

Programming with Microsoft® Visual Basic® 2017

An Introduction to Visual Studio 2017 and Visual Basic®

Chapter 1 – Lessons A and B



Chapter 1 – Concepts Lesson

- Concepts covered in this lesson:
 - Computer programming terminology
 - The programmer's job
 - The Visual Basic programming language
 - The Visual Studio IDE
 - Assigning names to objects

Computer Programming Terminology

- In essence, the word programming means *giving a mechanism the directions to accomplish a task*.
- A set of instructions that tells a computer how to accomplish a task is called a computer program or, more simply, a program.
- Programs are written by programmers using a variety of special languages called programming languages.
- An integrated development environment (IDE) is an environment that contains all of the tools and features you need to create, run, and test your programs.

Computer Programming Terminology

- A graphical user interface (or GUI) is what the person using your program (referred to as the user) sees and interacts with while your program is running.
- The user interface and its program instructions are referred to as an application.

```
1  Public Class frmMain
2      Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click
3          Me.Close()
4      End Sub
5
6      Private Sub tmrGetUp_Tick(sender As Object, e As EventArgs) Handles tmrGetUp.Tick
7          ' Blink the message.
8
9          lblMessage.Visible = Not lblMessage.Visible
10     End Sub
```

Good Morning Application

- Use Windows to locate and then open the VB2017\Chap01 folder on your computer's hard disk or on the device designated by your instructor.
- Double-click Good Morning.exe in the list of filenames.
- The application's user interface appears on the screen with a blinking "It's time to get up!!!!" message.
- Click the Exit button to close the application.

The Programmer's Job

- First, the programmer meets with the person responsible for describing the problem.
- The software developer will meet with the user and then outline the problem specification for the programmer.
- After the planning is complete, the programmer will translate the solution into computer instructions—a process called coding.
- The programmer then will test the program rigorously with sample data to make sure it works both correctly and to the user's satisfaction.

The Visual Basic Programming Language

- Visual Basic is an object-oriented programming language, which is a language that allows the programmer to use objects to accomplish a program's goal.
- In object-oriented programming, or OOP, an object is anything that can be seen, touched, or used.
- Every object in an object-oriented program is created from a class, which is a pattern that the computer uses to create the object.
- An object created from a class is called an instance of the class and is said to be instantiated from the class.

The Visual Studio IDE

- The Visual Studio IDE contains many different windows, each with its own special purpose.
- The designer window is where you create (or design) your application's GUI.
- A form is the foundation for the user interface in an application created for the Windows environment.
- You use the Toolbox window to add other objects, called controls, to the form.
- Each tool listed in the Toolbox window represents a class.

The Visual Studio IDE

- Each object has a set of attributes that determines its appearance and behavior.
- The attributes, called properties, are listed in the Properties window when the object is selected in the designer window.
- A project is also a container, but it stores only the files associated with that particular project.
- The Solution Explorer window displays a list of the projects contained in the current solution and the items contained in each project.

The Visual Studio IDE

- The Solution Explorer window indicates that the Einstein Solution contains the Einstein Project, which contains several items.
- The Einstein.png and Equation.png items are the names of files on your disk.
- These files contain the images that appear in the picture boxes on the form. The Main Form.vb item is also the name of a file on your disk.
- The Main Form.vb file stores the program instructions (code) that tell the three buttons how to respond when the user clicks them.

Assigning Names to Objects

- Each object has a set of properties attached to it.
- One of the most important of these properties is the Name property.
- This is because you use the Name property to refer to the object in code.
- This is because you use the Name property to refer to the object in code.

Assigning Names to Objects

Naming rules (these are required by Visual Basic)

1. Each object must have a unique name.
2. Each name must begin with a letter and contain only letters, numbers, and the underscore character.

Naming conventions used in this book

1. Each name will begin with an ID of three (or more) characters that represents the object's type—for example, *frm* for a form, *btn* for a button, and *txt* for a text box.
2. The remaining characters after the ID will indicate the object's purpose.
3. Each name will be entered using camel case: the ID will be in lowercase, and the first letter in each subsequent word in the name will be capitalized.

