Western University

CS 1037A

Assignment 1

Instructions: This assignment should be done individually. The deadline is Friday October 10, 2014 at 11:55 pm. Please refer to the course outline (**Assignments Submissions and Late Assignments Policy Section)** for the late penalty. Each student must submit to OWL a folder (zipped file) that contains all source code files and a ReadMe.pdf file. The ReadMe file should explain in details how to run the program for Question 1, Question 2 and Question 3. All files should work under Visual Studio C++. It is very important that you comment your code thoroughly. Marks will be deducted if the code is not commented properly.

Question 1: Guess My Number (35 points)

Write a program in C++ to:

1. Declare an array of length 25 (you can create it on the stack or on the heap).
2. Store the array with integer random numbers between 1 and 50.
3. Generate a random number between 5 and 55.
4. Use the binary search algorithm to search for the number generated (in part 3) in the array. If not found, repeat (part 3) until this number matches a value stored in an element of the array.
5. Display a message for the user asking to guess the value of the chosen element (the user has to guess a number between 1 and 50).
6. Check the valued entered by the user with the actual value, then:
    a. If the value entered by the user is greater than the actual value, then display a message "choose a smaller value"
    b. If the value entered by the user is smaller than the actual value, then display a message "choose a larger value"
    c. If the value entered by the user is equal to the actual value, then display a message "Bravo!! You guessed my number in ***x*** attempts" (where ***x*** is number of attempts the user took to guess the number). After that, ask the user to guess the location of the guessed (actual) number. The location should be between 0 and 24. In case more than one array element contain the same actual number, then the user has to guess the first array element.
        i. If the user enters an incorrect location, then display messages similar to 6.a and 6.b.
        ii. If the user guesses the correct location, then display a message "Bravo!! You guessed my number in ***x*** attempts and the location in ***y*** attempts" where ***y*** is the number of attempts the user took to guess the location).
        iii. Display a message "Game Over, do you want to play again? Press *y* for yes or any key for no.

1. If the user presses *y*, then shuffle the array and start again from part 3. Please note in this case, the program should shuffle the array created in part 1 and part 2. Do not create a new array with new random numbers.
2. If the user presses any key other than 'y', then exit the program

Question 2: Stock Class (35 points)

Write a class called *Stock* to save information about items in a retail store. The class should have:

1. Private member variables:
    a. *itemNumber* (*integer*)
    b. *quantity* (*integer*)
    c. *cost* (*double*) // this is per-unit cost of the item
2. Public member functions:
    a. Default constructor (to set all member variables to 0)
    b. Second constructor (this should accept an item's number, quantity and cost as arguments. Calls other class functions (setters) to copy these values into the appropriate member variables.
    c. *setItemNumber* (accepts an *integer* argument and copies it into the *itemNumber* member variable)
    d. *setQuantity* (accepts an *integer* argument and copies it into the *quantity* member variable)
    e. setCost (accepts a *double* argument and copies it into the cost member variable)
    f. getItemNumber (returns the value in itemNumber)
    g. getQuantity (returns the value in quantity)
    h. getCost (returns the value in cost)
    i. getTotalCost (Computes and returns the total cost which is *quantity***cost*)

Write a program in C++ to implement the above requirements. Your program should be split into three different files; one that contains the class, one for the functions' implementation and one is the client program. Moreover, the program should validate the user inputs to ensure that the negative values are not accepted for item number, quantity or cost. The validation must be done through two Boolean functions; one that validates integer arguments (non negative) and one that validates floating-point arguments.

Question 3: Movie Data (30 points)

Write a program in C++ that uses a structure named MovieData to hold the following information about a movie:

Title (string)
Director (string)
Year Released (integer)
Running time (integer in minutes)

When a MovieData variable is created, the values of the above four member variables should be set through a constructor. The program should create two MovieData variables (give any appropriate values to the four member variables) and pass (by value) each one in turn to a function called *displayMovie* that displays the information about the movie in a clearly formatted manner.