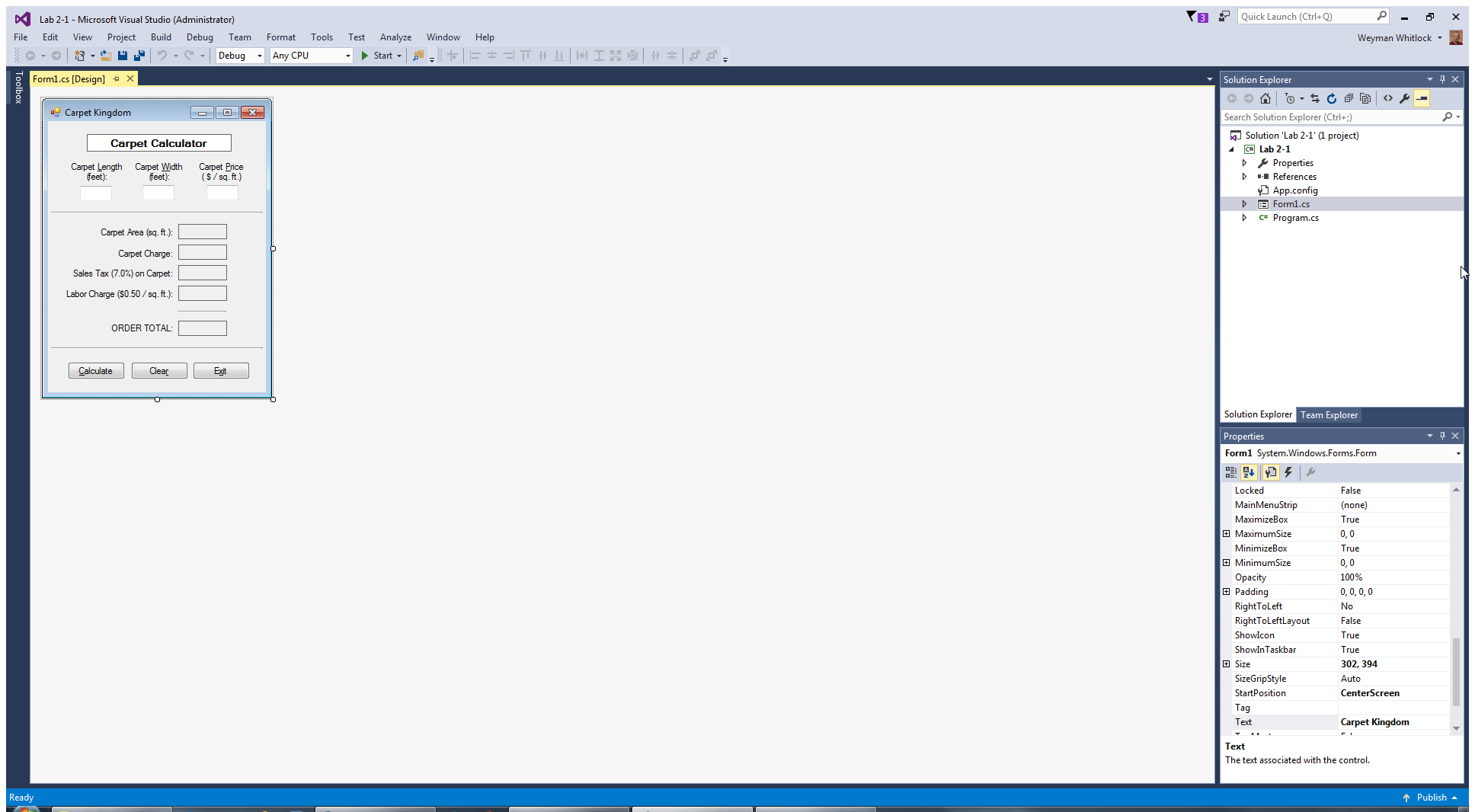
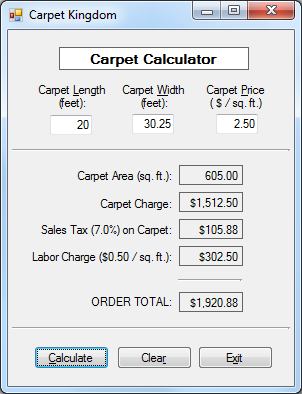
**Visual C# – Lab II**

The principal purposes of this lab exercise are to provide experience in:

* creating labels, text boxes, buttons and lines
* sizing and aligning controls on a form
* setting the background color of a label
* declaring and using constants and variables
* using the try-catch statement
* performing mathematical calculations
* converting data between data types
* formatting numeric data for display

 Design Time View: Run Time View:



The three text boxes shown are to be named:

lengthTextBox

widthTextBox

priceTextBox

The five output labels shown are to be named:

areaLabel

carpetChargeLabel

taxLabel

laborChargeLabel

orderTotalLabel

The three buttons shown are to be named:

calculateButton

clearButton

exitButton

Create a new Visual C# project named **Lab 2-1** and save it to your USB drive (or, to your hard drive if you are using your own laptop to do this lab exercise). Your goal, during design time of this project, is to create the form shown on the left, above. Note the following features of this form:

* The Title Bar of the form displays “**Carpet Kingdom**” (set with the Text property of the form).
* The “**Carpet Calculator**” title text is enclosed in a label that has its AutoSize property set to **False**, its BorderStyle property set to **FixedSingle**, its BackColor property set to **White**, and its TextAlign property set to **MiddleCenter**. A font size of **10** is used for the text in this label.
* The three text boxes used for data input in the top section of the form each have their TextAlign property set to **Right**. (Controls used to display numeric values should use this text alignment.) The five labels used for data output in the middle section of the form each have their TextAlign property set to **MiddleRight**.
* The two long horizontal lines used to separate the form visually into three sections, as well as the shorter horizontal line above the order total box, are each created by using a “trick”: using a GroupBox control with the Height component of its Size property set to **3**, and a blank Text property.
* All controls on the form are well-aligned with each other. Horizontal and vertical alignments of controls are set carefully for a professional-looking form. Tab Order is set correctly. (See pp. 175-177 in the textbook.) Access Keys are set for controls as shown on the form on the right above. (See pp. 178-179 in the textbook.)
* When the program runs, the form should appear centered on the screen. (See Chapter 3 PowerPoint slides.)

The code for this project should include the features described below. After completing each step, run the pro­gram to see if any errors are reported. Correct any such errors before proceeding with the next step. Remember to include comments throughout the program to identify the purpose of key statements and blocks of code.

* The first lines of the program are comments similar to the following  
    
  **// Programmer: Your Name  
  // Project: Lab 2-1  
  // Date: 09/14/2017  
  // Description: Form for taking a carpet order.**
* A try-catch statement is used in the Click event handler of the calculateButton control. (See pp. 157-160 in the textbook.) This is used so that any incorrect data entered by the user – e.g., a word instead of a number – will not cause the program to crash. For the catch section, code similar to that shown at the bottom of p. 159 in the textbook is used to display an appropriate error message for any incorrect data entry.
* Constants are used to store values for the sales tax rate and the labor charge rate. (See pp. 164-165 in the textbook.) A tax rate of 7% (0.07) is used to calculate the tax on the carpet charge amount. A labor rate of 50 cents (0.50) per square foot of carpet is charged for installation. Variables are declared for all other values stored and calculated by the program (described below).
* When the Calculate button is clicked, the following actions take place (via code within the try section):
  + The values entered into the three text boxes are parsed into numeric values and stored in numeric variables of appropriate data types. (See pp. 141-143 in the textbook.)
  + The carpet area is calculated as the product of the carpet length (entered by the user) times the carpet width (entered by the user). This value is displayed formatted with two decimal places. (See pp.144-145 and pp. 149-151 in the textbook.)
  + The carpet charge is calculated as the product of the carpet area times the carpet price per square foot (entered by the user). This value is displayed formatted as currency.
  + The sales tax is calculated as the product of the carpet charge times the sales tax rate (which is stored as a constant). (Sales tax is not applied to the labor charge.) This value is displayed formatted as currency.
  + The labor charge is calculated as the product of the carpet area times the labor rate (which is stored as a constant). This value is displayed formatted as currency.
  + The order total is calculated as the sum of the carpet charge plus the sales tax plus the labor charge. This value is displayed formatted as currency.
* When the Clear button is clicked, the three text boxes and the five labels used to display program output are all cleared. (See p. 119 in the textbook.) Then, the carpet length text box is given the focus, preparatory for processing another order. (See pp. 177-178 in the textbook.)
* When the Exit button is clicked, the program ends. (See p. 105 in the textbook.)

Save your project and then run it. Enter the sample data shown on the previous page and click the Calculate button. The appearance of the form, and the calculated values, should be as shown on the previous page.

When you have completed this project, you may exit Visual Studio. (Don’t forget to take your USB drive with you, if you are using one on a lab computer.)

You should seek additional practice in writing Visual C# programs to process data by attempting several of the Chapter 3 end-of-chapter Programming Problems on pages 196-199 of the textbook. Such repeated practice is crucial to mastering the skills presented in this chapter. Visit your instructor during his office hours (or the TAs for this course) to discuss questions or problems that you may have with these Programming Problems.