Examples

| | |
|---------------------------|---|
| <code>frmMain</code> | the main form in a project |
| <code>btnExit</code> | a button that ends the application when clicked |
| <code>txtFirstName</code> | a text box for entering a customer's first name |
| <code>picEquation</code> | a picture box that displays an image of an equation |
| <code>lblTotalDue</code> | a label that displays the total amount a customer owes |
| <code>chkDiscount</code> | a check box for specifying whether a customer gets a discount |

Chapter 1 – Apply the Concepts

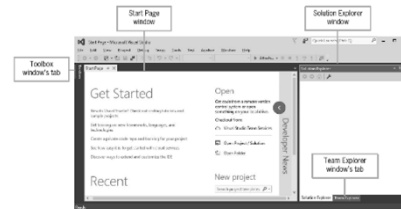
- After studying this lesson, you should be able to:
 - Start and configure Visual Studio Community 2017
 - Create a Windows Forms application
 - Manage the windows in the IDE
 - Change a form file's name
 - Change the properties of a form
 - Save a solution
 - Close and open a solution
 - Add a control to a form

Chapter 1 – Apply the Concepts

- Use the Format menu
- Lock the controls on the form
- Start and end an application
- Enter code and comments in the Code Editor window
- Print an application's code and interface
- Exit Visual Studio and run an executable file

Start and Configure Visual Studio Community 2017

- To start Visual Studio 2017:
 - Click Start and point to All Programs
 - Select Microsoft Visual Studio 2017
 - Click Tools on the menu bar
 - Click Import and Export Settings
 - Select the Reset all settings radio button
 - Click the Next button
 - Select the No, just reset settings, overwriting my current settings radio button
 - Click the Next button
 - Click Visual Basic
 - Click the Finish button.
 - Click the Close button to close the Import and Export Settings Wizard dialog box.



Start and Configure Visual Studio Community 2017

- Click Window on the menu bar
- Click Reset Window Layout
- Click the Yes button.
- Your menu bar might not contain the underlined letters, called access keys. You can show/hide the access keys by pressing the Alt key on your keyboard.

To configure Visual Studio 2017

- Click Tools on the menu bar
- Click Options to open the Options dialog box
- Click the Projects and Solutions node
- Expand the Text Editor node
- Expand the Basic node
- Expand the Code Style node
- Click Naming
- Locate the Types row in the dialog box
- Click the list arrow in its Severity column.

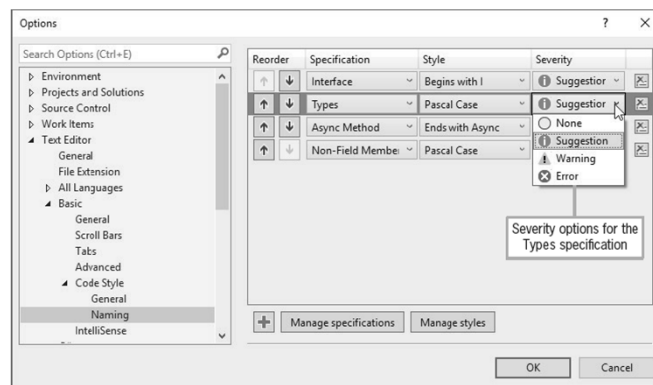
To configure Visual Studio 2017

- Click None in the list
- Locate the Non-Field Members row and change its Severity column entry to None
- Click the Code Style node
- Locate the Qualify property access with 'Me' row
- Click the list arrow in its Preference column
- Click Prefer 'Me.'
- Locate the Qualify method access with 'Me' row and change its Preference column entry to Prefer 'Me.'
- Scroll down the left column of the dialog box until you locate the Debugging node

To configure Visual Studio 2017

- Click the Debugging node
- Deselect the Step over properties and operators (Managed only) check box
- Also deselect the Enable Diagnostic Tool while debugging check box (if necessary) and the Show elapsed time PerfTip while debugging check box
- Click the Projects and Solutions node
- Select Save new projects when created
- Click the OK button to close the Options dialog box

To configure Visual Studio 2017



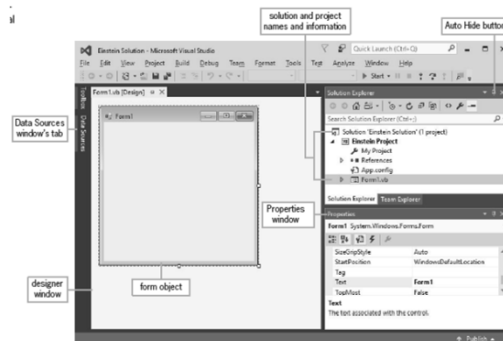
Create a Windows Forms Application

- To create a Visual Basic 2017 Windows Application:
 - Click FILE on the menu bar, and then click New Project
 - If necessary, click the Visual Basic node in the Installed Templates list, and then click Windows Forms App (.NET Framework)
 - Enter a name for the project in the Name Box
 - Locate a folder to store the project
 - Click the Select Folder button to close the project

Create a Windows Forms Application

- 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
- 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, along with other information pertaining to the project, appear in the Solution Explorer window.
- Visual Basic also automatically instantiates (creates) a form object, which appears in the designer window.

