

# Chapter 1

## An Introduction to Visual Studio 2017 and Visual Basic

### A Guide to this Instructor's Manual:

We have designed this Instructor's Manual to supplement and enhance your teaching experience through classroom activities and a cohesive chapter summary.

This document is organized chronologically, using the same headings that you see in the textbook. Under the headings you will find: lecture notes that summarize the section, Teaching Tips, Class Discussion Topics, and Additional Projects and Resources. Pay special attention to teaching tips and activities geared towards quizzing your students and enhancing their critical thinking skills.

In addition to this Instructor's Manual, our Instructor's Resources also contain PowerPoint Presentations, Test Banks, and other supplements to aid in your teaching experience.

### At a Glance

### Instructor's Manual Table of Contents

- Overview
- Focus Objectives
- Teaching Tips
- Quick Quizzes
- Apply Objectives
- Teaching Tips
- Quick Quizzes
- Class Discussion Topics
- Additional Projects
- Additional Resources
- Key Terms

## Lecture Notes

### Overview

This chapter:

- Introduces definitions for many of the terms used by programmers
- Introduces Microsoft's newest integrated development environment: Visual Studio 2017
- Talks about the latest version of the Visual Basic programming language
- Begins to apply this knowledge in an effort to get students comfortable with the Visual Basic programming language

### Focus on the Concepts Lesson

- **F-1 Computer programming terminology**
- **F-2 The programmer's job**
- **F-3 The Visual Basic programming language**
- **F-4 The Visual Studio IDE**
- **F-5 Assigning names to objects**

### Teaching Tips

#### **F-1 Computer Programming Terminology**

1. Start with the definition of **Programming**: *giving a mechanism the directions to accomplish a task*. Explain the idea of a set of computer instructions being a **program**.
2. Discuss briefly how **programmers** use **programming language** to create computer programs. Some popular languages are C++, C#, and Java.

<b>Teaching Tip</b>	Run the completed program to demonstrate the final result that will be achieved in this chapter. It's important that students know what their goal is. This gives them a better understanding of <i>why</i> they are doing the things they are doing.
---------------------	---

3. Introduce the concept of an integrated development environment (IDE).
4. Talk about the importance of a **graphical user interface (GUI)** for the user.
5. Discuss how the various parts of the instruction set and the GUI create an **application**.

## F-2 The Programmers Job

1. The primary job of the Programmer is **coding**.
2. Discuss the way in which coding becomes an application.
3. Remind students that not everybody on a coding team has to be a programmer – to build accounting applications, you may need the expertise of an accountant.

## F-3 The Visual Basic Programming Language

1. Give a brief overview of Object Oriented Programming Languages and remind students that Visual Basic is one such language.
2. Discuss objects in regards to programming.
3. Introduce the terms **class**, **instance**, and **instantiated** and discuss how they relate to objects and programming.

<b>Teaching Tip</b>	Remind students that all objects come from classes. The Windows Form object is no exception.
---------------------	--

4. Have students run the **Payment.exe** application and discuss.
5. Have the students run the **Einstein.exe** application and discuss.

### Mini Quiz 1-1

1. What is a computer program?  
Answer: A set of instructions that tells a computer how to accomplish a task.
2. What is a GUI?  
Answer: Stands for Graphical User Interface. It is what the user sees and interacts with when using an application.
3. What is the process of translating a solution into a computer program called?  
Answer: Coding
4. In object-oriented programming, what is a class?  
Answer: A pattern used to instantiate (create)

## F-4 The Visual Studio IDE

1. Describe the four common windows used in Visual Studio 2017 seen in Figure 1-5:
  - a. The Designer Window
  - b. The Toolbox Window
  - c. The Properties Window
  - d. The Solution Explorer Window
2. Discuss **form**, **controls**, and **properties** in relation to the common windows.
3. Introduce the **Code Editor Window**.

