ICT 710 Assignment 4: Arrays, Strings, and Dates (25%)

Due: See course outline for specific date

Instructions

In this lab you will practice working with one-dimensional arrays, strings, and dates. Also, you will continue working with methods.

Your starting point is Lab 2 solution. You may use your own or request a sample solution from the instructor (send an email).

Most of the functionality will be the same as in Lab 2. What changes is the way quiz marks are collected and processed. Rather than having 3 quizzes, your application will be able to process any number of quizzes, up to 10. Like in Lab 2, the lowest quiz mark will be dropped, and the remaining quizzes will count for 30% of the course grade.

All the quiz marks will be stored in an array and finding the lowest value and processing the remaining quizzes will be done by processing the array.

Also, finding the lowest quiz mark will be coded within a separate method.

Please refer to the rubric at the end of this document for evaluation details.

Requirements

1. Declare on the form level an int constant MAX with value 10. This will represent the maximum allowed number of quizzes.

2. Also on the form level, create an array that will hold the quizzes marks. It should have length MAX. Declare a variable of type int that will keep track of the actual number of quizzes and initialize it to 0.

3. In the design of the form, remove labels and text boxes for the individual quizzes, and replace them with one label and one textbox where each quiz mark will be repeatedly entered.

4. Create another button Add. The Click event procedure for this button stores the current quiz mark from the text box to the first unused element of the array, and increments the actual number of quizzes. The button Add should become disabled after the maximum number of quizzes has been added. (Hint: for testing purposes, change MAX to a smaller value like 3 or 5 to avoid having to enter a lot of data).

5. Create public method FindLowest that takes two parameters: an array of numbers of the same type as the quizzes array, and an int value that indicates the actual number of elements in the array. The method returns the lowest value in the array. The names of the parameters and local variables should be generic enough to enable using this method to find the lowest element of any array of the same type without causing naming confusion.

6. Call your method FindLowest, passing the array of quizzes marks and the actual number of quizzes and use the returned value in your further calculations.

7. Calculate and display the course mark and letter grade using the same logic as in Lab Assignment 2.

8. Validate all user inputs (quiz marks, midterm and final) using the same set of functions that check if the input is present, if it is a correct number, and if its value is between 0 and 100. Display meaningful error messages that tell the user exactly what is incorrect.

9. Display in the top right corner of the form the current date and time, up to a second. These values should be updating every second (Hint: you need a Timer control).

Zip the entire project folder of your application. Upload your zip file to the Assessment 2 Dropbox.

Evaluation

This assessment is graded out of 25 points and will be evaluated using the following rubric.

Learners may receive partial scores or a zero for unacceptable work.

Marking Component	Out Of
Form has appropriate controls and looks well organized	2
Array of quizzes and an int variable for the number of quizzes are created	2
Click event method for the Add button correctly adds another quiz mark to the array and updates the number of quizzes	4
Button Add is disabled when the maximum allowed number of quizzes is entered	1
Method `FindLowest` is defined and has parameters and return type as specified	3
Lowest quiz mark is identified correctly using method `FindLowest`	3
All calculations remain correct after the changes	3
Validating methods work correctly and display meaningful messages	3
Date and time is displayed in the top/right corner and it is updated every second	2
Code is written in good style and has ample comments	2
Total:	**25**