Create a Windows Forms Application



Managing the Windows in the IDE

- To open a window:
 - Click VIEW on the menu bar, and then click Properties Window or
 - Mouseover the Solution Explorer tab if the window is auto-hidden
- To close a window:
 - Use the Close button on the window's title bar
- To minimize a window to a tab:
 - Click the Auto Hide (push-pin) button on the Solution Explorer window

Manage the Windows in the IDE

- In the Solution Explorer window, the name of the selected object appears in the Properties window's Object box.
- The Properties window also contains two columns of information:
 - The left column, called the Properties list, displays the names of the selected object's properties.
 - The right column contains the Settings boxes; each box displays the current value (or setting) of its associated property.

Manage the Windows in the IDE

- If the names listed in the Properties list do not appear in alphabetical order, click the Alphabetical button, which is the second button on the Properties window's toolbar.
- Only the designer, Solution Explorer, and Properties windows are open; the Toolbox window is auto-hidden.
- (If necessary, close any other open or auto-hidden windows in the IDE.)

Change a Form File's Name

- The 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.
- All files with a .vb filename extension are also referred to as source files because they contain Visual Basic code.
- The code associated with the second form in the same project is stored in a file named Form2.vb, and so on.

Change a Form File's Name

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

- Click File Name in the Properties list for the Form1.vb file. Type Main Form.vb in the Settings box and press Enter.
- Main Form.vb appears in the Solution Explorer and Properties windows and on the designer window's.

Windows Form Designer Window

- Used to create a graphical user interface (GUI)
- Windows Form Object
 - Create a UI by adding objects
 - Buttons
 - Text boxes
 - The title bar contains a default caption
 - Also minimize, maximize, and close buttons

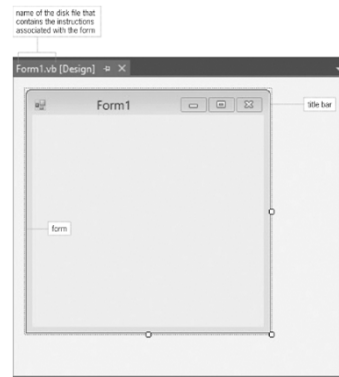


Figure 1-7 Windows Form Designer window

The Solution Explorer Window

- Displays a list of project contained in the current solution and items contained in each project
- Source file
 - A file containing program instructions (code)
- Form file
 - Contains code associated with a form
 - The first form's code is stored in Form1.vb
 - The second form's code is stored in Form2.vb

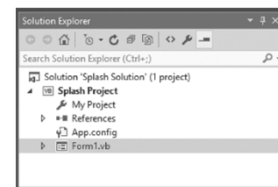


Figure 1-8 Solution Explorer window

The Properties Window

- Properties are the attributes of objects that determine appearance and behavior
- Object box
 - Contains the name of the selected object
- Properties list
 - Helps identify and access properties
- Settings box
 - Shows the current values of properties



Figure 1-9 Properties window

Properties of a Windows Form

- A class definition is a block of code that specifies (or defines) an object's appearance and behavior.
- Each namespace contains the code that defines a group of related classes.
- Namespace contains the definition of the Windows Form class and 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.

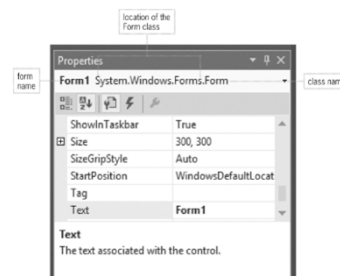


Figure 1-11 Properties window showing a partial listing of the form's properties

The Name Property

- Windows forms have a Name property
 - Use it to refer to the form in the code
 - Naming conventions are used for clarity
 - Hungarian notation
 - frmSpash ("frm" indicates a form, and "Splash" is the form's purpose)
 - Camel case
 - Enter ID characters in lowercase and then capitalize the first letter of each subsequent word in the name

The Font Property

- A font is the general shape of a character
- Font styles include regular, bold, and italic
- Font sizes (9, 12, 18, etc.) are in points, with 1 point equaling $\frac{1}{72}$ of an inch
- The recommended font for applications is Segoe (pronounced SEE-go)
 - This is Segoe UI, Size 17

To change the form's Font property:

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

The MaximizeBox Property

- The MaximizeBox property controls whether the maximize button is enabled, meaning the user can maximize the window to fill the screen, or disabled meaning they can't.
- To disable, set the MaximizeBox property to False.

