EEE101 C Programming and Software Engineering 1 – ASSESSMENT 4

Assessment Number	4
Contribution to Overall Marks	25%
Issue Date	9/11/2015
Submission Deadline	23/11/2015 at 0900 (9am)

Assessment Overview

This assessment aims at testing some concepts of C programming and the routine of code development using the software development process (**SDP**) presented in Lecture 1, 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 exercises.

EXERCISE 1 (5 POINTS OUT OF 10)

Write a **C** program that simulates a simple ATM machine. The program should be able to perform the following functions:

Contain 5 accounts. Each account should have an account number, an associated passcode and the current account value. These should be stored in a 2-dimensional array (matrix) in function main(); one row in the 2-dimensional array represents one customers data.

A user should be able to login using their account number and password. The login check should take place outside of function main().

Following successful login, users should be given the option to:

- View their balance
- Withdraw money
- Deposit money
- Quit

On selection of an option, the program should jump to a function (outside of function main()) designed to perform the desired operation. The program should only exit when the quit operation is selected.

Programming Constraints

ALL variables used **MUST** be **LOCAL** to their function.

EXERCISE 2 (5 POINTS OUT OF 10)

Write a C program that allows a user to play the game of “find the pairs”. The program should be able to perform the following functions:

- Fill a 4 x 4 matrix with 8 randomly placed pairs of random ASCII characters.
- Display the randomized matrix for a short time
- Allow the player to make guesses selecting two matrix elements and then displaying their contents.
- If the player guesses correctly the elements should remain showing, otherwise they should be hidden again.
- The player should be able to quit the program at any time by pressing q.
- On successfully finding all 8 pairs, the game should end displaying the number of guesses it took.

Programming Constraints

You will need at least one matrix to contain the randomized pairs and one to display the player’s progress. These matrices should be declared inside function main() as local variables.

Ideas

Try to keep your program as flexible as possible e.g. use constants to define matrix sizes so that these could be easily changed.

Use a generic function to display each matrix and simply pass the matrix you would like to display to the function.

What should be submitted?

You should submit the followings:

1) A short report (up to a few pages of text plus C source codes) detailing for each question:

   a) SDP steps 1 to 3 in the report (Report + Specification + Analysis + Algorithm Design) (40%)

   b) SDP step 4 (Implementation + Robustness): your C source code including the comments. (45%)

   c) SDP step 5 in the report (testing): you will explain how you have tested the correctness of your C program and will include some sample runs of your C Programs. (15%).

   Please refer to the file "EEE101 Marking Guidelines Assignments 4-6" on ICE for a detailed marking scheme.

2) The report in Microsoft Word or pdf format and C source code of your implementation for each question should be zipped into a single file, i.e. the zip file will contain **4** files, two reports and two source codes. (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 (.doc or .pdf), Source Code (.c) and Compressed file (.zip, or .rar)

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

For example
- 10115085_Zhang_Hanqing_4-1.docx
- 10115085_4-1.c

Zipped together into:
- 10115085_Zhang_Hanqing_4.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.