3. You use the ______________ method to add items to a combo box.
   a. Add
   b. AddItem
   c. AddList
   d. ItemAdd

4. Write the statement to refer to the second image stored in the ImageList1 control's Images collection.

You have completed the concepts section of Chapter 7. The Programming Tutorial section is next.

PROGRAMMING TUTORIAL 1

Creating the Slot Machine Application

In this tutorial, you create an application that simulates a slot machine. Figures 7-33 and 7-34 show the application's TOE chart and MainForm, respectively. The MainForm contains a table layout panel, three picture boxes, and two buttons. The application will store six different images in an image list control. When the user clicks the Click Here button, the button's Click event procedure will generate 10 sets of three random numbers. The random numbers will be used to select images from the image list control. The selected images will be displayed, one at a time, in the three picture boxes. If the final three random numbers are the same, the picture boxes will contain the same image. When that happens, the Click event procedure will display the message "Congratulations!" in a message box.

The Ch07ProgTut1 video demonstrates all of the steps in Programming Tutorial 1.

Task	Object	Event
1. Display 10 random images from the ImageList1 control, one at a time, in the leftPictureBox, centerPictureBox, and rightPictureBox 2. Display "Congratulations!" in a message box when the leftPictureBox, centerPictureBox, and rightPictureBox contain the same image	clickHereButton	Click
End the application	exitButton	Click
Store 6 different images	ImageList1	None
Display random ImageList1 images (from clickHereButton)	leftPictureBox, centerPictureBox, rightPictureBox	None

Figure 7-33 TOE chart for the Slot Machine application

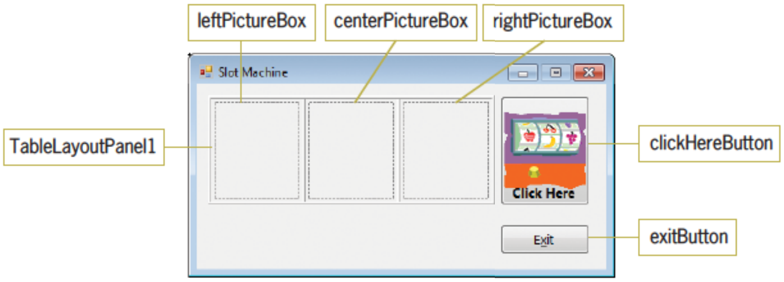


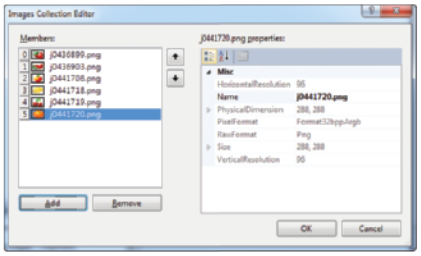
Figure 7-34 MainForm in the Slot Machine application

Instantiating the ImageList1 Control

Included in the data files for this book is a partially completed Slot Machine application. Before you begin coding the application, you will need to instantiate the ImageList1 control and then store the six images in its Images collection. (All of the images used in the application are from Microsoft's Clip Art collection and were downloaded from *www.office.microsoft.com/en-us/clipart*.)

To instantiate the ImageList1 control and then store images in its Images collection:

1. Start Visual Studio or the Express Edition of Visual Basic. If necessary, open the Solution Explorer window.
2. Open the **Slot Machine Solution** (**Slot Machine Solution.sln**) file, which is contained in the VbReloaded2010\Chap07\Slot Machine Solution folder. Open the designer window (if necessary), and then auto-hide the Solution Explorer window.
3. If necessary, open the Toolbox window and expand the Components section. Click **ImageList** and then drag an image list control to the form. Release the mouse button. The ImageList1 control appears in the component tray.
4. Click the ImageList1 control's **task box** and then click **Choose images** to open the Images Collection Editor window. Click the **Add** button. Open the VbReloaded2010\Chap07\Slot Machine Solution\Slot Machine Project\Resources folder. Click **j0436899.png** in the list of filenames. (The file contains an image of a strawberry.) Click the **Open** button. The filename and a small image appear in the Members list section of the Images Collection Editor window.
5. On your own, add the following five image files: j0436903.png (watermelon),j0441708.png (apple), j0441718.png (bananas), j0441719.png (cherries), and j0441720.png (orange). The completed Images Collection Editor window is shown in Figure 7-35.