The StartPosition Property

- The computer uses the form's StartPosition property to determine the form's initial position on the screen.
 - Splash screens usually appear in the middle of the screen
- Choose CenterScreen for the StartPosition

The Text Property

- The Text property controls the text displayed in the form's title bar
 - The default text for the Text property of Form 1 is Form1
- Change the default name to something meaningful
 - The Name property is used by the programmer
 - The Text property is read by the user when the application is running

The Size Property

- A form can be sized by 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 the keyboard
- In addition, you can set the object's Size property, which contains two numbers, measured in pixels:
 - The first number represents the width of the form
 - The second number represents the height of the form

Restoring a Property's Value

- Right-click the name of the property in the Properties list and then click Reset on the context menu

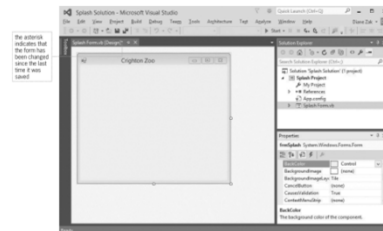


Figure 1-12 Status of the form in the IDE

Saving, Closing, Opening, and Exiting a Solution

Saving a Solution

- An asterisk (*) appears on the designer tab whenever you make a change to the form
- Save the solution by clicking FILE on the menu bar and then clicking Save All

Closing the Current Solution

- Click FILE on the menu bar and then click Close Solution
- To exit Visual Studio 2017, click FILE on the menu bar and then click Exit

Opening an Existing Solution

- Click FILE on the menu bar, and then locate and open the solution from the list of filenames available

Exiting Visual Studio 2017

- Click FILE on the menu bar, and then click EXIT

The Toolbox Window

- Class is a pattern from which an object is created
- Object is an instance with behaviors and attributes
- Toolbox window (or toolbox)
 - Contains tools for designing a user interface
 - Each tool represents a class
 - Controls are instantiated using tools from the toolbox
- Controls are GUI objects that appear on a form

The Toolbox Window

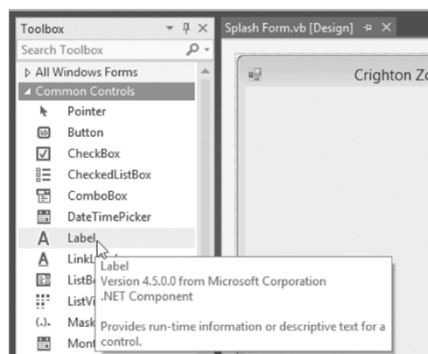


Figure 1-13 Toolbox window showing the purpose of the Label tool

The Label Tool

- Used to create a label control
 - Displays text that cannot be edited at run time
- Name property
 - A unique name for a control
- Text property
 - Specifies the value that appears in a label control
- Location property
 - Specifies the location of a control on a form
 - Includes an X value and a Y value

The Label Tool

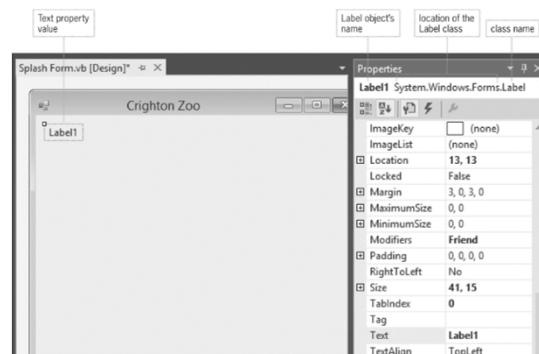


Figure 1-15 Label control added to the form

Setting the Text Property

- Select the control
 - Click text in the Properties list
 - Type in the text you want displayed, e.g. *Come visit our residents!*
 - Press Enter
 - Verify that the AutoSize property is set to True

Changing a Property for Multiple Controls

- A control inherits the font attribute of the form
 - You may want to change the default values
- To simultaneously modify similar controls:
 - Click one control and then press and hold the Ctrl key
 - Click other similar control(s) in the form, or click and drag to group the controls
 - Set the property value in the Properties list



