LESSON A

After studying Lesson A, you should be able to:

• Start and customize Visual Studio 2010 or Visual Basic 2010 Express
• Create a Visual Basic 2010 Windows application
• Manage the windows in the IDE
• Set the properties of an object
• Restore a property to its default setting
• Save a solution
• Close and open an existing solution

The Splash Screen Application

In this chapter, you will create a splash screen using Visual Basic 2010. As mentioned in the Overview, Visual Basic 2010 is available as a stand-alone product (called Visual Basic 2010 Express) or as part of Visual Studio 2010. Before you can use Visual Basic 2010 to create an application, you first must start either Visual Studio 2010 or Visual Basic 2010 Express.

To start Visual Studio 2010 or Visual Basic 2010 Express:

1. Click the Start button on the Windows 7 taskbar and then point to All Programs.
2. If you are using Visual Studio 2010, click Microsoft Visual Studio 2010 on the All Programs menu and then click Microsoft Visual Studio 2010. If the Choose Default Environment Settings dialog box appears, click Visual Basic Development Settings and then click Start Visual Studio.

   If you are using Visual Basic 2010 Express, click Microsoft Visual Studio 2010 Express on the All Programs menu and then click Microsoft Visual Basic 2010 Express.

3. Click Window on the menu bar, click Reset Window Layout, and then click the Yes button. When you start Visual Studio 2010 Professional, your screen will appear similar to Figure 1-2. When you start Visual Basic 2010 Express, your screen will appear similar to Figure 1-3.

   Important note: To select a different window layout, click Tools on the menu bar. If you are using the Express edition, point to Settings. Click Import and Export Settings, select the Reset all settings radio button, click the Next button, select the appropriate radio button, click the Next button, click the preferred settings collection, and then click the Finish button.
Next, you will configure Visual Studio or Visual Basic Express so that your screen agrees with the figures and tutorial steps in this book.

**To configure Visual Studio or Visual Basic Express:**

1. If you are using Visual Basic 2010 Express, click **Tools** on the menu bar, point to **Settings**, and then click **Expert Settings**.

2. Click **Tools** on the menu bar and then click **Options** to open the Options dialog box. If necessary, deselect the **Show all settings** check box. Click the **Projects and Solutions** node. Use the information
shown in Figure 1-4 to select and deselect the appropriate check boxes. (Your dialog box will look slightly different if you are using Visual Basic 2010 Express.) When you are finished, click the **OK** button to close the Options dialog box.

![Options dialog box](image)

**Figure 1-4** Options dialog box

The splash screen will be a Windows application, which means it will have a Windows user interface and run on a desktop computer. Recall that a user interface is what the user sees and interacts with while an application is running. Windows applications in Visual Basic are composed of solutions, projects, and files. A solution is a container that stores the projects and files for an entire application. Although the solutions in this book contain only one project, a solution can contain several projects. A project also is a container, but it stores only the files associated with that particular project.

**To create a Visual Basic 2010 Windows application:**

1. Click **File** on the menu bar and then click **New Project** to open the New Project dialog box.

2. If necessary, click **Visual Basic** in the Installed Templates list. *If you are using Visual Studio*, expand the Visual Basic node (if necessary) and then (if necessary) click **Windows**.

3. If necessary, click **Windows Forms Application** in the middle column of the dialog box.

4. Change the name entered in the Name box to **Splash Project**.

5. Click the **Browse** button to open the Project Location dialog box. Locate and then click the **VB2010\Chap01** folder. Click the **Select Folder** button to close the Project Location dialog box.

