12

Web Applications

12.1 Programming for the Web, Part I

In earlier chapters all of our programs ran directly under Windows. Now we'll create programs that run in a Web browser (such as Internet Explorer, Firefox, Chrome, or Safari). The programs in this chapter cannot be created with Visual Basic 2010 Express; we'll need to use Visual Web Developer 2010 Express (provided on the DVD accompanying this book) or a full version of Visual Studio 2010. Our discussion will refer to Visual Web Developer 2010 Express (abbreviated as VWD), but it can easily be modified to apply to Visual Studio.

When you use a Web browser, your computer (referred to as a **client**) is making requests to another computer (referred to as a **server**). Some requests ask the server to send a particular Web page. Others require the server to perform an operation using input provided by the client and then to return a new version of a page. Visual Web Developer uses a Microsoft technology called ASP.NET to respond to requests. (ASP stands for **Active Server Pages**.)

Although the client and server are normally separate computers, VWD allows Web programs to be developed on a single computer. When VWD runs a Web program, it launches a Web server and then opens a Web browser that connects to the server. After the programs are developed, they can be transferred to a dedicated server and can then be accessed by any computer's Web browser. A dedicated server running programs created with VWD requires special Web server software called IIS (Internet Information Services).

VideoNote
Programming for the Web

Creating a Web Program with Visual Web Developer

As we will soon see, a Web program is created in much the same way as a Visual Basic Windows program. We break up the creation of a Web program here into three walkthroughs to be performed in succession. After each walkthrough we present a table showing how the steps of the Web program walkthrough differ from their Visual Basic counterparts.

First walkthrough: Starting a new Web program

1. Click on the Windows *Start* button, click on *All Programs*, and click on *Microsoft Visual Web Developer 2010 Express* in the list of programs. (A Start page very similar to the Visual Basic Start page will appear.)
2. Click on *Options* in the *Tools* menu, select *General* from the left side of the Options dialog box, and click on the *Design View* radio button in the *Start Pages in* section. (This step will not have to be done for subsequent programs.) Optionally, set the value for the "Tab and indent size" to 2. Click on the *OK* button to close the Options window.
3. Click on *New Web Site* in the *File* menu. (A New Web Site dialog box will appear. *Visual Basic* should be selected as the Installed Template and *File System* as the "Web location." If not, change these items.)
4. The wide combo box at the center-bottom of the dialog box gives the location and name of the program. (The default name will be WebSite1.) We recommend changing the name to something like MyWebProgram. Also, you might want to change the path.
5. Double-click on *ASP.NET Web Site*. The IDE for VWD appears in design mode. See Fig. 12.1. The text and controls will be entered inside the red rectangle with tab "MainContent (Custom)". We will refer to this region as the **Main Content region**.

Table 12.1 shows three ways in which starting a program in VWD differs from starting a program in Visual Basic.

Second walkthrough: Designing the Web page

Note 1: This walkthrough is a continuation of the first walkthrough.

Note 2: In VWD, text can be typed into a page and formatted in somewhat the same way as in a word processor.