## F-5 Assigning Names to Objects

1. Explain the **Name property** and how each object must have a name.
2. Go over naming rules
  - a. Each object must have a unique name
  - b. Each name must begin with a letter and contain only letters, numbers, and the underscore character

<b>Teaching Tip</b>	Remind students of the importance of using a naming convention for objects when designing applications. Point out that the naming convention followed in this book is to precede all control names with a three-character ID indicating the type of control, such as “lbl” for label controls.
---------------------	--

3. Go over the naming conventions being used in the textbook.

### Mini-Quiz 1-2

1. Which window in the IDE lists the tools you can use to add objects to a form?  
Answer: Toolbox
2. While designing an interface, which window in the IDE allows you to change the default value of an object’s property?  
Answer: Properties
3. Using the naming rules and conventions listed in Figure 1-7, which of the following are valid names and which are not? Explain why the names are not valid.
  - a. lblTotal
  - b. txtFirst.Name
  - c. lblCity&State
  - d. btnCalc Total
  - e. txtFirstQuarter

Answer: A and E are correct. B, C, and D are incorrect due to the use of invalid characters.

## **Apply the Concepts Lesson**

- A-1 Start and configure Visual Studio Community 2017
- A-2 Create a Windows Forms application
- A-3 Manage the windows in the IDE
- A-4 Change a form file's name
- A-5 Change the properties of a form
- A-6 Save a solution
- A-7 Close and open a solution
- A-8 Add a control to a form
- A-9 Use the Format menu
- A-10 Lock the controls on the form
- A-11 Start and end an application
- A-12 Enter code and comments in the Code Editor window
- A-13 Print an application's code and interface
- A-14 Exit Visual Studio and run an executable file

### **A-1 Start and Configure Visual Studio Community 2017**

1. Walk students through setting up Visual Studio Community and clearing the settings to defaults.
2. Walk students through the configuration process so that all students are working with the same settings in various menus.

### **A-2 Create a Windows Forms Application**

1. Walk students through the first steps of creating a Windows Form Application.
  - a. Click File, click New Project, click Visual Basic node (if necessary)
  - b. Click Windows Forms App (.NET Framework)
2. Have students change the name.

<b>Teaching Tip</b>	Point out the file extensions for the various files in the project and solution. Ensure that students understand that all of these files are required. It is a common mistake for new programmers to believe that only one file is required.
---------------------	--

3. Browse to find project location dialog.

**Teaching Tip**

A common problem for new programmers is misunderstanding where the files reside. To avoid overwriting files that have the same name, encourage students to create a new directory for each new project. Also, remind students that unlike other languages they may know, Visual Basic requires students to save the application *before* they start creating screens and writing code.

4. Alert students they may have to select the Create directory for solution.
5. Click OK button.

**A-3 Manage the Windows in the IDE**

1. Describe the major components of the IDE shown in Figures 1-12 and 1-13. Point out the location of the Solution Explorer window, the Properties window, and the Windows Form Designer window.
2. Remind students that you either close or auto-hide windows not currently being used.
3. Introduce the Auto Hide button and explain how it works.

**Teaching Tip**

Demonstrate how to use Auto Hide and how to force windows to stay open. It can be very disconcerting to students when windows appear and disappear seemingly without reason.

4. Describe the Alphabetical button in the Properties window.

**Teaching Tip**

Encourage students to use the Alphabetical button to quickly locate properties. If they do not use alphabetical order, they must know how Microsoft has grouped the properties.

**A-4 Change a Form File's Name**

1. Code associated with the first form included in a project is automatically stored in a file referred to as a **form file**, named Form1.vb.

**Teaching  
Tip**

Ensure that students understand the difference between a form object's name and the external filename containing the form. Also, distinguish between a form object's name and the text property of a form.