6. If necessary, select the **Create directory for solution** check box in the New Project dialog box. Change the name entered in the Solution name box to **Splash Solution**. Figures 1-5 and 1-6 show the completed New Project dialog box in Visual Studio 2010 Professional and Visual Basic 2010 Express, respectively. The drive letter will be different if you are saving to a device other than your computer’s hard drive—for example, if you are saving to a flash drive.
CHAPTER 1  An Introduction to Visual Basic 2010

7. Click the **OK** button to close the New Project dialog box. The computer creates a solution and adds a Visual Basic project to the solution. The names of the solution and project, as well as other information pertaining to the project, are recorded in the Solution Explorer window. See Figure 1-7. Notice that, in addition to the windows shown earlier in Figures 1-2 and 1-3, three other windows appear in the IDE: Windows Form Designer, Properties, and Data Sources. (If you are using Visual Basic 2010 Express, your title bar will say “Splash Solution – Microsoft Visual Basic 2010 Express. In addition, your screen will not have the Team Explorer window.)
Managing the Windows in the IDE

In most cases, you will find it easier to work in the IDE if you either close or auto-hide the windows you are not currently using. The easiest way to close an open window is to click the Close button on the window’s title bar. In most cases, the View menu provides an appropriate option for opening a closed window. Rather than closing a window, you also can auto-hide it. You auto-hide a window using the Auto Hide button (refer to Figure 1-7) on the window’s title bar. The Auto Hide button is a toggle button: clicking it once activates it, and clicking it again deactivates it. The Toolbox and Data Sources windows in Figure 1-7 are auto-hidden windows.

To close, open, auto-hide, and display windows in the IDE:

1. Click the Close button on the Properties window’s title bar to close the window. Now, click View on the menu bar and then click Properties Window to open the window.

2. If your IDE contains the Team Explorer window, click the window’s tab and then click the Close button on its title bar.

3. Click the Auto Hide (vertical pushpin) button on the Solution Explorer window. The Solution Explorer window is minimized and appears as a tab on the edge of the IDE.

4. To temporarily display the Solution Explorer window, place your mouse pointer on the Solution Explorer tab. The Solution Explorer window slides into view. Notice that the Auto Hide button is now a horizontal pushpin rather than a vertical pushpin.
5. Move your mouse pointer away from the Solution Explorer window. The window is minimized and appears as a tab again.

6. To permanently display the Solution Explorer window, place your mouse pointer on the Solution Explorer tab and then click the Auto Hide (horizontal pushpin) button on the window’s title bar. The vertical pushpin replaces the horizontal pushpin on the button.

7. On your own, close the Data Sources window.

8. Figure 1-8 shows the current status of the windows in the IDE. Only the Windows Form Designer, Solution Explorer, and Properties windows are open; the Toolbox window is auto-hidden. If necessary, click Form1.vb in the Solution Explorer window. If the items in the Properties window do not appear in alphabetical order, click the Alphabetical button.

In the next several sections, you will take a closer look at the Windows Form Designer, Solution Explorer, and Properties windows. (The Toolbox window is covered in Lesson B.)

The Windows Form Designer Window

Figure 1-9 shows the Windows Form Designer window, where you create (or design) the graphical user interface, referred to as a GUI, for your project. Only a Windows Form object appears in the designer window shown in the figure. A Windows Form object, or form, is the foundation for the user interface in a Windows application. You create the user interface by adding other objects, such as buttons and text boxes, to the form. Notice that a title bar appears at the top of the form. The title bar contains a default caption—in this case, Form1—as well as Minimize, Maximize, and Close buttons. At the top of the designer window is a tab labeled Form1.vb [Design]. [Design] identifies the window as the designer window. Form1.vb is the name of the...
file (on your computer’s hard disk or on another device) that contains the Visual Basic instructions associated with the form.

As you learned in the Overview, all objects in an object-oriented program are instantiated (created) from a class. A form, for example, is an instance of the Windows Form class. The form is automatically instantiated for you when you create a Windows application.

The Solution Explorer Window

The Solution Explorer window displays a list of the projects contained in the current solution and the items contained in each project. Figure 1-10 shows the Solution Explorer window for the Splash Solution, which contains one project named Splash Project. Within the Splash Project are the My Project folder and a file named Form1.vb. The project also contains other items, which typically are kept hidden. However, you can display the additional items by clicking the Show All Files button. You would click the button again to hide the items. The .vb on the Form1.vb filename indicates that the file is a Visual Basic source file. A source file is a file that contains program instructions, called code. The Form1.vb file contains the code associated with the form displayed in the designer window. You can view the code using the Code Editor window, which you will learn about in Lesson C.