0 j0436899.png
1 j0436903.png
2 j0441708.png
3 j0441718.png
4 j0441719.png
5 j0441720.png

j0441720.png properties:

Misc
HorizontalResolution 96
Name j0441720.png
PhysicalDimension 288, 288
PixelFormat Format32bppArgb
RawFormat Png
Size 288, 288
VerticalResolution 96

Figure 7-35 Completed Images Collection Editor window

6. Click the **OK** button to add the six images to the ImageList1 control's Images collection.
7. Change the Image Size in the ImageList Tasks box to **85, 96** (which is the size of the picture boxes on the form). Change the Image Bit Depth in the ImageList Tasks box to **Depth32Bit**.
8. Click the **form** to close the ImageList Tasks box. Close the Toolbox window and then save the solution.

Coding the Slot Machine Application

To complete the application, you just need to code it. According to the application's TOE chart, only the Click event procedures for the clickHereButton and exitButton need to be coded.

To begin coding the application:

1. Open the Code Editor window. Notice that the exitButton's Click event procedure has already been coded for you. In addition, the Option statements have already been entered in the General Declarations section.
2. In the comments that appear in the General Declarations section, replace <your name> and <current date> with your name and the current date.

Figure 7-36 shows the pseudocode for the clickHereButton's Click event procedure. As the pseudocode indicates, the procedure will use random numbers to display the appropriate images in the leftPictureBox, centerPictureBox, and rightPictureBox. It then will compare the random numbers to determine whether the "Congratulations!" message should be displayed.

```
clickHereButton Click event procedure
1.  disable the clickHereButton
2.  repeat for spins from 1 through 10 in increments of 1
          generate a random number from 0 through 5 and store it in a variable named leftIndex
          use the leftIndex variable to display the appropriate ImageList1 image in the leftPictureBox
          refresh the screen and pause the application

          generate a random number from 0 through 5 and store it in a variable named centerIndex
          use the centerIndex variable to display the appropriate ImageList1 image in the centerPictureBox
          refresh the screen and pause the application

          generate a random number from 0 through 5 and store it in a variable named rightIndex
          use the rightIndex variable to display the appropriate ImageList1 image in the rightPictureBox
          refresh the screen and pause the application
    end repeat for
3.  if the leftIndex, centerIndex, and rightIndex variables contain the same number
          display "Congratulations!" in a message box
    end if
4.  enable the clickHereButton
```

Figure 7-36 Pseudocode for the clickHereButton's Click event procedure

To code the clickHereButton's Click event procedure and then test the procedure's code:

1. Open the code template for the clickHereButton's Click event procedure. Type **' simulates a slot machine** and press **Enter** twice.
2. The procedure will need a Random object to represent the pseudo-random number generator, and three Integer variables to store the random numbers. Enter the following four Dim statements. Press **Enter** twice after typing the last Dim statement.

```
Dim randGen As New Random
Dim leftIndex As Integer
Dim centerIndex As Integer
Dim rightIndex As Integer
```

3. The first step in the pseudocode is to disable the clickHereButton. Enter the following assignment statement:

```
clickHereButton.Enabled = False
```

4. Step 2 in the pseudocode is a counter-controlled loop that repeats the loop body 10 times. Enter the following For clause:

```
For spins As Integer = 1 To 10
```

5. Notice that the Code Editor enters the `Next` clause for you. Change the `Next` clause to **Next spins** and press **Enter**.
6. The first instruction in the loop body should generate a random number from 0 through 5 and store it in the `leftIndex` variable. In the blank line between the For and Next clauses, enter the following assignment statement:

```
leftIndex = randGen.Next(0, 6)
```

7. The second instruction in the loop body should use the value in the `leftIndex` variable to display the appropriate image in the leftPictureBox. Enter the following assignment statement:

```
leftPictureBox.Image =
    ImageList1.Images.Item(leftIndex)
```