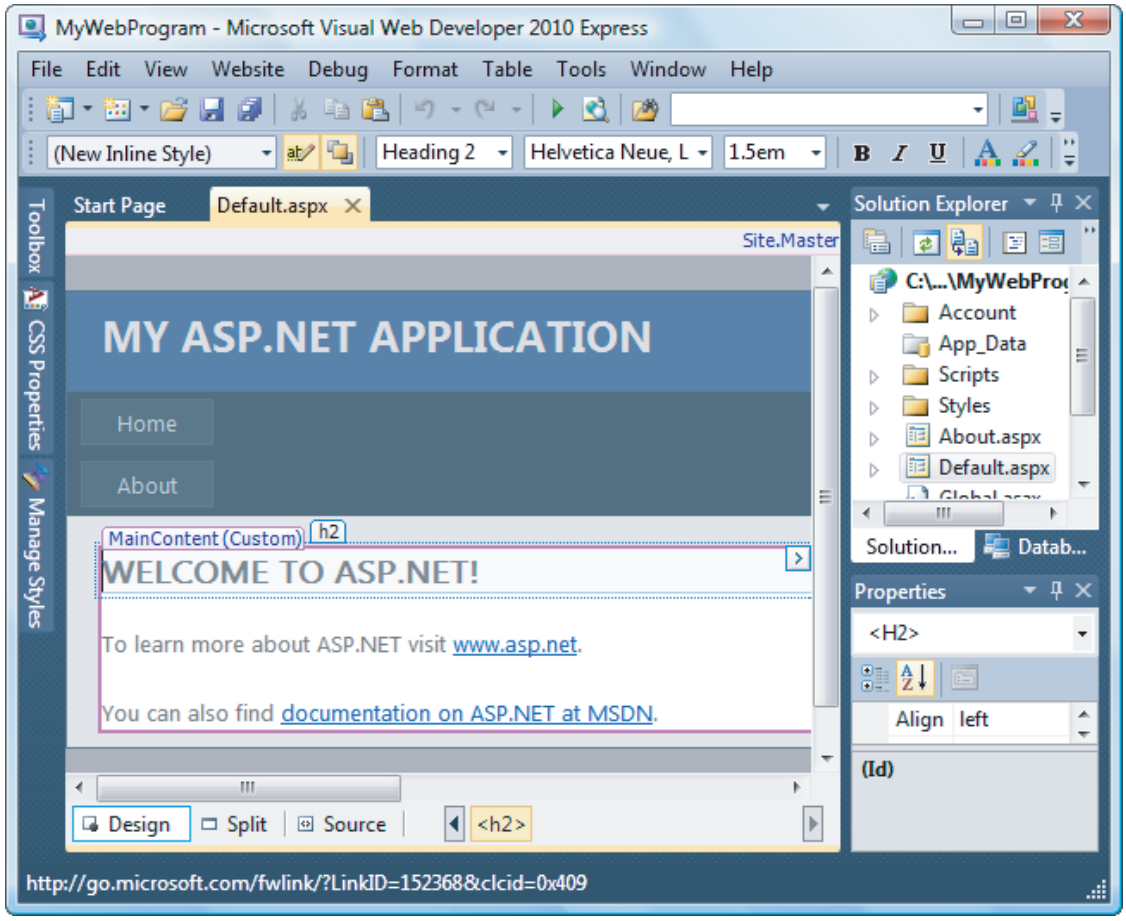


FIGURE 12.1 The VWD integrated development environment in design mode.

TABLE 12.1 Web program counterparts to starting a Visual Basic program.

Visual Basic Windows Program	VWD Program
1. Program can be created with Visual Basic Express.	Program must be created with Visual Web Developer Express or Visual Studio.
2. You launch a program by clicking on *New Project* from the *File* menu and then double-clicking on *Windows Forms Application*.	You launch a program by clicking on *New Web Site* from the *File* menu and then double-clicking *ASP.NET Web Site*.
3. The name given to a program when it is launched is temporary. The actual name is specified with the *Save All* command.	The name given to a program when it is launched is permanent. It is *not* specified after you click on the *Save All* button.

1. Select the text in the Main Content region and then press the DEL key to delete the text in the region.
2. Type the words "Tip Calculator" into the Main Content window, drag the cursor across the two words and click on the *Bold* button (**B**) on the Toolbar. (The words now appear in boldface. Other buttons on the Toolbar can be used to change the font name, the font size, and the foreground and background colors of the words.)
3. Press the Enter key and type the words "Cost of meal:". (Notice that a faint blue rectangle with the tag labeled *p* appears. Think of *p* as standing for "paragraph". As with a word processor, the pressing of the Enter key creates a new paragraph.)

4. Press the space bar twice and then double-click on the TextBox control in the *Standard* group of the Toolbox. Notice that a text box appears at the insertion point (that is, at the cursor location).
5. Make sure that the text box is selected. In the Properties window, click on the *Alphabetize* button, locate the ID property near the top of the Properties window, and set the value of the ID property to txtCost. (VWD's ID property is the counterpart to Visual Basic's Name property. Notice that the tab above the text box reads "asp:TextBox#txtCost".)
6. Position the cursor to the right of the text box, press the Enter key, type "Percent tip (such as, 15, 18.5, or 20):", press the space bar twice, and double-click on the TextBox control in the Toolbar. (Another text box appears.)
7. Set the text box's ID property to txtPercent.
8. Position the cursor to the right of the text box, press the Enter key, double-click on the Button control in the Toolbox (a button appears at the insertion point), set the button's ID property to btnCalculate, and set its Text property to "Calculate Tip".
9. Position the cursor to the right of the button, press the Enter key, type "Amount of tip:", press the space bar twice, double-click on the TextBox control in the Toolbox (a text box appears at the insertion point), set the text box's ID property to txtTip, set its ReadOnly property to *True*, and position the cursor to the right of the text box. The screen now appears as in Fig. 12.2.