Figure 1-16 Label controls selected on the form

The PictureBox Tool

- Creates a picture box control
 - Displays an image on a form
 - Task box: Lists tasks associated with a picture box
- Use the task box to import an image into a project
- A project's resource file contains imported images

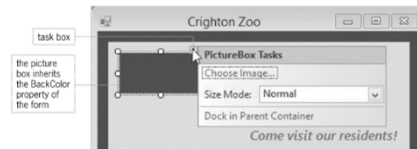


Figure 1-19 Open task list for a picture box

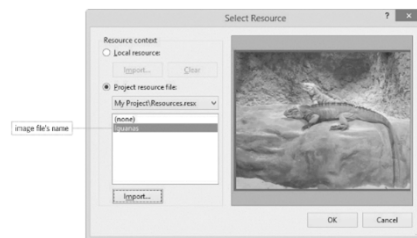


Figure 1-20 Completed Select Resource dialog box

The PictureBox Tool

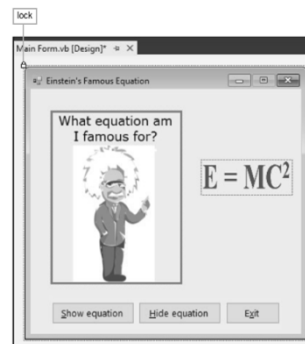
- Using the Format Menu to Align and Size
- The Format menu provides options to manipulate form controls
 - Center in Form: Centers one or more controls
 - Align: Aligns selected controls by left, right, top, or bottom borders
 - Make Same Size: Makes the width and/or height of selected controls uniform
- Select as the first control the one whose size or location you want to match
 - Called the reference control

The Button Tool

- Used to create a button control
 - Used to perform an action when clicked
 - Examples: OK and Cancel buttons
- Common means of exiting a Windows application:
 - Exit option on the FILE menu
 - Exit button

Lock the Controls on the Form

- To lock the controls on the form and then save the solution:
- Right-click the form and then click Lock Controls. A small lock appears in the upperleft corner of the form.



Starting and Ending an Application

- Startup form
 - Displays when an application is first started
- The Project Designer window specifies the startup form or the executable code to be run
 - Right-click My Project in the Solution Explorer window and then click Open
 - Or, click PROJECT on the menu bar and then click *<project name> Properties*
- Application tab
 - Specify startup form

Starting and Ending an Application

- Executable file
 - A file that can be run outside of the IDE
 - Its filename ends with .exe
 - It is automatically created when a Visual Basic application is started
- Two ways to start an application in the IDE:
 - Click DEBUG on the menu bar and then click Start Debugging
 - Press the F5 key

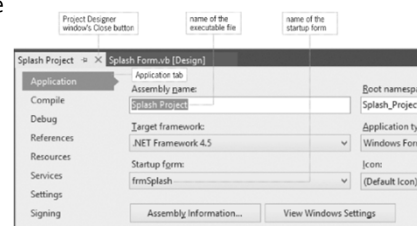


Figure 1-22 Application pane in the Project Designer window

Enter Code and Comments in the Code Editor Window

- After creating your application's interface, you can begin entering the Visual Basic instructions (code) that tell the controls how to respond to the user's actions. Those actions—such as clicking, double-clicking, and scrolling—are called events.
- Event procedure is a set of Visual Basic instructions that are processed only when the event occurs.
- Instructions that are processed (executed) by the computer are also called statements.

Enter Code and Comments in the Code Editor Window

- To open the Code Editor window, right-click the form
- Click View Code on the context menu.
- The Code Editor window opens in the IDE.
- If the line numbers do not appear in your Code Editor window:
 - Click Tools on the menu bar
 - Click Options
 - Expand the Text Editor node
 - Click Basic
 - Select the Line numbers check box
 - Click the OK button.