8. Next, the loop body should refresh the screen and pause the application. Enter the following two statements. Press **Enter** twice after typing the second statement.

```
Me.Refresh()
System.Threading.Thread.Sleep(50)
```

9. The next three instructions in the pseudocode are to generate a random number and store it in the `centerIndex` variable, then use the variable's value to display the appropriate image in the centerPictureBox, and then refresh the screen and pause the application. Enter the following statements. Press **Enter** twice after typing the last statement.

```
centerIndex = randGen.Next(0, 6)
centerPictureBox.Image =
    ImageList1.Images.Item(centerIndex)
Me.Refresh()
System.Threading.Thread.Sleep(50)
```

10. The last three instructions in the loop body are to generate a random number and store it in the `rightIndex` variable, then use the variable's value to display the appropriate image in the rightPictureBox, and then refresh the screen and pause the application. Enter the following statements:

```
rightIndex = randGen.Next(0, 6)
rightPictureBox.Image =
    ImageList1.Images.Item(rightIndex)
Me.Refresh()
System.Threading.Thread.Sleep(50)
```

11. Step 3 in the pseudocode is a single-alternative selection structure that determines whether the "Congratulations!" message should be displayed. Click the **blank line** below the `Next spins` clause and then press **Enter** to insert another blank line. Enter the following selection structure:

```
If leftIndex = centerIndex AndAlso
    leftIndex = rightIndex Then
    MessageBox.Show("Congratulations!", "Winner"
            MessageBoxButtons.OK,
            MessageBoxIcon.Information)
End If
```

12. The last step in the pseudocode is to enable the clickHereButton. Click immediately after the letter **f** in the End If clause and then press **Enter**. Enter the following assignment statement:

```
clickHereButton.Enabled = True
```

13. Save the solution and then start the application. Click the **Click Here** button. See Figure 7-37. Because random numbers determine the images assigned to the three picture boxes, your application might display different images than those shown in the figure. In addition, the "Congratulations!" message may appear in a message box on your screen. If necessary, click the **OK** button to close the message box.

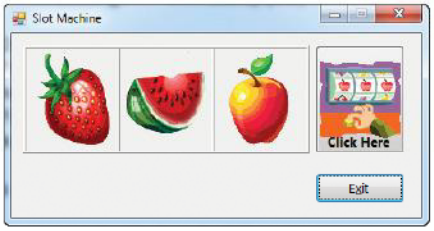
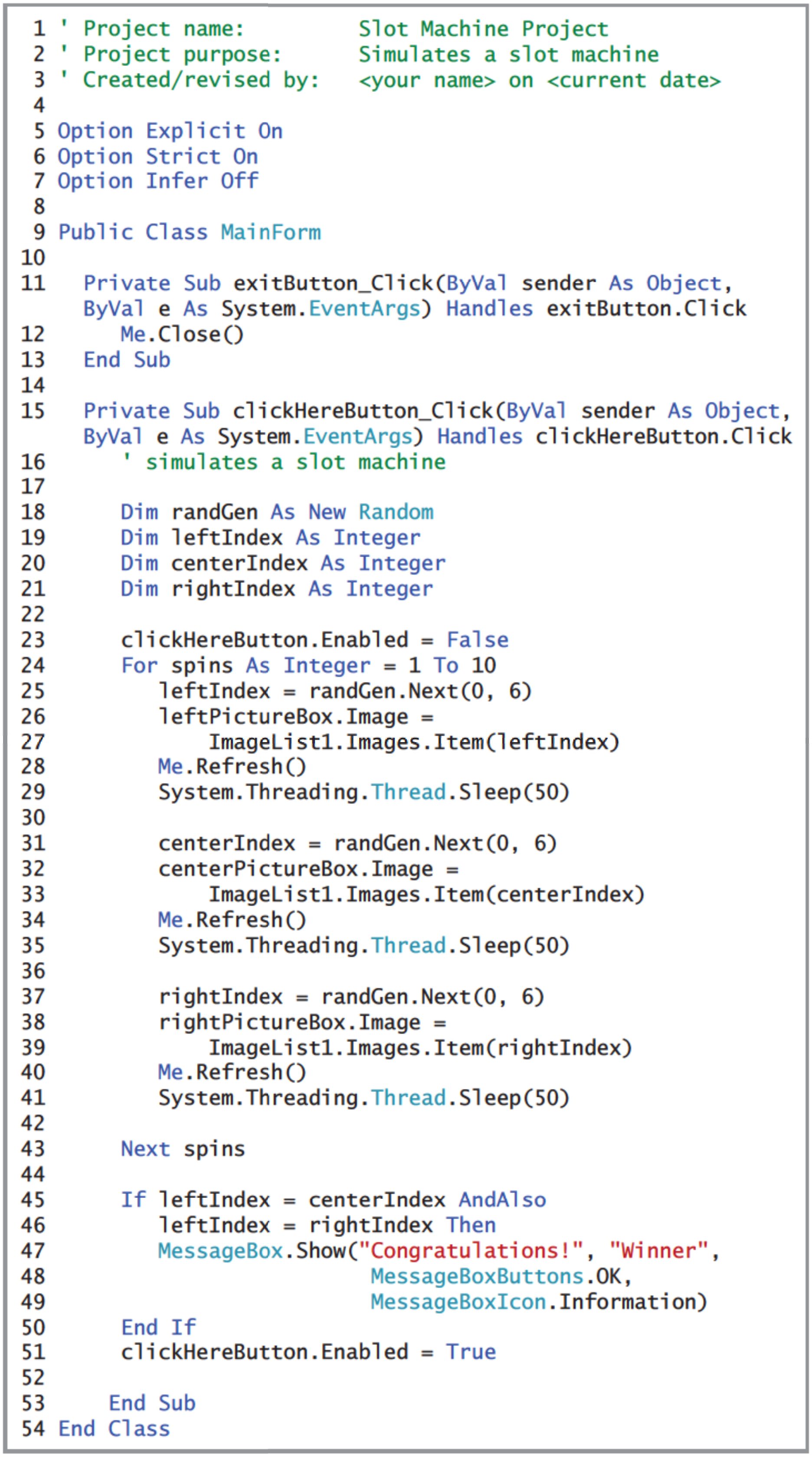