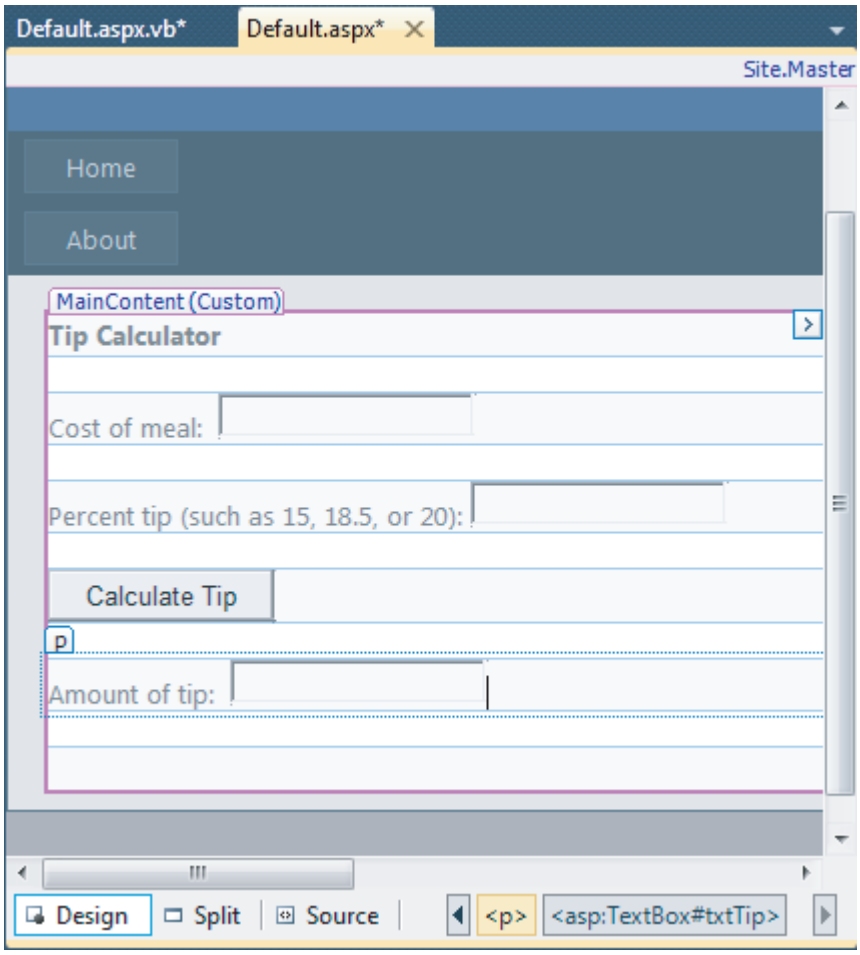


FIGURE 12.2 **The Designer for our program.**

The counterpart in VWD of a Visual Basic Windows form is referred to as a **Web page**. Table 12.2 shows seven ways in which designing a Web page differs from designing a Visual Basic form.

TABLE 12.2 Web page counterparts to designing a Visual Basic Windows form.

	Visual Basic Windows Form	VWD Web Page
1.	There is no cursor on the form designer.	The page designer has a cursor (also referred to as the *insertion point*).
2.	Text to appear permanently on a form is usually placed in a label and formatted with settings from the Properties window.	Permanent text (called *static text*) can be typed directly into the page at the insertion point and formatted from the Toolbar in the same way as text typed into a word processor.
3.	The most frequently used controls (such as Button, ListBox, and TextBox) are found in the *Common Controls* group of the Toolbox.	The most frequently used controls (such as Button, ListBox, and TextBox) are found in the *Standard* group of the Toolbox.
4.	When you double-click on a control in the Toolbox, the control appears either in the upper-left corner of the form or just below the most recently created control.	When you double-click on a control in the Toolbox, the control appears at the insertion point.
5.	The name of a control is specified by setting its Name property.	The name of a control is specified by setting its ID property.
6.	In design mode, controls are placed anywhere on a form and aligned with the *Format* menu.	In design mode, text and controls are placed in a top-to-bottom fashion, each entered at the insertion point.
7.	The tab for the Designer reads "*frmName*.vb [Design]".	The tab for the Designer reads "Default.aspx".