The Form1.vb source file is referred to as a form file, because it contains the code associated with a form. The code associated with the first form included in a project is automatically stored in a form file named Form1.vb. The code
associated with the second form in the same project is stored in a form file named Form2.vb, and so on. Because a project can contain many forms and, therefore, many form files, it is a good practice to give each form file a more meaningful name. Doing this will help you keep track of the various form files in the project. You can use the Properties window to change the filename.

The Properties Window

As is everything in an object-oriented language, a file is an object. Each object has a set of attributes that determine its appearance and behavior. The attributes are called properties and are listed in the Properties window. When an object is created, a default value is assigned to each of its properties. The Properties window shown in Figure 1-11 lists the default values assigned to the properties of the Form1.vb file. (You do not need to widen your Properties window to match Figure 1-11.) As indicated in the figure, the Properties window includes an Object box and a Properties list. The Object box contains the name of the selected object. In this case, it contains Form1.vb, which is the name of the form file. The Properties list has two columns. The left column displays the names of the selected object’s properties. You can use the Alphabetical and Categorized buttons to display the names either alphabetically or by category, respectively. However, it’s usually easier to work with the Properties window when the properties are listed in alphabetical order, as they are in Figure 1-11. The right column in the Properties list is called the Settings box and displays the current value (or setting) of each of the properties. A brief description of the selected property appears in the Description pane.

To use the Properties window to change the form file’s name:

1. Form1.vb should be selected in the Solution Explorer window. Click File Name in the Properties list and then type Splash Form.vb. Be sure to include the .vb extension on the filename; otherwise, the computer will not recognize the file as a source file.

2. Press Enter. Splash Form.vb appears in the Solution Explorer and Properties windows and on the designer window’s tab, as shown in Figure 1-12.
Properties of a Windows Form

Like a file, a Windows form also has a set of properties. The form’s properties will appear in the Properties window when you select the form in the designer window.

To view the properties of the form:

1. Click the form in the designer window. The form’s properties appear in the Properties window.

2. If the properties are not listed alphabetically, click the Alphabetical button. The Properties window in Figure 1-13 shows a partial listing of the properties of a Windows form.

Notice that Form1 System.Windows.Forms.Form appears in the Object box in Figure 1-13. Form1 is the name of the form. The name is automatically assigned to the form when the form is instantiated (created). In System.Windows.Forms.Form, Form is the name of the class used to instantiate the form. System.Windows.Forms is the namespace that contains
the Form class definition. A **class definition** is a block of code that specifies (or defines) an object’s appearance and behavior. All class definitions in Visual Basic 2010 are contained in namespaces, which you can picture as blocks of memory cells inside the computer. Each **namespace** contains the code that defines a group of related classes. The System.Windows.Forms namespace contains the definition of the Windows Form class. It also contains the class definitions for objects you add to a form, such as buttons and text boxes. The period that separates each word in System.Windows.Forms.Form is called the **dot member access operator**. Similar to the backslash (\) in a folder path, the dot member access operator indicates a hierarchy, but of namespaces rather than folders. In other words, the backslash in the path C:\VB2010\Chap01\Splash Solution\Splash Project\Splash Form.vb indicates that the Splash Form.vb file is contained in (or is a member of) the Splash Project folder, which is a member of the Splash Solution folder, which is a member of the Chap01 folder, which is a member of the VB2010 folder, which is a member of the C drive. Likewise, the name System.Windows.Forms.Form indicates that the Form class is a member of the Forms namespace, which is a member of the Windows namespace, which is a member of the System namespace. The dot member access operator allows the computer to locate the Form class in the computer’s internal memory, similar to the way the backslash (\) allows the computer to locate the Splash Form.vb file on your computer’s disk.

