ECET 16400 Lab 04   Spring 2018

The pre-lab assignment: Tutorials 4-4, 4-6, Review the Math Class in chapter 3
Review Chapter 4 Reading & Algorithm Workbench: 1-9

Lab 4:
Lab Objective: Build on knowledge acquired from pre-lab assignment.
You will use textbox controls and code to get and validate user input. Use math calculations and decision blocks to process user input and then use formatting to output information in label controls. Use list box or radio buttons and switch case decision structure to process user input. 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, tab order (pg175-177), accept and cancel button assignments (pg179), 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.
- Catching user input errors (do not use the try-catch, instead use the TryParse method as exampled in tutorial 4-4)
- Accept and cancel button assignments
- Logical form object tab order
- Project works as instructed
- Correct files submitted in Canvas assignments

Procedure:
Before you begin to program, <u>Read all instructions carefully</u> and be sure you understand what and why you are doing something. Have a clear picture as to how your completed program should look and perform.

For lab 4 you will program a solution for
An Ohms Law Calculator
Read Programming Description below
Do not begin to program until you complete <u>one</u> design process task

Design process tasks: Only one design sheet required!
1.) On a standard sheet of paper define your inputs, process and outputs then sketch your user interface. Be sure to include all the inputs, outputs and button(s) as described in the problem. For your process, either draw a decision flow chart or use pseudo code. For this lab assignment, you will not create a Properties document. Instead; for each object on your design document define the main properties such as object name and text (if any).
This document must be checked by the instructor before you begin to program. Once checked, reference this document as you program your solution. When complete you will submit this document with your assignment files.

Program Description: Ohms Law Calculator
You are to create a program that calculates ohms law (V= I * R) based on a user selection.

Read **Program coding Logic flow** and **Programming your solution** for further program requirements.

Program coding Logic flow:
Your program should allow the user to select the option of solving for current, resistance, or voltage. Use a list box as displayed below.
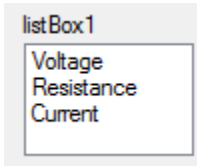
Your program should provide two text boxes and a calculate button. Once the user selects an option from the list box, the user will enter their two values and then click on the calculate button. Your program should display an appropriate calculated value to a label. Include any additional appropriate objects to display information/instructions, clear, exit button.

1. User will select an option from the list box
   a. Anytime the user selects an item from the listbox, appropriate labels on your form should update with appropriate text for the operands, operator, and output information.
   For example: If user selects Voltage then your text boxes should have labels that display as shown:
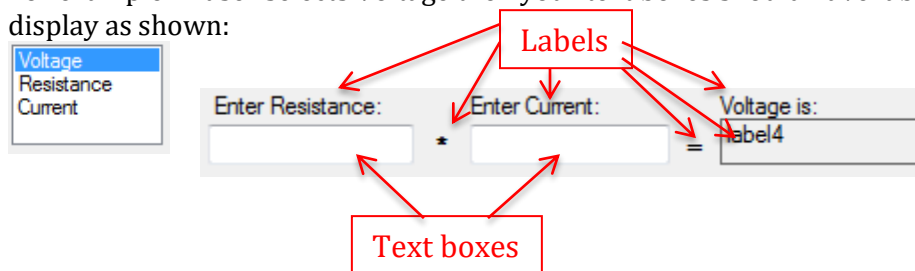
   If user selects Resistance then your text boxes should have labels that display as shown:
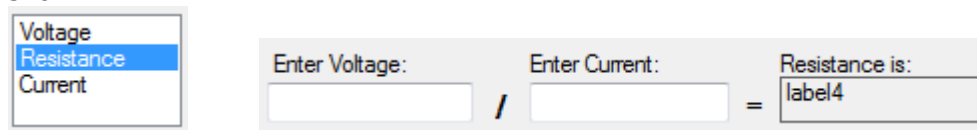
   You may arrange your labels and text boxes in any logical configuration.

2. Once the user makes their list box selection, they will enter values in two text boxes
3. Then the user will click on the calculate button
4. Once the calculate button is clicked, the appropriate calculated value will display in a label.
5. The user will have the option button to exit the program.

Code Outline:
On a sheet on paper, outline the following (do not just start coding!):
Your program will have at least one field (Global) variable declaration and **three** click event **handler methods**.

   1. **Exit button click event handler:**

This method will have code to close the program

2. **Listbox SelectedIndexChanged click event handler:**
   This method will have code to get the user's selection and update label information according to user selection.
   - Your code will assign the user's selection to a field (global) variable and then
   - Use a switch case to update the information labels according to user selection.
     - You must use a switch case decision clause (refer to in-class example)

3. **Calculate button click event handler:**
   This method will have code to validate the user inputs, determine appropriate calculation and display the output.
   a. Determine and then declare the local variables you will need in this event handler.
          i. For example: You will need variables to hold the user's two text box inputs:
             double operand1;
             double operand2;
          ii. You will need other variables as well.
   b. Next use an *if* with an *else* statement block to: Validate that an item has been selected from the list box.

```
if(listBox1.SelectedIndex != -1)
{
   // step 3.c. goes here
   // this is where you validate user input
}
else
{
   //send the user an appropriate message, reset focus, clear textbox;
}
```

   c. Nested inside the first `if` block (described in step 3.b.), your code should: Validate first user input
          Example on Page 237-238

```
if (double.TryParse( //stuff goes here))
{
   // step 3.d goes here

}
else
{
   //send the user an appropriate message, reset focus, clear textboxes;
}
```

   d. Nested inside the second `if` block (described in step 3.c.) your code should: Validate second user input.
          a. Inside this *if* block your code should: Process your user input.
                 i. You are required to use a switch case to calculate and output the appropriate result.
                 ii. You will switch on the same variable used in the first switch case.
                 iii. Caution: when solving for Current or Resistance do not allow a "divide by zero"
                 iv. Format your output calculation to two decimal places.

e. Make sure to write the `else` statements for each of your `if` blocks.
f. **You must show the Instructor or TA your nested "if statements" outline before you begin programming.**

Programming your solution:
1. Start Visual Studio and **Name your project YourName_Lab4**
2. **Select a convenient location to save your project**
3. Rename the **Form1.cs file name to ohmsLawCalc** (accept all changes if prompted).
4. Change the text on the form to Ohms Law Calculator
    a. Ensure your Form1's Text property updated and the Solution Explorer window displays the updated file names. If not….stop and ask for assistance.
5. **Build your program to ensure all files properly loaded**. (Build | Build Solution)
6. In the [Design] window create your user interface: be sure to properly re-name all objects (controls) placed on your user interface, and then write the necessary code.
    a. Use your design sheet as reference and create your user interface, then code your solution.
    b. Program in small chunks: Write user input validation first (if-else, try parse).
7. Note: for each lab assignment you are required to comment your code and include a commented header as described in Lab 2 instructions.
8. You are to assign accept and cancel buttons appropriately (page 179). And use proper tab order (pg 175-177).
9. User interface: be sure to properly re-name all objects (controls)
10. Your form should include a clear and an exit button. Format output to two decimal places
11. Run test data to ensure your program catches user entry errors and outputs correct results. Once your program can pass the "test data" you may add additional flare and design to the user interface. This could include a background picture on the form, color change, etc.….. But don't overdo it. Your form design should be easy for the user to read and use.
12. When you complete your project, if necessary, update your design document to match your completed project. Make notes on any changes.
13. Once your project is complete, if still in lab, demonstrate it to the instructor or a TA.

Submission:
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 which must include your code outline. If you have any problems locating these files or creating a zipped project folder ask for help during the lab session. If possible, 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 messages via email.