Figure 7-37 Result of clicking the Click Here button

14. Click the **Click Here** button until there is a winner. (You may need to click the button many times.) Then click the **OK** button to close the message box.

15. Click the **Exit** button to end the application. Close the Code Editor window, and then close the solution. Figure 7-38 shows the application's code.



```
' Project name:         Slot Machine Project
' Project purpose:      Simulates a slot machine
' Created/revised by:   <your name> on <current date>

Option Explicit On
Option Strict On
Option Infer Off

Public Class MainForm

  Private Sub exitButton_Click(ByVal sender As Object,
  ByVal e As System.EventArgs) Handles exitButton.Click
     Me.Close()
  End Sub

  Private Sub clickHereButton_Click(ByVal sender As Object,
  ByVal e As System.EventArgs) Handles clickHereButton.Click
     ' simulates a slot machine

     Dim randGen As New Random
     Dim leftIndex As Integer
     Dim centerIndex As Integer
     Dim rightIndex As Integer

     clickHereButton.Enabled = False
     For spins As Integer = 1 To 10
        leftIndex = randGen.Next(0, 6)
        leftPictureBox.Image =
            ImageList1.Images.Item(leftIndex)
        Me.Refresh()
        System.Threading.Thread.Sleep(50)

        centerIndex = randGen.Next(0, 6)
        centerPictureBox.Image =
            ImageList1.Images.Item(centerIndex)
        Me.Refresh()
        System.Threading.Thread.Sleep(50)

        rightIndex = randGen.Next(0, 6)
        rightPictureBox.Image =
            ImageList1.Images.Item(rightIndex)
        Me.Refresh()
        System.Threading.Thread.Sleep(50)

     Next spins

     If leftIndex = centerIndex AndAlso
        leftIndex = rightIndex Then
        MessageBox.Show("Congratulations!", "Winner",
                         MessageBoxButtons.OK,
                         MessageBoxIcon.Information)
     End If
     clickHereButton.Enabled = True

    End Sub
End Class
```

Figure 7-38 Code for the Slot Machine application