**The Name Property**

As you do to a form file, you should assign a more meaningful name to a Windows form because doing so will help you keep track of the various forms in a project. Unlike a file, a Windows form has a Name property rather than a File Name property. You use the name entered in an object’s Name property to refer to the object in code, so each object must have a unique name. The name you assign to an object must begin with a letter and contain only letters, numbers, and the underscore character. The name cannot include punctuation characters or spaces. There are several conventions for naming objects in Visual Basic. In this book, you will use a naming convention called Hungarian notation. Names in Hungarian notation begin with a three (or more) character ID that represents the object’s type, with the remaining characters in the name representing the object’s purpose. For example, using Hungarian notation, you might assign the name frmSplash to the current form. The “frm” identifies the object as a form, and “Splash” reminds you of the form’s purpose. Hungarian notation names are entered using **camel case**, which means you enter the ID characters in lowercase and then capitalize the first letter of each subsequent word in the name. Camel case refers to the fact that the uppercase letters appear as “humps” in the name because they are taller than the lowercase letters.

**To change the name of the form:**

1. Drag the scroll box in the Properties window to the top of the vertical scroll bar. As you scroll, notice the various properties associated with a form. Also notice that the items within parentheses appear at the top of the Properties list.
2. Click **Name** in the Properties list. Type *frmSplash* and press **Enter**. An asterisk (*) appears on the designer window's tab. The asterisk indicates that the form has been changed since the last time it was saved.

**The Text Property**

In addition to changing the form's Name property, you also should change its Text property, which controls the text displayed in the form's title bar. The text also appears when you hover your mouse pointer over the application's button on the Windows 7 taskbar while the application is running. Form1 is the default value assigned to the Text property of the first form in a project. In this case, "Country Charm Inn" would be a more descriptive value.

**To set the Text property of the form:**

1. Scroll down the Properties window until you see the Text property in the Properties list and then click **Text**.

2. Type *Country Charm Inn* and press **Enter**. The new text appears in the property's Settings box and also in the form's title bar.

The Name and Text properties of a Windows form should always be changed to more meaningful values. The Name property is used by the programmer when coding the application. The Text property, on the other hand, is read by the user while the application is running.

**The StartPosition Property**

When an application is started, the computer uses the form's StartPosition property to determine the form's initial position on the screen. The frmSplash form represents a splash screen, which typically appears in the middle of the screen.

**To center a form on the screen when the application is started:**

1. Click **StartPosition** in the Properties list and then click the list arrow in the Settings box.

2. Click **CenterScreen** in the list.

**The Font Property**

A form's Font property determines the type, style, and size of the font used to display the text on the form. A font is the general shape of the characters in the text. Segoe UI, Tahoma, and Microsoft Sans Serif are examples of font types. Font styles include regular, bold, and italic. The numbers 9, 12, and 18 are examples of font sizes, which typically are measured in points, with one point equaling 1/72 of an inch. The recommended font for applications created for systems running Windows 7 (or Windows Vista) is Segoe UI, because it offers improved readability. Segoe is pronounced SEE-go, and UI stands for user interface. For most of the elements in the interface, you will use a font size of 9-point. However, to make the figures in the book more readable, some of the interfaces created in this book will use the 11-point Segoe UI font.
To set the form’s Font property:

1. Click Font in the Properties list and then click the…(ellipsis) button in the Settings box to open the Font dialog box.

2. Locate and then click the Segoe UI font in the Font box. Click 9 in the Size box and then click the OK button. (Don’t be concerned that the size of the form changed.)

The Size Property

As you can with any Windows object, you can size a form by selecting it and then dragging the sizing handles that appear around it. You also can size an object by selecting it and then pressing and holding down the Shift key as you press the up, down, right, or left arrow key on your keyboard. In addition, you can set the object’s Size property.

To set the form’s Size property:

1. Click Size in the Properties list. Notice that the Size property contains two numbers separated by a comma and a space. The first number represents the width of the form, measured in pixels. The second number represents the height, also measured in pixels. A pixel, which is short for “picture element,” is one spot in a grid of thousands of such spots that form an image either produced on the screen by a computer or printed on a page by a printer.

2. Type 685, 460 in the Size property’s Settings box and press Enter. Expand the Size property by clicking the arrow that appears next to the property. Notice that the first number listed in the property represents the width, and the second number represents the height. Click the arrow again to collapse the property.

Setting and Restoring a Property’s Value

In the next set of steps, you will practice setting and then restoring a property's value. More specifically, you will set and then restore the value of the form's BackColor property, which determines the background color of the form.