2. Any file with a .vb extension are known as **source files**.
3. Change the name by clicking **File Name** in the properties list and typing in a new name  
– Remind students to name this project **Main Form.vb**

**Mini-Quiz 1-3**

1. Which menu provides options for opening a closed window in the IDE?  
Answer: View
2. What is the name of the vertical pushpin button on a window's title bar?  
Answer: Auto Hide Button
3. What filename extension indicates that the file is a Visual Basic source file?  
Answer: .vb

**A-5 Change the Properties of a Form**

1. To change the properties of a form, the form must be selected in the designer window.
2. Click the **form** in the designer window.
3. Go over the various aspects of the **Properties** window with students.
4. Explain **class definition**, **namespace**, and **dot member access operator**.
5. Refer to Figure 1-16 for a list of commonly used properties.
  - a. Discuss the **name property**, the **font property**, and the **MaximizeBox**, **StartPosition**, and **Text Properties**.

**Teaching  
Tip**

Now is a good time to give a short lesson covering serif and sans serif fonts, and when to use each. Numerous studies have been conducted on the readability of serif versus sans serif typefaces without reaching a consistent conclusion. Some studies indicate that serif typefaces may be more readable in print, but there is no consensus. <http://en.wikipedia.org/wiki/Serif> - cite note-1 Studies of on-screen use are also ambiguous, suggesting that low screen resolutions make serifs more difficult to discern, with a resulting erosion of readability compared to sans serif fonts.

**Mini-Quiz 1-4**

1. What character is the dot member access operator?  
Answer: a period
2. What is the recommended font type for Windows 10 applications?  
Answer: Segoe UI
3. Which of a form's properties determines the location of the form when the application is started?  
Answer: StartPosition
4. To display the words ABC Company in a form's title bar, you need to set which of the form's properties?  
Answer: Text

**A-6 Save a Solution**

1. Remind students that an asterisk (\*) on the designer window tab means changes haven't been saved.
2. Recommend saving every 10 to 15 minutes to mitigate the amount of work lost in the event of a catastrophe.

**A-7 Close and Open a Solution**

1. Always close a solution when done working on it because it ensures that all the project files contained in the solution are also closed.
2. Only the **Close Solution** option under the **File** menu will close the solution.
3. Click **File** and then **Open Project** to open an existing solution.

**Teaching  
Tip**

Students who are new to Visual Studio often try to open their solutions directly from Windows Explorer. Because they are more likely to remember the name of the form or project than the solution, they often fail to open the existing solution and instead allow Visual Studio to create a new solution each time they open a form or project. It is a good idea to show students the .sln file and admonish them to use that file to open the solution from Windows Explorer.

**A-8 Add a Control to a Form**

1. Use the **Toolbox** tab and expand the **Common Controls** node if necessary.
2. Go over all the steps necessary to add a picture box to the form.
3. Go over all the steps necessary to add a button.
  - a. Provide the naming convention for buttons
  - b. Remind students that a button's Text property determines the text that appears on the button's face and should include an access key (discussed in Chapter 2)

**Mini-Quiz 1-5**

1. What is the three-character ID used when naming picture boxes?  
Answer: pic
2. What is the three-character ID used when naming buttons?  
Answer: btn
3. What is the purpose of an access key?  
Answer: It allows the user to access a control using the Alt key in combination with the control's access key
4. What character is used to designate an access key?  
Answer: &

**A-9 Use the Format Menu**

1. Go over the **Format menu** options in Figure 1-25.
2. Go over selecting controls before using the format menu.
3. Point out that the Format menu provides options to manipulate the controls on the form.
4. Introduce the term **reference control**.
5. Introduce the Align, Make Same Size, and Center in Form options, and describe the effects of each option.

6. Describe how to center a control horizontally on the form.

### **A-10 Lock the Controls on the Form**

1. Lock controls on the form once they are moved into place to stop accidental movement from occurring with errant clicks.
2. Go into the Format menu to access the Lock Controls option.
3. Always save after locking controls in place.

### **A-11 Start and End an Application**

1. Introduce the concept of a **startup form**.
2. Use Figure 1-27 to describe how to verify the startup form by using the project properties.
3. Introduce the concept of an **executable file**, and point out that this is what is normally given to the user.
4. Use Figure 1-28 to describe how to exit the application.