Third walkthrough: Coding and executing a Web program

Note: This walkthrough is a continuation of the second walkthrough.

1. Double-click on the *Calculate* button. (The Code Editor will appear with a template for the button's default event procedure, btnCalculate_Click. In VWD, the code in the Code Editor is called the **code-behind**.)
2. Type the following three lines of code into the btnCalculate_Click event procedure:

```
Dim cost As Double = CDbl(txtCost.Text)
Dim percent As Double = CDbl(txtPercent.Text) / 100
txtTip.Text = FormatCurrency(percent * cost)
```

3. Press Ctrl + F5 to run the program without debugging. (Your Web browser will open after a few seconds. Figure 12.3 shows the upper-left corner of the Web page in the browser.)
4. Type *20* into txtCost, type *15* into txtPercent, and click on the button. The value $3.00 will appear in txtTip.
5. Click on the browser's *Close* button to terminate the program and return to the Code Editor. Notice that the tab for the Code Editor reads *Default.aspx.vb*. (An Output window appears below the Code Editor. You can remove the Output window by clicking on its *Close* button.)
6. At this point you can alter the code in the Code Editor if desired or click on the *Default.aspx* tab or return to the Designer and make further changes to the form.

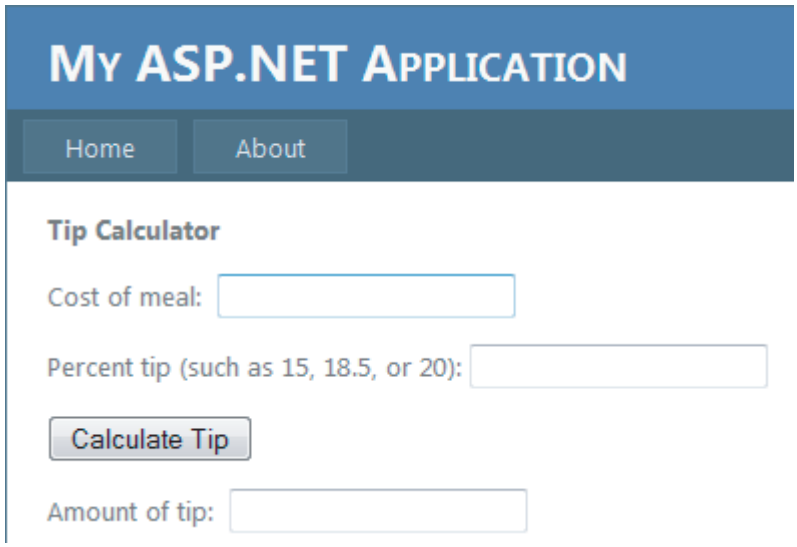


FIGURE 12.3 **Contents of the Web browser after the program is run.**

7. Click on the *Save All* button to save the program and then click on *Close Project* in the *File* menu to close the program. (***Note:*** If you are using Visual Studio instead of VWD, click on *Close Solution* in the *File* menu to close the program.)

Table 12.3 shows four ways in which coding and executing a Web program differs from coding and executing a Visual Basic form.

TABLE 12.3 **Web program counterparts to coding and executing a Visual Basic program.**

	Visual Basic Program	Web Program
1.	The tab for the Code Editor reads "frmName.vb".	The tab for the Code Editor reads "Default.aspx.vb".
2.	We usually run a program by pressing F5 or clicking on the *Start Debugging* button on the Toolbar.	We will usually run a program by pressing Ctrl + F5 to start *without* debugging.
3.	Programs are run as a Windows application.	Programs are run in a Web browser interacting with a Web server.
4.	Programs are terminated by clicking on the form's *Close* button or pressing Alt + F4.	Programs are terminated by closing the browser.

Using a Table to Lay Out a Web Page's Content

The layout of the page created in the second walkthrough is not as nice looking as those we have been creating with Visual Basic. In our Visual Basic design, the three text boxes would be left-aligned and the button would be centered horizontally in the form. A table control can be used with VWD to improve the appearance of the Web page.

A table control is a rectangular array of cells, where text and/or controls can be placed into each cell. The columns of the table are used to align text and controls. The following walkthrough uses a table to create an improved version of the previous program.

A table walkthrough: Using a table to improve the design of a Web program

Note: This walkthrough is a not a continuation of the previous walkthroughs. It produces the same program, but with a more attractive page design.