To set and then restore the form’s BackColor property value:

1. Click BackColor in the Properties list and then click the list arrow in the Settings box. Click the Custom tab and then click a red square to change the background color of the form to red.

2. Now, right-click BackColor in the Properties list and then click Reset on the context menu. The background color of the form returns to its default setting. Figure 1-14 shows the status of the form in the IDE.
Figure 1-14 Status of the form in the IDE

Saving a Solution

Notice the asterisk (*) that appears on the designer window’s tab in Figure 1-14. The asterisk indicates that a change was made to the form since the last time it was saved. It is a good practice to save the current solution every 10 or 15 minutes so that you will not lose a lot of your work if the computer loses power. You can save the solution by clicking File on the menu bar and then clicking Save All. You also can click the Save All button on the Standard toolbar. When you save the solution, the computer saves any changes made to the files included in the solution. Saving the solution also removes the asterisk that appears on the designer window’s tab.

To save the current solution:

1. Click File on the menu bar and then click Save All. No asterisk appears on the designer window’s tab, indicating that all changes made to the form have been saved.

Closing the Current Solution

When you are finished working on a solution, you should close it. Closing a solution closes all projects and files contained in the solution.

To close the Splash Solution:

1. Click File on the menu bar. Notice that the menu contains a Close option and a Close Solution option. The Close option does not close the solution; instead, it merely closes the designer window in the IDE. Only the Close Solution option closes the solution.

2. Click Close Solution. The Solution Explorer window indicates that no solution is currently open in the IDE.
Opening an Existing Solution

You can use the File menu to open an existing solution. If a solution is already open in the IDE, it is closed before another solution is opened. The names of solution files end with .sln.

To open the Splash Solution:

1. Click File on the menu bar and then click Open Project to open the Open Project dialog box.

2. Locate and then open the VB2010\Chap01\Splash Solution folder. Click Splash Solution (Splash Solution.sln) in the list of filenames and then click the Open button.

3. The Solution Explorer window indicates that the solution is open. If the designer window is not open, right-click Splash Form.vb in the Solution Explorer window and then click View Designer.

Exiting Visual Studio 2010 or Visual Basic 2010 Express

Finally, you learn how to exit Visual Studio 2010 or Visual Basic 2010 Express. You will complete the splash screen in the remaining two lessons. You can exit Visual Studio or Visual Basic Express using either the Close button on the title bar or the Exit option on the File menu.

To exit Visual Studio 2010 or Visual Basic 2010 Express:

1. Click File on the menu bar and then click Exit.

Lesson A Summary

- To start Visual Studio 2010 or Visual Basic 2010 Express:
  
  If you are using Visual Studio 2010, click the Start button, point to All Programs, click Microsoft Visual Studio 2010, and then click Microsoft Visual Studio 2010. If you are using Visual Basic 2010 Express, click the Start button, point to All Programs, click Microsoft Visual Studio 2010 Express, and then click Microsoft Visual Basic 2010 Express.

- To configure Visual Studio or Visual Basic Express:
  
  If you are using Visual Basic 2010 Express, click Tools, point to Settings, and then click Expert Settings. Click Tools, click Options, deselect the Show all settings check box, click the Projects and Solutions node, and then use the information shown earlier in Figure 1-4 to select and deselect the appropriate check boxes. Click the OK button.

- To create a Visual Basic 2010 Windows application:
  
  Start either Visual Studio 2010 or Visual Basic 2010 Express. Click File and then click New Project. If necessary, click Visual Basic in the Installed Templates list. If you are using Visual Studio, expand the Visual Basic node
(if necessary) and then (if necessary) click Windows. If necessary, click Windows Forms Application. Enter an appropriate name and location in the Name and Location boxes, respectively. Select the Create directory for solution check box. Enter an appropriate name in the Solution name box and then click the OK button.

- To reset the window layout in the IDE:
  Click Window, click Reset Window Layout, and then click the Yes button.

- To close and open a window in the IDE:
  Close the window by clicking the Close button on its title bar. Use the appropriate option on the View menu to open the window.

