Exercise 6-1    Enhance the Invoice Total application

In this exercise, you’ll enhance the Invoice Total application of chapter 5 by adding a procedure that determines the discount percent to it. You’ll also add another event handler to the application.

1. Open the application that’s in the C:\VB 2010\Chapter 06\InvoiceTotal directory.

2. Start a Sub procedure with three parameters (customer type, subtotal, and discount percent) that will set the discount percent based on the customer type and subtotal. Then, copy the code for doing that from the event handler of the btnCalculate_Click procedure to the new Sub procedure, and modify this code so it works correctly.

3. Modify the code in the btnCalculate_Click procedure so it calls the Sub procedure to get the discount percent. As you enter the call, notice how the IntelliSense feature helps you enter the arguments. Then, test these changes to make sure they work.

4. Add a Function procedure with two parameters (customer type and subtotal) that returns the discount percent. Then, modify the calling statement and any other code in the btnCalculate_Click procedure so it uses the Function procedure to get the discount percent. Now, test this change.

5. Add one event handler named ClearAllBoxes that handles the TextChanged event for both the Customer Type and Subtotal text boxes. This event handler should clear the values in the text boxes that display the results. After you test this change, close this project.
Exercise 6-2  Enhance the Future Value application

In this exercise, you’ll experiment with and enhance the Future Value application that was presented in this chapter.

Experiment with the FutureValue procedure
1. Open the application that’s in the C:\VB 2010\Chapter 06\FutureValue directory. This is the application that’s presented in figure 6-10. Start by testing this application.

2. Comment out the Return statement in the FutureValue function, and test the application again to see that the function still works.

3. Instead of using FutureValue within the For loop in the function, declare and use a variable named newFutureValue. Then, use the Return statement to return the value in that variable. Now, test this to make sure it works.

Code an event handler for a form event
4. Use the drop-down lists at the top of the Code Editor to start an event handler for the DoubleClick event of the form. Next, change the name of the event handler to ClearAllBoxes, and write the code for this handler so it sets the Text property for all four text boxes to an empty string. Then, test the application to make sure this works. (Be sure to double-click on the form, not on the form’s title bar, because that will maximize the form.)

Code event handlers for other control events
5. In the Form Designer, select the Future Value text box. Next, select the MouseHover event, read its description, drop down its list, and select the ClearAllBoxes event handler so it will be executed whenever the user lets the mouse hover over this text box. Then, test the application to make sure this works.

6. In the Form Designer, select the Yearly Interest Rate text box, and double-click on the DoubleClick event to generate its event handler. Next, write the code for this handler so it sets the value in the Yearly Interest Rate text box to 12. Then, test this enhancement.

7. Use your own imagination to work with other events. When you’re through experimenting with events, close the application.