1. Start a new Web program with the name TipCalculator, and clear the contents of the Main Content region as before.
2. Click on *Insert Table* from the *Table* menu to produce the Insert Table dialog box shown in Fig. 12.4.

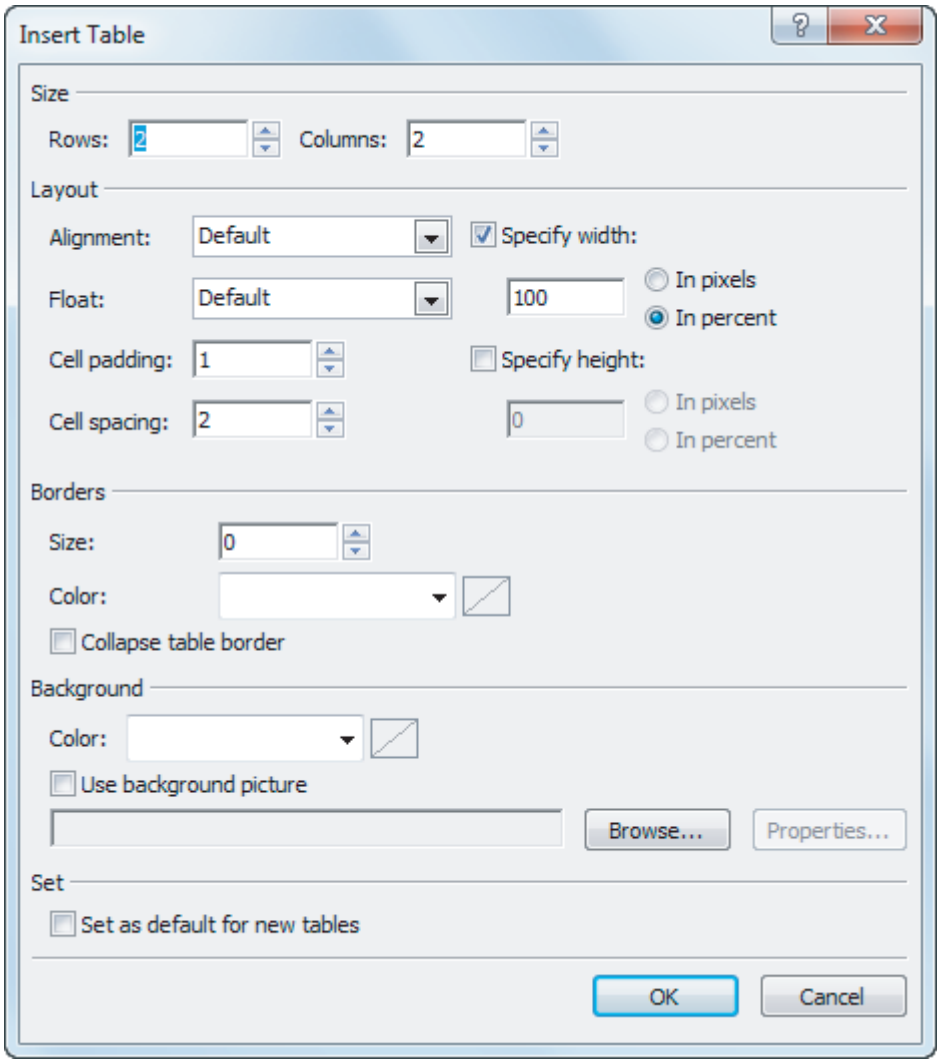


FIGURE 12.4 The Insert Table dialog box.

3. Set the number of rows to 5 and the number of columns to 2. (The labels for the text boxes will be placed right-justified into the first column and the text boxes will be placed left-justified into the second column.)
4. Click on the first *In pixels* radio button and change the number in the accompanying text box from 100 to 500. (The table will be 500 pixels wide.)
5. Click on the *OK* button to generate the faint 5-by-2 grid shown in Fig. 12.5. (We will use the grid at design time to align text and controls. The grid will not be visible at run time.)