- To auto-hide a window in the IDE:
  Click the Auto Hide (vertical pushpin) button on the window’s title bar.

- To temporarily display an auto-hidden window in the IDE:
  Place your mouse pointer on the window’s tab.

- To permanently display an auto-hidden window in the IDE:
  Click the Auto Hide (horizontal pushpin) button on the window’s title bar.

- To set the value of a property:
  Select the object whose property you want to set and then select the appropriate property in the Properties list. Type the new property value in the selected property’s Settings box, or choose the value from the list, color palette, or dialog box.

- To give a more meaningful name to an object:
  Set the object’s Name property.

- To control the text appearing in the form’s title bar, as well as the text that appears when you hover your mouse pointer over the application’s button on the Windows 7 taskbar while the application is running:
  Set the form’s Text property.

- To specify the starting location of the form:
  Set the form’s StartPosition property.

- To specify the type, style, and size of the font used to display text on the form:
  Set the form’s Font property.

- To size a form:
  Drag the form’s sizing handles. You also can set the form’s Size, Height, and Width values in the Properties window. In addition, you can select the form and then press and hold down the Shift key as you press the up, down, left, or right arrow key on your keyboard.

- To change the background color of a form:
  Set the form’s BackColor property.
• To restore a property to its default setting:
  Right-click the property in the Properties list and then click Reset.

• To save a solution:
  Click File on the menu bar and then click Save All. You also can click the
  Save All button on the Standard toolbar.

• To close a solution:
  Click File on the menu bar and then click Close Solution.

• To open an existing solution:
  Click File on the menu bar and then click Open Project. Locate and then
  open the application's solution folder. Click the solution filename, which
  ends with .sln. Click the Open button. If the designer window is not open,
  right-click the form file's name in the Solution Explorer window and then
  click View Designer.

• To exit Visual Studio 2010 or Visual Basic 2010 Express:
  Click the Close button on the Visual Studio 2010 or Visual Basic 2010
  Express title bar. You also can click File on the menu bar and then click Exit.

Lesson A Key Terms

Camel case—used when entering object names in Hungarian notation;
the practice of entering the object’s ID characters in lowercase and then
capitalizing the first letter of each subsequent word in the name

Class definition—a block of code that specifies (or defines) an object’s
appearance and behavior

Code—program instructions

Dot member access operator—a period; used to indicate a hierarchy

Form—the foundation for the user interface in a Windows application; also
called a Windows Form object

Form file—a file that contains the code associated with a Windows form

GUI—graphical user interface

Namespace—a block of memory cells inside the computer; contains the code
that defines a group of related classes

Object box—the section of the Properties window that contains the name of
the selected object

Point—used to measure font size; 1/72 of an inch

Properties—the attributes that control an object’s appearance and behavior

Properties list—the section of the Properties window that lists the names of the
properties associated with the selected object, as well as each property’s value

Properties window—the window that lists an object’s attributes (properties)
**Settings box**—the right column of the Properties list; displays each property’s current value (setting)

**Solution Explorer window**—the window that displays a list of the projects contained in the current solution and the items contained in each project

**Source file**—a file that contains code

**Windows Form Designer window**—the window in which you create an application’s GUI

**Windows Form object**—the foundation for the user interface in a Windows application; referred to more simply as a form

---

**Lesson A Review Questions**

1. When a form has been modified since the last time it was saved, what appears on its tab in the designer window?
   a. an ampersand (&)
   b. an asterisk (*)
   c. a percent sign (%)
   d. a plus sign (+)

2. You use the ________ window to set the characteristics that control an object’s appearance and behavior.
   a. Characteristics
   b. Object
   c. Properties
   d. Toolbox

3. The ________ window lists the projects and files included in a solution.
   a. Object
   b. Project
   c. Properties
   d. Solution Explorer

4. The names of solution files in Visual Basic 2010 end with ________.
   a. .prg
   b. .sln
   c. .src
   d. .vb
5. Which of the following statements is true?
   a. You can auto-hide a window by clicking the Auto Hide (vertical pushpin) button on its title bar.
   b. An auto-hidden window appears as a tab on the edge of the IDE.
   c. You temporarily display an auto-hidden window by placing your mouse pointer on its tab.
   d. all of the above

