)

   b) SDP step 4 (Implementation(35%) + Robustness(10%)): your C source code including the comments. (45%). **Do not** paste the code into your report just submit the .c source code file and write the file name in your report under the implementation section.

   c) SDP step 5 in the report (testing): you will explain your testing methodology including: what you wanted to test, how you have tested it and the outcome of your tests. (15%). Note: you do not need to include screenshots of all testing results. You can include some to show correct operation and or failures, but avoid including pages and pages of screenshots for every test you perform.

   Please refer to the file "EEE101 Marking Guidelines Assignment 3" for the detailed marking scheme.

2) The report in Microsoft Word or pdf format and C source code of your implementation should be zipped into a single file, i.e. the zip file will contain 2 files, one document and one source code. (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.)

The naming of Report (.docx or .pdf), Source Code (.c) and Compressed file (.zip or .rar)

- StudentID_LastName_FirstName_AssignmentNumber.docx or .pdf
- StudentID_ AssignmentNumber.c
- StudentID_LastName_FirstName_AssignmentNumber.zip or .rar

For example

- 123456789_Einstein_Albert_3.docx
- 123456789_3.c

Zipped together into:

- 123456789_Einstein_Albert_3.zip

How the work should be submitted?

Should be submitted electronically through ICE so that the marker can run your programs during marking. Feedback and your grade will also be given through ICE.

Remember that you are responsible for ensuring that your C code will run in Visual Studio 2013 and that if it does not without documentary explanation you may get a 0 mark for your implementation.