FIGURE 12.5 **A table with five rows and two columns.**

6. Hover the mouse over the top of the first column until a down-arrow (⬇) appears; then click on the left mouse button. (The first column has now been selected.) In the Properties window, set the Align property to *right*. This will cause the entries in that column to be right-justified. ***Note:*** There is no need to set the Align property for the right column, since its default alignment is *left*.
7. Type "Cost of meal:" into the first column of the second row. Move the cursor to the second column of the second row and double-click on the TextBox control in the Toolbox. (A text box appears at the insertion point.) Give the text box the name txtCost by setting its ID property to txtCost.
8. Use a process similar to Step 7 to place "Percent tip (such as 15, 18.5, or 20):" into the left column of the third row and a text box named txtPercent into the right column.
9. Use a process similar to Step 7 to place "Amount of tip:" into the left column of the fifth row and a read-only text box named txtTip into the right column.
10. Drag the mouse over the cells in the first row to select the row. Then hover over *Modify* in the *Table* menu and click on *Merge Cells*. (The first row will now be reduced from two cells to one cell.)
11. With the first row still highlighted, set its Align property to *center*, type the words "Tip Calculator" into the row, and make the words boldface.
12. In a manner similar to Steps 10 and 11, place a button named btnCalculate and text "Calculate Tip" in the center of the fourth row.
13. Resize the text boxes, columns, rows, and table to remove extraneous space. The best way to resize a text box is to change the value of its Width property. To resize a column, hover the mouse over the double bar on the right side of the column until a horizontal double-arrow (⇔) appears; then drag the arrow. To resize a row, hover the mouse over the double bar at the bottom side of the row until a vertical double-arrow (⇕) appears; then drag the arrow. To resize the table, hover the mouse over the upper-left corner of the table until an arrow cross (✥) appears; then click the left mouse button. The table will now be selected, and it can be resized by dragging its sizing handles. The Main Content region of the Web page should look like Fig. 12.6.
14. Double-click on the button and enter the following code into the btnCalculate event procedure:

```
Dim cost As Double = CDbl(txtCost.Text)
Dim percent As Double = CDbl(txtPercent.Text) / 100
txtTip.Text = FormatCurrency(percent * cost)
```

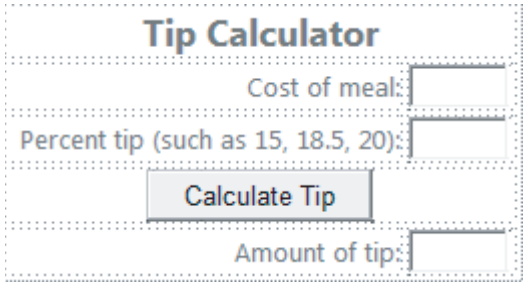


FIGURE 12.6 Web page in design mode.

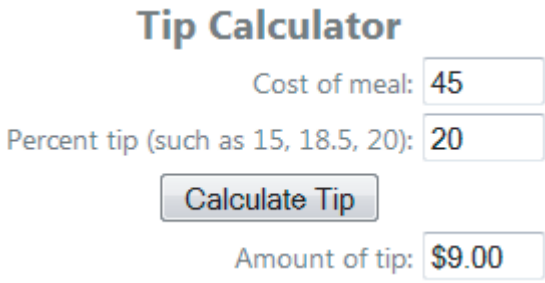


FIGURE 12.7 Web page in run time.

15. Press Ctrl + F5 to run the program, enter values into the first two text boxes, and click on the button. Figure 12.7 shows a possible output.

Some additional features of tables are as follows:

1. In the walkthrough we used a down-arrow to select an entire column. To select an entire row in an analogous way, hover the mouse over the left side of the row until a right-arrow (➡) appears; then click the left mouse button.
2. To add a new row to a table, click on a row of the table, click on *Insert* in the *Table* menu, and click on *Row Above* or *Row Below*. A similar process can be used to add a new column.
3. To delete a row or column from a table, hover over *Delete* in the *Table* menu, and then click on *Delete Rows* or *Delete Columns*.
4. To delete a table, hover the mouse over the upper-left corner of the table until an arrow cross appears, press the left mouse button to select the entire table, and then click on the DEL key.
5. Both rows and cells have an Align property.
6. The Align property specifies horizontal alignment. The valign property specifies vertical alignment.

Accessing a Text File in a Web Program

With Visual Basic, we placed text files in the program's *bin**Debug* folder and accessed the file with a statement of the form

```
Dim strArrayName() As String = IO.File.ReadAllLines(filespec)
```

VWD does not have a *bin**Debug* folder. With VWD, text files are usually placed in the App_Data folder of the Solution Explorer window and accessed with a statement of the form

```
Dim strArrayName() As String =
                        IO.File.ReadAllLines(MapPath("App_Data\" & filename))
```

We can then use LINQ with such an array, as we did earlier with Visual Basic programs.

