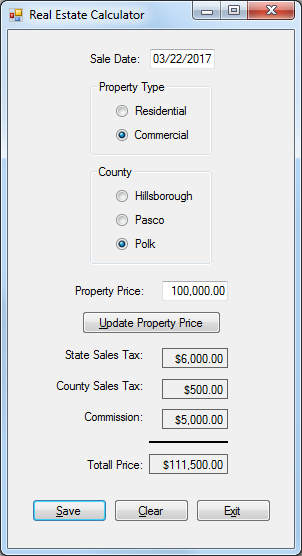
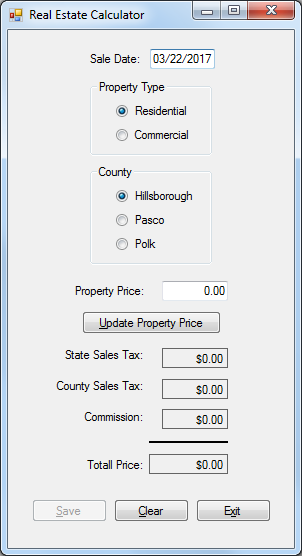
**Visual C# – Lab IV**

**Form after loading: Form after data entry and updating:**



**You are to create the data-entry form shown above. Include all of the following in your program:**

NOTE: This program differs from previous work you have done in that there is not a “Calculate” button used to gen­erate calculated values. Instead, there are three different actions that the user can take that will cause the four calcu­lated values to automatically update: 1) changing the Property Type radio button setting, 2) changing the County radio button setting, or 3) changing the Property Price entered and clicking the Update Property Price button.

* Using a project name of **Lab 4**-**1**, create a form and controls as shown in the diagram above. Name the controls using standard naming conventions (such as totalPriceLabel).
* In the program code, take care to properly declare all constants and variables needed by your program. Make sure that the declarations are for appropriate data types and that the declarations are located in appropriate locations in the program code. *Hint: In this program you should declare six constants at the class level to handle the sales tax rates and commission rates described below. The five variables used for the values used in the calculations can be declared there also so that they are declared only one time.*
* The Sale Date field should utilize a masked text box control with a mask of 00/00/0000 in order to display the date with that format. Have the control automatically default to displaying the current date (i.e., the date the computer’s system clock provides) by placing the following code in the form’s Load event handler: **saleDateMaskedTextBox.Text = DateTime.Now.ToString("MM/dd/yyyy");**
* Have the program begin with the Residential radio button selected as the default property type and the Hillsborough radio button selected as the default county. *Hint: This can be established by setting the Checked property of each of these two radio buttons to true at design time.* Also at startup, the Save button should have its Enabled property set to false. *This setting can be done at design time or in the form’s Load event handler.*
* The Property Price field is a text box into which the user will enter a numeric value. The Update Property Price button will be clicked by the user each time that the Property Price value is changed. *(See next page for more.)*
* The program user will not directly enter data for the values of State Sales Tax, County Sales Tax, Commis­sion, and Total Price; thus, these boxes should be labels, not text boxes.
* Whenever the user clicks any of the radio buttons, or clicks the Update Property Price button, the program is to perform a variety of actions. These code for these actions should be in a custom method named UpdateTotals. Initially, this method is to check to see that a valid, non-null, numeric value has been entered for Property Price. If that text box is empty, or a non-numeric value is entered, a message box should remind the user to enter a valid value and the focus should be sent to that text box. *Hint: Use a TryParse method in an if-else statement to do this. The if-else statement should begin with the following line (assuming you have used these names):*  
    
   **if (decimal.TryParse(propertyPriceTextBox.Text, out propertyPrice))**
* When there is a valid value for Property Price, additional code in the UpdateTotals method should use it in cal­culating and displaying the values of the calculated fields. Here are the data values to use in the computations:

State Sales Tax Rate: 0.06

*These values will not change during the execution of the program and should thus be declared as constants (which can be placed at the class level for easy access in performing periodic updates).*

County Sales Tax Rates: 0.01 for Hillsborough

0.00 for Pasco

0.005 for Polk

Commission Rates: 0.06 for Residential

0.05 for Commercial

The values above are multiplied times the Purchase Price to determine the State Sales Tax, County Sales Tax, and Commission values that are to be displayed on the form. The Total Price value to be displayed is the sum of the Purchase Price and the three calculated values just described. After these calculations are performed, the four calculated values should all be displayed with a currency format. All values should also display right-aligned in their controls. (The screen shot above shows sample data you can use to validate your calcula­tions.) Finally, if the Property Price text box contains a valid numeric value greater than zero, the UpdateTotals method should change the value of the Enabled property of the Save button to true.

* The code for the Save button’s click event handler should save of the form’s data to a text file. This text file should be named Properties.txt and it should be located in the default location. (C# will store the file in the bin\Debug subfolder of the project folder by default.) Write your code so that each use of the Save button appends the current form data to the end of the existing file. The format of the data in the text file can be seen in the example data below:

Date: 03/22/2017

Property Price: $100,000.00

State Sales Tax: $6,000.00

County Sales Tax: $500.00

Commission: $5,000.00

Total Price: $111,500.00

The Save button’s event handler code should utilize a Try-Catch structure to hold its code. Code utilizing a StreamWriter object should be included in the Try block to write the data (shown above) to the text file. After the data is written to the file, remember to close the file. Then, use a message box to display a message confirm­ing to the user that the data was written to the file. Include appropriate title bar text and an information icon in this message box. Finally, the event handler code should call the custom ClearForm method (described below).

* A custom method named ClearForm should be written. When this method runs, all user-entered data and calculated data is to be cleared and the form returned to its original state. All default settings should be restored, including resetting the radio buttons and displaying the current date in the Sale Date control. The Save button should also be disabled. This event should conclude with the focus being sent to the Property Price text box. This custom method should be called whenever the user clicks the Clear button.
* When the user clicks the Exit button, the Close method should be used to end the program.
* There should be Access Keys (as shown above) and ToolTips for each of the four buttons on the form.
* NEW: One additional piece of code is needed in this program. We have not used this technique in any of our previous programs. Double-click on the propertyPriceTextBox control. This will create a TextChanged event handler for this text box. The code placed in this event handler will run every time that a change is made to the text in the text box. We want any such change in this text to result in the Save button being disabled until the UpdateTotals method runs again (which will re-enable the button). Use the following code in this event handler:

**saveButton.Enabled = false;**