

ECET 16400 Lab 05 Spring 2018

Lab 05 instructions:

Lab 05 Pre-Lab:

- By completing the following, you will now learn how to input, process and output data using multiple controls.
- Reference and supporting material for **Part 1** of this lab:
 - Chapter 3:
 - Read all material and practice Chapter 3 Tutorials
 - **IMPORTANT:**
These Tutorials lay the foundation for: Reading input, data processing with controls and variables and formatting output. You must know this material to be successful in the next section.
 - **For success in this lab assignment:** Make sure you can answer the following questions:
 - Checkpoint questions:
 - Text box controls: 3.1 - 3.4,
 - Variables: 3.5-3.13,
 - Numeric Data types: 3.14-3.17,
 - Performing Calculations: 3.18-3.24,
 - Inputting and outputting numeric values: 3.25-3.29,
 - Formatting numbers: 3.30-3.34,
 - Exception Handling: 3.35-3.40,
 - Constants and scope of a variable: 3.41-3.46
 - Suggested Multiple Choice, True/False and Short Answer questions
 - Chapter 4:
 - Read Chapter 4
 - Read and practice Tutorials
 - **IMPORTANT:**
These Tutorials introduce key programming techniques used for decision structures, Logical operators and relational operators.
 - **For success in this lab assignment:** Make sure you can answer the following questions:
 - Checkpoint questions (all in chapter 4)
 - Algorithm Workbench questions (1 - 9).
- Reference and supporting material for **Part 2** of this lab:
 - Chapter 5:
 - Read Chapter 5
 - Read and practice Tutorials
 - **IMPORTANT:**
These Tutorials introduce key programming techniques used for decision structures, Logical operators and relational operators.
 - **For success in this lab assignment:** Make sure you can answer the following questions:
 - Tutorial 5-3
 - Checkpoint questions: 5.1 – 5.17
 - Algorithm Workbench questions (1 - 4).
 - Suggested Multiple choice, True/False and short Answer questions

Assignment Instructions

- **Objective:**
 - Part 1. Create a Power Calculator program based on the required reading and chapter tutorials.
 - Part 2. Modify your working program such that the Power Calculator program utilizes a loop.

ECET 16400 Lab 05 Spring 2018

The correct .cs file, correct .exe file, the correct zipped solution file and a design sheet word document that contains all of your UI design and completed pseudo code must be submitted to receive full credit!

If you are not sure if you have the correct files see the document “How to find your files” in the Resources Modules. It clearly shows which files you need to submit.

- **Lab 05 Program description:**

You are going to create a project that shows the user how much power is used based on two user inputs: **total work** (energy) that has been done and **time** (in seconds) for the total number of seconds this work has been done.

- For Electrical Energy, in physics, the following is true:

Watts are a measure of Power or how fast Energy is used.

Where P = Power(in Watts), E = Energy(in Joules), and t = time(in seconds)

and so... $P_{(W)} = E_{(J)} / t_{(s)}$ and $E_{(J)} = P_{(W)} * t_{(s)}$

- **Part 1:**

Program Description:

Essentially the user will input how much **total Energy** in joules used and how long **Time** in seconds. Your program will then output, to a label, the calculated result for **Power** in Watts.

For example: if the user inputs **1000 Joules** were used in **10 seconds**, the outputLabel would display similar to the following:

The Power result in Watts is 100W.

- 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.

- A Calculate button will take the user’s two inputs and display the Power calculation result in a label.
 - Variables must have appropriate data type and meaningful names
 - Exception handling must be included in this program with if-else try parse logic (advanced) or try-catch parse logic (average)
 - Ensure user input is only positive (non-negative) numbers. Do not allow letters or negative numbers in these input boxes.
 - Appropriate error messages should prompt the user accordingly
 - Calculated output must be a formatted text with result properly calculated and displayed in a label. Example output string: The Power result in Watts is 100W
- 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 yourlastnameLab02_164. Rename the Form1.cs file to WattsCalc.cs (accept all changes if prompted). Change the text on the form to Power Calculator.
- 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.**
- Working from the pseudo code, invoke and write each event-handler method.
 - Write small segments of code and test your program often!

ECET 16400 Lab 05 Spring 2018

- 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.
 - Once your program works as desired: get three 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 types in “good” data. And 3. Of the program with error message when the user types in “bad” data. You will include these screen shots at the end of your design sheet.
- **Part 2:**
 - Before you proceed: Make sure you have your screen shots from Part 1**
 - Program Description:

Once Part 1 is working correctly you are to modify your program such that your program will also utilize a loop to display in a List Box how much work (Energy in Joules) was done for each second.

 - For example, Using the Power result calculated (as exemplified in Part 1), the ListBox output would look similar to the following:
After 1 second the work done was 100 Joules.
After 2 second the work done was 200 Joules.
After 3 second the work done was 300 Joules.
After 4 second the work done was 400 Joules.
After 5 second the work done was 500 Joules.
After 6 second the work done was 600 Joules.
After 7 second the work done was 700 Joules.
After 8 second the work done was 800 Joules.
After 9 second the work done was 900 Joules
After 10 second the work done was 1000 Joules.
- **Part 2:**
 - 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).
 - 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.** **And**, also, a list will display in the List Box. For the sake of simplicity and for size considerations of your list box, edit your previous program to limit the total number of seconds entered to the range from 0 to 10.
Plan (write, or flow chart) your logic before you begin to code.

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 on your form 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 from the pseudo code, modify the appropriate event-handler method(s).
 - When you complete the code, thoroughly test the project.

ECET 16400 Lab 05 Spring 2018

Finally:

You must assign accept and cancel button assignments (page 179), Proper form tab order (page 175-176)

NOTE – COMMENT, COMMENT, COMMENT. Comment everything you program in the event handlers. Your code should have comments so anyone that looks at your program knows what it is doing and why. It doesn't have to be a long drawn out explanation unless it's absolutely necessary. The more concise you can make the comment the better, unless you deem it necessary to the viewer of your code to know in extra detail why you did what you did. Do not forget your commented header: as exemplified in Lab 02.

In addition to instruction requirements, 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
- Accept and cancel button assignments
- Logical form object tab order
- Project works as instructed
- Correct files submitted in Canvas assignments

Be sure to “save all” and run your program each time you make **any** changes to your program design and/or code, including comments. Otherwise your changes will not reflect on your submitted executable project.

- **Lab 05 Submission:**
Submit your project via the appropriate Canvas Assignments tool. You will **submit three files**: (the .exe executable file, the Visual C# .cs file and a zipped file of your entire project.) **and your Designer UI design document** that contains all of your completed pseudo code and screen shots as well. If you have any problems locating these files, see “How to find your files” document in Resources Module.

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