Binding a Control to a LINQ Query

The following pair of statements display the results of a LINQ query in a list box:

```
lstBox.DataSource = query
lstBox.DataBind()
```

In VWD, the counterpart of the DataGridView control is the GridView control. The following pair of statements bind a GridView control to a query:

```
grvGrid.DataSource = query
grvGrid.DataBind()
```

There is no need to set a GridView's CurrentCell property to Nothing, or to specify its size. (VWD will automatically size the control.) By default a GridView control does not use row headers, therefore there is no RowHeadersVisible property that needs to be set to False. Also, there is no need to set the SelectedItem property of a list box to Nothing. Column headers can be specified with statements such as

```
grvGrid.HeaderRow.Cells(0).Text = header for first column
grvGrid.HeaderRow.Cells(1).Text = header for second column
```

Opening an Existing Web Program

The following steps open the program MyWebProgram that was created in the first three walkthroughs:

1. Click on *Open Web Site* in the *File* menu.
2. Navigate to the folder named *MyWebProgram* that was created in the walkthrough.
3. Click on the *Open* button. (If a program is currently loaded, you may be prompted to save it.)
4. If the Designer or Code Editor is not visible, right-click on the file Default.aspx in the Solution Explorer window and click on *View Designer* or *View Code*.

Building on an Existing Web Program

Some exercises in this text require you to create a new program that extends an already created program. The following steps save you from having to design the page and enter the code a second time:

1. Start the new program, note the path to its folder, and close the new program.
2. Launch Windows Explorer.
3. Open the folder containing the old program, press Ctrl + A to select its contents, and press Ctrl + C to copy the contents of the folder into the Clipboard.
4. Open the folder containing the new program, press Ctrl + A to select its contents, press the DEL key to delete the contents, and press Ctrl + V to paste the contents of the old folder into the new folder.
5. In VWD, click on *Recent Projects and Solutions* in the *File* menu, and click on the first line of the drop-down list that appears.

You have created a new program that is a copy of the old program. You can now proceed to make modifications.

Comments

1. Important reminders: When starting a new Web program, click on *New Web Site*, not *New Project*, from the *File* menu. When opening an existing Web program, click on *Open Web*

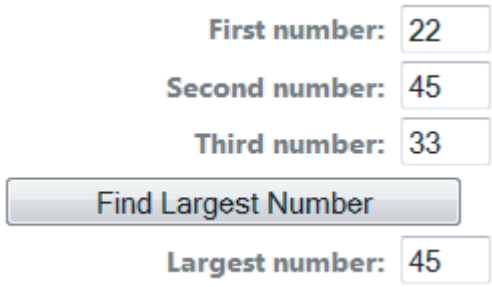
Site, not *Open Project*, from the *File* menu. However, when closing a Web program, click on *Close Project*.

2. We have been running Web programs by pressing Ctrl + F5 to run *without* debugging. However, if you prefer, you can click on the *Start Debugging* button in the Toolbar instead. If so, you might have to click on an *OK* button in a dialog box to proceed. Also, the program might still be running after you exit the browser. If so, you will have to click on *Stop Debugging* in the *Debug* menu to terminate the program.
3. Renaming a Web program is much easier than renaming a Windows program. Just close the program and use Windows Explorer to rename the program's folder.
4. In design mode, the bottom of the Designer contains three buttons labeled *Design*, *Split*, and *Source*. If you click on the *Source* button, you will see ASP.NET code corresponding to the design of the page. This code can be altered directly to change the page design. If you press the *Split* button, you will see a split screen showing both the page and its ASP.NET code.

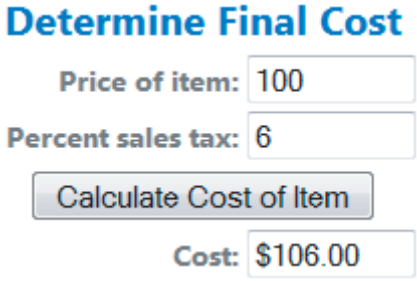
EXERCISES 12.1

In Exercises 1 through 4, write a Web program corresponding to the outcome shown.

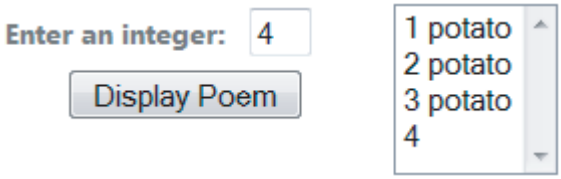
1.