### **Mini-Quiz 1-6**

1. What menu provides options for centering two or more controls on the form?  
Answer: Format
2. To use the menu from Question 1 to make the btnCalc control the same size as the btnExit control, which of the two controls should you select first?  
Answer: btnExit
3. Why is it a good idea to lock the controls on the form?  
Answer: Doing so prevents you from inadvertently moving the control during design time.
4. What filename extension indicates that the file is an executable file that can be run outside of the IDE?  
Answer: .exe

### **A-12 Enter Code and Comments in the Code Editor Window**

1. Introduce the concept of **events** and **event procedures** in a Windows application.

<b>Teaching Tip</b>	Remind students that Windows is an event-driven operating system. Therefore, a Windows application must be written to respond to events that are initiated by user actions.
---------------------	---

- Describe how to open the Code Editor window in Visual Studio.
- Use Figure 1-29 to describe the code that will be placed in the Code Editor window by default.

<b>Teaching Tip</b>	Point out that students should name objects before writing code so that the event procedure names will match the object names.
---------------------	--

- Use Figure 1-30 to explain how to expand and collapse a region of code.
- Use Figure 1-31 to introduce the concept of a code template for event procedures.
- Introduce the term **syntax**, and describe the syntax of the **procedure header** and **procedure footer**.
- Introduce the concept of a **keyword**, and explain the effects of the keyword `Private`.
- Introduce the concept of a **sub procedure**, and describe the required syntax elements of a sub procedure.

<b>Teaching Tip</b>	Spend some time discussing the color coding of text that appears in the Code Editor window. Ensure that students understand that this provides another way for them to validate that they have used the correct keywords and syntax.
---------------------	--

- Introduce the term **method**.
- Introduce the `Me.Close()` and its application in the `btnExit_Click` procedure for closing the current form.

<b>Teaching Tip</b>	Ensure that students understand the distinction between closing a form and exiting the application.
---------------------	---

- Use Figures 1-32 and 1-33 to demonstrate the use of IntelliSense to code the `Me.Close()` instruction.

12. Stress the importance of always testing a program after writing code.
13. Introduce Assignment Statements and Comments.
14. Discuss the terms **run time** and **assignment statement**.
15. Go over the **assignment operator**.
16. Discuss the use of **comments** in coding and how they can help clarify the code for other coders. Remind students that comments have no effect on the code itself.

## A-14 Exit Visual Studio and Run an Executable File

1. Demonstrate closing out of Visual Studio.
2. Demonstrate opening an Executable file, such as the one they just completed following the instructions of this chapter.

### Mini-Quiz 1-7

1. What is an event procedure?  
Answer: Answer: A set of instructions processed only when its associated event occurs.
2. What is a keyword?  
Answer: A word that has a special meaning in a programming language.
3. If an application contains only one form, what Visual Basic statement tells the computer to end the application?  
Answer: Me.Close()
4. Write an assignment statement that assigns the keyword False to the btnPrint control's Enabled property.  
Answer: btnPrint.Enabled = False
5. In the Code Editor window, what character designates that the text that follows it is a comment?  
Answer: an apostrophe

## Summary

- Programs are the step-by-step instructions that tell a computer how to perform a task.
- Programmers use various programming languages to communicate with the computer.
- Programmers are responsible for translating a problem's solution into instructions that the computer can understand.