6. The ________ property controls the text displayed in a form's title bar.
   a. Caption
   b. Text
   c. Title
   d. TitleBar

7. You give an object a more meaningful name by setting the object's ________ property.
   a. Application
   b. Caption
   c. Name
   d. Text

8. The ________ property determines the initial position of a form when the application is started.
   a. InitialLocation
   b. Location
   c. StartLocation
   d. StartPosition

9. Explain the difference between a form's Text property and its Name property.

10. Explain the difference between a form file and a form.

11. What does the dot member access operator indicate in the text System.Windows.Forms.Label?

Lesson A Exercises

1. If necessary, start Visual Studio 2010 or Visual Basic 2010 Express and permanently display the Solution Explorer window. Use the File menu to open the Charities Solution (Charities Solution.sln) file, which is contained in the VB2010\Chap01\Charities Solution folder.
If necessary, right-click the form file's name in the Solution Explorer window and then click View Designer. Change the form's Name property to frmMain. Change the form's BackColor property to light blue. Change the form's Font property to Segoe UI, 9pt. Change the form's StartPosition property to CenterScreen. Change the form's Text property to Charities Unlimited. Click File on the menu bar and then click Save All to save the solution. Click File on the menu bar and then click Close Solution to close the solution.

2. If necessary, start Visual Studio 2010 or Visual Basic 2010 Express and permanently display the Solution Explorer window. Create a Visual Basic Windows application. Use the following names for the solution, project, and form file, respectively: Photo Solution, Photo Project, and Main Form.vb. Save the application in the VB2010\Chap01 folder. Change the form's name to frmMain. The form's title bar should say Photos Incorporated; set the appropriate property. The form should be centered on the screen when it first appears; set the appropriate property. Change the background color of the form to light blue. Any text on the form should appear in the Segoe UI, 9pt font; set the appropriate property. Save and then close the solution.

3. If necessary, start Visual Studio 2010 or Visual Basic 2010 Express and permanently display the Solution Explorer window. Create a Visual Basic Windows application. Use the following names for the solution, project, and form file, respectively: Yorktown Solution, Yorktown Project, and Main Form.vb. Save the solution in the VB2010\Chap01 folder. Change the form's name to frmMain. The form's title bar should say Yorktown Shopping Center; set the appropriate property and then widen the form. The form should be centered on the screen when it first appears; set the appropriate property. Any text on the form should appear in the Segoe UI, 9pt font; set the appropriate property. Save and then close the solution.

4. In this exercise, you learn about a form's ControlBox, MaximizeBox, and MinimizeBox properties. If necessary, start Visual Studio 2010 or Visual Basic 2010 Express and permanently display the Solution Explorer window. Open the Greenwood Solution (Greenwood Solution.sln) file contained in the VB2010\Chap01\Greenwood Solution folder. If necessary, open the designer window.

   a. Use the Properties window to view the properties of the form. Click the ControlBox property. What is the purpose of this property? (Hint: Refer to the Description pane in the Properties window.) Set the ControlBox property to False. How does this setting affect the form? Set the ControlBox property to True.

   b. Click the MaximizeBox property. What is the purpose of this property? Set the MaximizeBox property to False. How does this setting affect the form? Set the MaximizeBox property to True.
c. Click the MinimizeBox property. What is the purpose of this property? Set the MinimizeBox property to False. How does this setting affect the form? Set the MinimizeBox property to True. Close the solution without saving it.

5. In this exercise, you research two properties of a form. If necessary, start Visual Studio 2010 or Visual Basic 2010 Express and permanently display the Solution Explorer window. Open the Greenwood Solution (Greenwood Solution.sln) file contained in the VB2010\Chap01\Greenwood Solution folder. If necessary, open the designer window. Use the Properties window to view the properties of the form. What property determines whether an icon is displayed in the form’s title bar? What property determines whether the form appears on the Windows taskbar when the application is running? Close the solution without saving it.