The Code Editor Window

- Events are user actions such as double-clicking
- Event procedure is a set of program instructions that are processed when an even occurs
- The Code Editor provides the code template to help you follow the rules of the Visual Basic language. The rules of a programming language are called its syntax.
- Code Editor window
 - To open it, right-click the form and then click View Code

The Code Editor Window

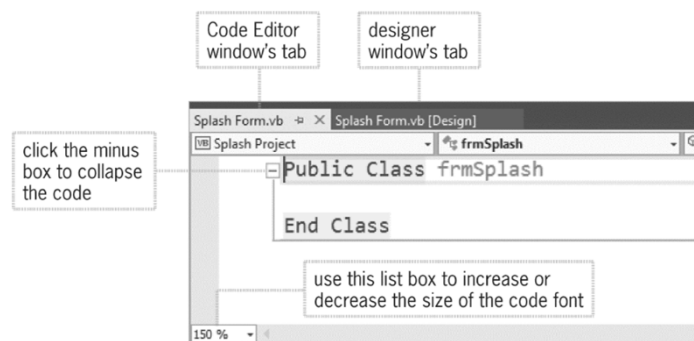


Figure 1-24 Code Editor window opened in the IDE

The Code Editor Window

- OOP or Object-oriented programming; events are clickable
- Syntax are the rules of a programming language
- The first line in the code template is called the procedure header, and the last line is called the procedure footer.
- Procedure header is the first line in the code template for an event procedure
- Procedure footer is the last line in the code template for an event procedure

The Code Editor Window

- A keyword is a word that has a special meaning in a programming language, and it appears in a different color from the rest of the code.

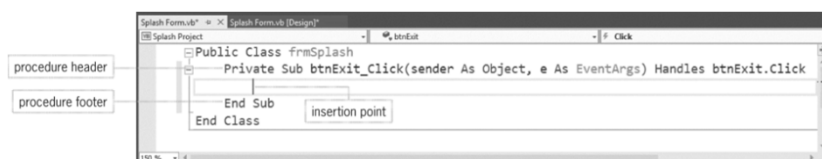


Figure 1-26 btnExit control's Click event procedure

Sub Procedure

- A block of code that performs a specific task
- Keywords in a sub procedure block:
 - `Sub`: starts a sub procedure
 - `Private`: restricts the use of the procedure to the form where it appears
 - `End Sub`: indicated the end of a sub procedure
- Event sub procedure names include the object name and event name
 - Example: `btnExit.Click`

The `Me.Close()` Instruction

- The `Me.Close()` statement tells the computer to close the current form.
- In the instruction, `Me` is a keyword that refers to the current form, and `Close` is one of the methods available in Visual Basic.
- A method is a predefined procedure that you can call (or invoke) when needed.

The Me.Close() Instruction

- Method
 - A predefined procedure
 - Can be called (or invoked) when needed
- Me.Close() method
 - Causes the current form to close
 - If it is the only form in an application, the application terminates
- IntelliSense feature
 - Provides appropriate selections as pop-up lists
 - Type the first few letters of your choice
 - Then use the arrow keys to select an item in the list

The Me.Close() Instruction

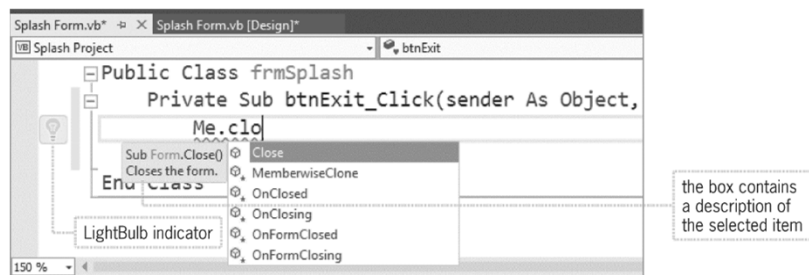


Figure 1-27 List displayed by the IntelliSense feature

Assignment Statements and Comments

- You can set an object's properties during design time, which is when you are building the interface.
- You can also set an object's properties during run time, which occurs while the application is running; you do this by using an assignment statement.
- An assignment statement is one of many different types of Visual Basic instructions.
- The expression can be a keyword, a number, or a string literal, which is defined as zero or more characters enclosed in quotation marks.

Assignment Statements and Comments

- The operator indicates that the *property* is a member of the *object*.
- The equal sign in an assignment statement is called the assignment operator.

Assignment Statement

Syntax
`object.property = expression`

assignment operator

Examples

`picEquation.Visible = False`

assigns the keyword `False` to the `picEquation`'s `Visible` property

`lblDue.Width = 120`

assigns the number 120 to the `lblDue`'s `Width` property

`txtState.Text = "Ohio"`

assigns the string literal "Ohio" to the `txtState`'s `Text` property

Chapter 1 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.

Chapter 1 Summary

- An object-oriented programming language, such as Visual Basic, enables programmers to use objects (e.g., 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 application's GUI are the designer Window, Toolbox Window, solution explorer Window, and properties Window.

Chapter 1 Summary

- 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.