- Programmers rigorously test a program before releasing the program to the user.
- An object-oriented programming language, such as Visual Basic, enables programmers to use objects (for example, check boxes and buttons) to accomplish a program's goal. An object is anything that can be seen, touched, or used.
- Every object in an object-oriented program is instantiated (created) from a class, which is a pattern that tells the computer how the object should look and behave. An object is referred to as an instance of the class.
- The four windows you use most often when designing your application's GUI are the designer window, Toolbox window, Solution Explorer window, and Properties window.
- Each tool in the toolbox represents a class.
- Each object has a set of properties that determines its appearance and behavior.
- Windows applications in Visual Basic are composed of solutions, projects, and files.
- You enter your application's program instructions in the Code Editor window.
- An object's name, which is entered in its Name property, can be used to refer to the object in code.
- To start Visual Studio 2017:  
Click the Start button on the Windows 10 taskbar. Locate and then click Visual Studio 2017.
- To change the default environment settings:  
Click Tools, 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 settings collection you want to use, click the Finish button, and then click the Close button to close the Import and Export Settings Wizard dialog box.
- To reset the window layout in the IDE:  
Click Window, click Reset Window Layout, and then click the Yes button.
- To configure Visual Studio:  
Click Tools, click Options, click the Projects and Solutions node, and then use the information shown earlier in Figure 1-9 to select and deselect the appropriate check boxes. Next, expand the Text Editor node, expand the Basic node, expand the Code Style node, click Naming, and then change the Severity column in the Types and Non-Field Members rows to None. Click the Code Style node, and then change the Preference column in the Qualify property access with 'Me' and Qualify method access with 'Me' rows to Prefer 'Me.' Finally, click the Debugging node, deselect the Step over properties and operators (Managed only) check box, deselect the Enable Diagnostic

tools while debugging check box, deselect the Show elapsed time PerfTip while debugging check box, and then click the OK button to close the Options dialog box.

- To create a Visual Basic 2017 Windows Forms application:  
Start Visual Studio 2017. Click File, click New Project, click the Visual Basic node, and then click Windows Forms App (.NET Framework). Enter an appropriate name and location in the Name and Location boxes, respectively. If necessary, select the Create directory for solution check box. Enter an appropriate name in the Solution name box and then click the OK button.
- To size a window:  
Position your mouse pointer on one of the window's borders and then drag the border.
- 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:  
Click the window's tab.
- To permanently display an auto-hidden window in the IDE:  
Click the window's tab to display the window, and then click the Auto Hide (horizontal pushpin) button on the window's title bar.
- The code associated with a form is stored in a form file whose filename ends with .vb. All files with a .vb filename extension are referred to as source files because they contain Visual Basic code.
- To set the value of a property:  
Select the object whose property you want to set and then click 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 change the name of a form file:  
Set the form file's File Name property in the Properties window.
- To change an object's name:  
Set the object's Name 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 disable/enable the form's Maximize button:  
Set the form's MaximizeBox property.

- To specify the starting location of the form:  
Set the form's `StartPosition` property.
- To control the text appearing in the form's title bar:  
Set the form's `Text` property.
- To save a solution:  
Click **File** on the menu bar and then click **Save All**. You can also 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 file name, 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 add a control to a form:  
Click a tool in the toolbox, but do not release the mouse button. Hold down the mouse button as you drag the tool to the form, and then release the mouse button. As demonstrated in the **Ch01-Adding a Control** video, you can also click a tool and then click the form. In addition, you can click a tool, place the mouse pointer on the form, and then press the left mouse button and drag the mouse pointer until the control is the desired size. You can also double-click a tool in the toolbox.
- To display a graphic in a control in the user interface:  
Use the **PictureBox** tool to instantiate a picture box control. Use the task box or **Properties** window to set the control's `Image` and `SizeMode` properties.
- To display a standard button that performs an action when clicked:  
Use the **Button** tool to instantiate a button control.
- An access key allows the user to select a control by pressing the **Alt** key in combination with the access key.
- To assign an access key to a button:  
In the button's `Text` property, precede the access key character with an ampersand (`&`).
- To select multiple controls on a form:  
Click the first control you want to select, and then **Ctrl+click** each of the other controls you want to select. As demonstrated in the **Ch01-Format Menu** video, you can also select a group of controls on the form by placing the mouse pointer slightly above and to the left of the first control you want to select, and then pressing the left mouse button and dragging. A dotted rectangle appears as you drag. When all of the controls

you want to select are within (or at least touched by) the dotted rectangle, release the mouse button. All of the controls surrounded or touched by the dotted rectangle will be selected.