2.



3.



4.

5. If a bond is purchased for n dollars and sold one year later for m dollars, then the *discount rate* is $\frac{m-n}{m}$ and the *interest rate* is $\frac{m-n}{n}$, where each is expressed as a percent. Write a Web program to calculate the discount rate and interest rate for a bond. See Fig. 12.8.

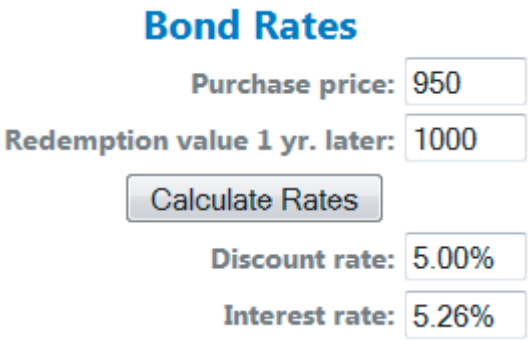


FIGURE 12.8 An outcome of Exercise 5.

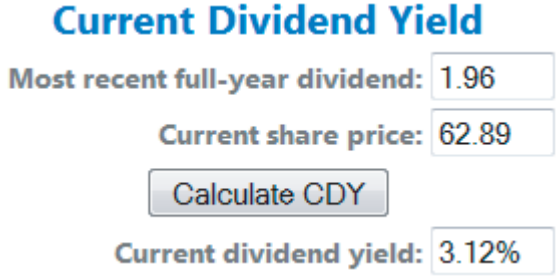


FIGURE 12.9 An outcome of Exercise 6.

6. The *current dividend yield* for common stock is calculated with the formula

$$\text{current dividend yield} = \frac{\text{most recent full-year dividend}}{\text{current share price}}.$$

Write a Web program to calculate the current dividend yield for a stock. See Fig. 12.9.

In Exercises 7 through 16, rework the example or exercise as a Web program.

7. Example 9 of Section 4.2
8. Example 3 of Section 5.1
9. Exercise 35 of Section 5.2
10. Exercise 36 of Section 5.2
11. Example 3 of Section 5.3
12. Exercise 11 of Section 5.3
13. Example 3 of Section 6.2
14. Exercise 30 of Section 6.1
15. Exercise 48 of Section 7.1
16. Example 9 of Section 7.1

The file States.txt contains the 50 U.S. states in the order in which they joined the union. Use this text file in Exercises 17 through 20.

17. Write a program whose page contains a button and a list box. When the user clicks on the button, the names of the states that end with "ia" should be displayed in the list box.

18. Write a program whose page contains a button and a list box. When the user clicks on the button, the program should determine the greatest length of the names of the states and then display (in the list box) all states having names of that length.

19. Write a program whose page contains a text box, a button, and a list box. When the user enters a letter into the text box and clicks on the button, the states beginning with that letter should be displayed in the list box.

20. Write a program whose page contains two buttons, a text box, and a list box. See Fig. 12.10. The user should be able to display a list of all states beginning with a vowel or all states beginning with a consonant (in alphabetical order) and determine the number of such states. ***Note:*** List boxes in VWD do not have a Sorted property.

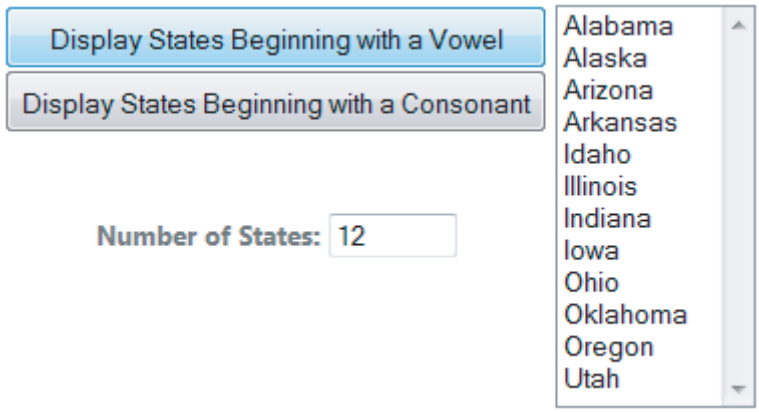


FIGURE 12.10 An outcome of Exercise 20.

12.2 Programming for the Web, Part II

In this section we explore some controls that are unique to VWD and some special features of Visual Web Developer.