SENG1000 – C/C++ PROGRAMMING

MAJOR ASSIGNMENT 1A - DEAL OR NO DEAL

This assignment is designed for students who already know C/C++ (or a similar language) and want more of a challenge and want to work ahead (WAY ahead).

I mean it. This assignment is NOT for someone who does not have much experience with programming.

It is expected that you will do this assignment without asking for help with programming in C or C++. I am not willing to teach you material that you don't know in advance of when it's taught to everyone else (it's not fair to the other students who are doing the simpler assignment and have very little experience).

OVERVIEW

Write a console program in C that plays the "Deal or No Deal" game.

GENERAL COURSE OBJECTIVES ADDRESSED IN THIS ASSIGNMENT

- Analyze problems.
- Design computer code from written problems using structured design methodology.
- Design computer programs for modularity, maintainability, and ease of reuse.
- Troubleshoot and debug problems.
- Many other course objectives are covered by this assignment.

ACADEMIC INTEGRITY AND LATE PENALTIES

- Link to Academic Integrity Information
- Link to Late Policy

EVALUATION

- The evaluation of this assignment will be done as detailed in the Marking lecture from Week 2.
- If you do this assignment well, you **may** be eligible for a 50 mark bonus (out of 100). This **only** applies if you do the assignment well.
- This assignment **cannot** be resubmitted..
- It is vitally important to note that you **give the user a good user experience** in order to get a good mark.
    - Vital in this is providing the user with **all** of the information that they need in order to make an easy choice of suitcase and an intelligent decision on the banker's offer.
- If you hand in both Assignment 1 and 1A, Assignment 1A **will not be marked**. I mean it.

PREPARATION

- Know the C language very well.

- Be aware of all Course Requirements from the Course Notes.

REQUIREMENTS

GENERAL OBJECTIVE

- Simulate the TV game show *Deal or No Deal* as a text console-based program.
- An online graphical version of this television game show can be found at https://zone.msn.com/gameplayer/gameplayerHTML.aspx?game=dealornodeal&instance=default.
    - There is no guarantee that the above link will be valid for you when you try it out. If it is not, you can find videos that show the game being played on TV.
- Keep in mind that there are some additional features in the online game that don't exist in the TV game that should not be implemented (specifically, the timeout on user input and the "counteroffer").

GAME DETAILS:

- There are 26 "suitcases", each containing an amount of money (amounts listed in the online game). No two suitcases contain the same amount. Each game has the suitcase amounts assigned randomly.
- The contestant selects a suitcase number, which then becomes their suitcase.
- After picking the case, the contestant selects six of the remaining suitcases. The amounts contained within those cases are revealed, one at a time.
- After the sixth suitcase is revealed, The Banker makes an offer to buy the contestant's case for a certain amount, calculated by the program based on the cash amounts still in play. If the contestant accepts the deal, the game is over and the contestant's suitcase contents are revealed.
- If the contestant turns down the offer, they must then select another five of the remaining cases to reveal. The Banker proposes another deal, which is accepted or rejected.
- Play continues, revealing a lesser number of suitcases each time, until a deal is accepted or all suitcases are revealed.
- When there are only two suitcases left without a deal being accepted, the player is given the option of switching their suitcase before revealing the contents of the two cases.
- The graphical and sound aspects of the online game should not be replicated. What is desired in this assignment is replication of the gameplay.

BANKER'S OFFER:

- In determining a fair amount for The Banker to offer, you are allowed to average the remaining amounts and multiply that value by a random value between 75% and 110%. Alternatively, if you find a better way of doing it, you can use it, as long as it truly is "better" (yes, this is subjective).
- If you choose to use a formula or algorithm that you found on the Internet, you absolutely **must** cite where you got it in a comment near where you implement it. The method of citing is found in the SET Coding Standards.

COMMENTING AND INDENTATION:

- You must have full comments as described in the Commenting lecture.

OTHER REQUIREMENTS:

- Use the SET Coding Standards (found on eConestoga) that are relevant.
- You must adhere to the Course Requirements as found in the Course Notes.
- All variables must be declared within functions (i.e. it must not use global variables (covered later in the Scope and Style lecture)).
- Appropriate programming style as discussed in lecture and in the Course Notes must be used.
- It is assumed that you will adhere to all course requirements detailed in the Course Notes readings so far in the course. **This requirement holds for all subsequent assignments.**
- Do not clear the screen (this requirement is true for all assignments).

CHECKLIST REQUIREMENTS

- Create a requirements checklist. This should contain the specific requirements from this assignment as well as any relevant requirements that have been covered in lecture or that are found in the SET Coding Standards or SET Submission Standards. Do it in whatever form you wish. Hand in your completed checklist in PDF form as checklist.pdf. Not having this checklist will result in a cap of 80 on your mark.

FILE NAMING REQUIREMENTS

- You must call your source file m1a.cpp.
- You must call your checklist checklist.pdf.

SUBMISSION REQUIREMENTS

- Do not hand in any other source files besides those mentioned in the File Naming Requirements.
- Follow the instructions in the SET Submission Standards and the lecture on Submitting Assignments to submit your program. Submit both files in one submission to the correct Assignment folder.
- Once you have submitted your files, make sure that you've received the eConestoga e-mail confirming your submission. Do not submit that e-mail (simply keep it for your own records until you get your mark).

ADDITIONAL INFORMATION

- Helpful Hint: You will want to use srand(), rand(), fgets(), sscanf(), printf(). You are allowed to use the C++ language features that we learn in this course (cin, cout, string class object) but none beyond that.
- It is important that you pay attention to the bolded items within these requirements.