- To cancel the selection of one or more controls:  
You cancel the selection of one control by pressing and holding down the Ctrl key as you click the control. You cancel the selection of all of the selected controls by releasing the Ctrl key and then clicking the form or any unselected control on the form.
- To make two or more controls on the form the same size:  
Select the reference control and then select the other controls you want to size. Click Format on the menu bar, point to Make Same Size, and then click the appropriate option.
- To align the borders of two or more controls on the form:  
Select the reference control and then select the other controls you want to align. Click Format on the menu bar, point to Align, and then click the appropriate option.
- To lock/unlock the controls on a form:  
Click the form (or any control on the form), click Format, and then click Lock Controls. Or you can right-click the form (or any control on the form) and then click Lock Controls on the context menu.
- To verify or change the names of the startup form and executable file:  
Use the Application pane in the Project Designer window. You can open the Project Designer window by right-clicking My Project in the Solution Explorer window, and then clicking Open on the context menu. Or, you can click Project on the menu bar and then click *<project name> Properties* on the menu.
- To start and stop an application:  
You can start an application by clicking Debug on the menu bar and then clicking Start Debugging. You can also press the F5 key on your keyboard or click the Start button on the Standard toolbar. You can stop an application by clicking the form's Close button. You can also make the designer window the active window, then click Debug, and then click Stop Debugging.
- When you start a Visual Basic application, the computer automatically creates an executable file that can be run outside of the IDE. The file's name ends with .exe.
- You tell an object how to respond to the user's actions by writing an event procedure, which can contain comments and Visual Basic statements.
- To open the Code Editor window:  
Right-click the form and then click View Code on the context menu.
- To show/hide the line numbers in the Code Editor window:

Click Tools, click Options, expand the Text Editor node, click Basic, select/deselect the Line numbers check box, and then click OK.

- To collapse or expand a region of code in the Code Editor window:  
Click the minus box next to the region of code to collapse the code; click the plus box to expand the code.
- To display an object's event procedure in the Code Editor window:  
Open the Code Editor window. Use the Object list box to select the object's name, and then use the Event list box to select the event.
- To allow the user to close the current form while an application is running:  
Enter the `Me.Close()` statement in an event procedure.
- To turn on/off the IntelliSense feature in the Code Editor window:  
Click Tools, click Options, expand the Text Editor node, click Basic, select/deselect the Auto list members check box, and then click the OK button.
- During run time, you can use an assignment statement to assign a value to an object's property.
- All comments in the Code Editor window begin with an apostrophe.
- To print an application's code:  
With the Code Editor window the active window, click File on the menu bar and then click Print. Select the printer (if necessary) and then click Print.
- To print an application's interface:  
As demonstrated in the Ch01-Snipping Tool video, make the designer the active window. Click the Start button on the Windows 10 taskbar, click Windows Accessories, and then click Snipping Tool. Click the New button. Drag your mouse pointer around the form and then release the mouse button. Click File, select the printer (if necessary), and then click Print.
- To exit Visual Studio:  
Click the Close button on the Visual Studio title bar. You can also click File on the menu bar and then click Exit.
- To run an application's executable (.exe) file outside of the IDE:  
Locate the file in the project's bin\Debug folder and then double-click it.

## **Class Discussion Topics**

1. Ask students to describe some applications they think are well made. What do they feel is important in a well-made application? What aspects of an application make it bad?

2. What other uses may there be for hiding and showing information in an application? Thinking beyond the simple Einstein Application, what higher level uses may there be for such a simple action?

## **Additional Projects**

1. Ask students to design a new Application that shows and hides an image of their families.
2. Show students a screen that contains many objects (labels, check boxes, text boxes, etc.) and have them tell how many objects they can count. Even after telling them that *everything* they see is an object, they most likely will not count correctly.
3. Ask students to discuss what they've learned about how Visual Basic functions as a programming language. Discuss objects and properties extensively.

