ECET 16400 Lab 8   Fall 2018

Lab 8 Instructions

- **Objective**:
  Build on knowledge acquired from pre-lab assignment.
  You will use controls and code to open and read a file's contents into an array, use looping logic to fill and display the array values. Use methods and method calls process data. Use appropriate decision structures with logical and relational operators to determine average, largest and smallest values in the array. Use formatting to output information in list box and label controls.

- **Lab 08 Program description:**
  - **Part 1**.
    Program Description
    - See text book page 468: Programming Problems 1. Total Sales
  - Part 1:
    1.1. Plan your program UI Design: (Use the Designer User Interface document)
    - You are to come up with an appropriate user interface: This is where you should draw out your design on a sheet of paper and identify each object (name your controls). There are no specific sizes or colors of fonts. You have read enough to this point to where you know how to change all these properties. You must create something that looks "professional".

    1.2. Plan the Event Handler Methods: use Pseudo code, flow charts, etc..
    - The application should allow the user to click on the "Get Sales" button. The "Get Sales" click event should have the following void method call.
      - **ProcessSalesFile Method:** accepts no arguments and does not return anything. This method will open the text file, fill the array, display the array elements in the list box, calculate and display the total sales
      - Variables must have appropriate data type and meaningful names
      - Exception handling must be included in this program: Use the try-catch
        - Appropriate error messages should prompt the user
      - Output to list box must be a formatted to two decimal places
      - Output to label must be properly calculated and formatted to two decimal places.
      - A Clear button should clear everything and set the focus back to the first input box
      - An Exit button will exit the user out of the program.

    1.3. Write the program code:
    - Create a new project. Call the project yourlastnameLab08_164. Rename the Form1.cs file to DisplaySales.cs (accept all changes if prompted). Change the text on the form to Sales Analysis.
    - Place and Set the properties of each object, as you have planned.
    - **Before you proceed:** Make sure all objects (controls) are properly re-named and properties assigned as instructed.
    - **Build your program to ensure all changes load correctly.**
    - Get the provided Sales.txt file and save in your projects debug folder
    - Working form the pseudo code, invoke and write each event-handler method.
    - When you complete the code, thoroughly test the project.
      - Make several "good" and "bad" input tests to ensure your program performs correctly and handles all user input.
        - Use books' examples
        - Output must be displayed as instructed
      - Once your program works as desired: get screen shots of your program when running. 1. Of the user interface when it first starts. 2. Of the program with result displayed when the user clicks the Get Sales button.

- **Part 2**.
- Program Description:
- Once Part 1 is working correctly modify your working program from Part 1 such that it meets the description for Programming Problems #2. Sales Analysis: Text Book page 468. Keep all the function of Part 1. Just add the code logic as follows:
  - PorcessSalesFlie method will then pass the array in method calls to the appropriate value returning method, and then ProcessSalesFile method will display the returned values in labels on the form.
  - The application should have the following value-returning methods called from the ProcessSalesFile method.
  - **GetAverage** – accepts a double array argument and returns the average of the values in the array
  - **GetHighest** – accepts a double array argument and returns the highest value in the array
  - **GetLowest** – accepts a double array argument and returns the lowest value in the array

- **Part 2 Planning:**

2.1. Plan your program UI Design:

- You are to modify your user interface. This is where you should modify your previous UI design to include and identify each new object (control)
- Add a appropriate labels to include control labels for the average, highest and lowest
  - Reference the books' examples

2.2. Plan the Event Handler Methods: use Pseudo code, flow charts, etc

- When the Calculate button is clicked all of Part 1 should perform as previously instructed. But now your program will call methods to perform the functions as described above. GetAverage, GetHighest, and Get Lowest methods

  Identify on your design sheet where, in your outline logic, you are to put the loop logic. Flow chart or pseudo-code this logic including the output.

Do not attempt to change your UI and code in your program until You understand what you are doing and how/where to add the additional controls and code logic.

2.3. Write the program code:

- Place each new object and Set their properties as you have planned.
- **Before you proceed:** Make sure all new objects (controls) are properly re-named and properties assigned as instructed.
- **Build your program to ensure all changes load correctly.**
- Working form the pseudo code, modify the appropriate event-handler method(s). And include the additional methods.
- When you complete the code, thoroughly test the project.

Your final application with do the following:

The application should allow the user to click on the "Get Sales" button. The "Get Sales" click event should have the following void method call.

ProcessSalesFile Method: accepts no arguments and does not return anything. This method will open the text file, fill the array, display the array elements in the list box, calculate and display the total sales, then pass the array in method calls to the appropriate value returning method, finally ProcessSalesFile method will display the returned values in labels on the form.

The application should have the following value-returning methods called from the ProcessSalesFile method.

GetAverage – accepts a double array argument and returns the average of the values in the array

GetHighest – accepts a double array argument and returns the highest value in the array

GetLowest – accepts a double array argument and returns the lowest value in the array

You will be graded on:

- User interface design: completeness, clarity, and ease of use. This would include labels telling the user what to do, buttons with descriptive text, easy to read, logical tab order, accept and cancel button assignments (pg179), and the first text box set to focus() when the form loads and other appropriate times.
- All Click Event Handlers contain only method calls (except for the clear and exit methods)
- Use of proper data types for calculations and visually appealing.
- Source code: this would include code organization, no extra control methods in code, complete code comments, objects re-named.
- Catch user input errors (for this lab you will need to use the try-catch) Accept and cancel button assignments
- Logical form object tab order
- Project works as instructed
- Correct files submitted in Canvas assignment

Submit your project via the appropriate Canvas Assignments tool. You will submit three files, (the executable .exe file, the Visual C# .cs file and the zipped project folder) and one design document with your screen shots included. If you have any problems locating these files or creating a zipped project folder ask for help during the lab session. You should demonstrate the program to the lab instructor before the leaving Lab session.

Note the closing date and time; as late submissions will not be accepted, nor any files sent as attachments via email.

See outlines on next pages:

```
private void ProcessSalesFile()
{
        // declare some variables


        1.Stream Reader Object
    Try
     {
        2. Open the file to read

                Loop through the file to get each value a put in array
        3. Close the file

        Once the array is filled with values from the file…
        Loop through the array to calculate the total
                In this loop also display each array value to the list box

        Once you have the total….display the total to a label.
    }
    Catch
   {
      // message to user
    }


}
```

PartII:

```
private void ProcessSalesFile()
{
        // declare some variables
   try
   {
        1.Stream Reader Object

        2. Open the file to read

                Loop through the file to get each value a put in array
        3. Close the file

        Once the array is filled with values from the file…
        Loop through the array to calculate the total
                In this loop also display each array value to the list box

        Once you have the total….display the total to a label.

        // Part2

        // double variable to hold returned value = call the GetAverage Method pass the array

        // double variable to hold returned value  = call the GetHighest Method pass the array

        //double variable to hold returned value = call the GetLowest Method pass the array
   }
   Catch()
   {
     // message to user
    }
}


//GetAverage method
Private double GetAverage(double [] iArray)
{
        //loop through the iArray to calculate the total and then
        //Calculate the average
        //return the average
}


//GetHighest Method
private double GetHighest(double [] hArray)
{
        //loop through the hArray to find the highest value in the array
        //return the highest

}




//GetLowest Method
private double GetLowest(double [] lArray)
{
        //loop through the lArray to find the lowest value in the array
        //return the lowest

}
```