## **Additional Resources**

1. Tools for Visual Studio 2017:  
<https://docs.microsoft.com/en-us/visualstudio/welcome-to-visual-studio>
2. Microsoft Consulting Services Naming Conventions for Visual Basic:  
<http://support.microsoft.com/kb/110264>
3. Visual Studio 2015 Getting Started Tutorials:  
[https://msdn.microsoft.com/en-us/library/dd492171\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/dd492171(v=vs.140).aspx)
4. Visual Studio 2017 Release Notes:  
<https://www.visualstudio.com/en-us/news/releasenotes/vs2017-relnotes>

## **Key Terms**

- **Application**—a GUI along with its program instructions
- **Assignment operator**—the equal sign in an assignment statement
- **Assignment statement**—an instruction that assigns a value to something, such as to the property of an object
- **Camel case**—used when entering object names; 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**—a pattern that the computer uses to create (instantiate) an object
- **Class definition**—a block of code that specifies (or defines) an object's appearance and behavior
- **Code**—program instructions
- **Code Editor window**—where you enter the program instructions (code) for your application
- **Coding**—the process of translating a solution into a language that the computer can understand
- **Comment**—a line of text that serves to internally document a program; begins with an apostrophe
- **Computer program**—the directions given to computers; also called a *program*
- **Controls**—objects (such as a picture box or a button) added to a form
- **Design time**—occurs when you are building an interface
- **Designer window**—used to create an application's GUI
- **Dot member access operator**—the period used to indicate a hierarchy
- **Event procedure**—a set of Visual Basic instructions that tell an object how to respond to an event
- **Events**—actions to which an object can respond; examples include clicking and double-clicking
- **Executable file**—a file that can be run outside of the IDE; the file name ends with the .exe extension
- **Form**—the foundation for the user interface in a Windows Forms application; also called a *Windows Form object*
- **Form file**—a file that contains the code associated with a Windows form

- **Graphical user interface**—what the user sees and interacts with while your program is running; also called a *GUI*
- **GUI**—graphical user interface
- **IDE**—integrated development environment
- **Instance**—an object created (instantiated) from a class
- **Instantiated**—the process of creating an object from a class
- **Integrated development environment**—an environment that contains all of the tools and features you need to create, run, and test your programs; also called an *IDE*
- **Keyword**—a word that has a special meaning in a programming language
- **Me.Close() statement**—tells the computer to close the current form
- **Method**—a predefined procedure that you can call (invoke) when needed
- **Name property**—assigns a name to an object; the name can be used to refer to the object in code
- **Namespace**—a block of memory cells inside the computer; contains the code that defines a group of related classes
- **Object**—anything that can be seen, touched, or used
- **Object-oriented programming language**—a programming language that allows the programmer to use objects to accomplish a program's goal
- **OOP**—object-oriented programming
- **Picture box**—used to display an image
- **Procedure footer**—the last line in a procedure
- **Procedure header**—the first line in a procedure
- **Program**—the directions given to computers; also called a *computer program*

- **Programmers**—the people who write computer programs
- **Programming**—the process of giving a mechanism the directions to accomplish a task
- **Programming languages**—languages used to communicate with a computer
- **Properties**—the attributes that determine an object’s appearance and behavior
- **Properties window**—lists an object’s attributes (properties)
- **Reference control**—the first control selected in a group of controls; this is the control whose size and/or location you want the other selected controls to match
- **Run time**—occurs while an application is running
- **Solution Explorer**—displays a list of the projects contained in the current solution and the items contained in each project
- **Source files**— files that contain program instructions; in Visual Basic, the names of source files end with .vb
- **Startup form**—the form that appears automatically when an application is started
- **Statements**—Visual Basic instructions that are processed (executed) by the computer
- **String literal**—zero or more characters enclosed in quotation marks
- **Sub procedure**—a block of code that performs a specific task
- **Syntax**—the rules of a programming language
- **Toolbox window**—contains the tools used when creating an interface (each tool represents a class); referred to more simply as the *toolbox*