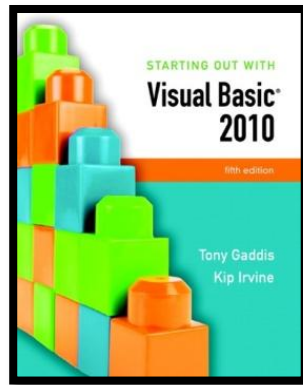STARTING OUT WITH

Visual Basic® 2010

fifth edition

Tony Gaddis

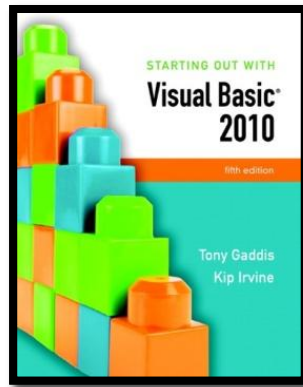**Kip Irvine**

Chapter 2

Creating Applications with Visual Basic

Section 2.1

FOCUS ON PROBLEM SOLVING: BUILDING THE *DIRECTIONS* APPLICATION

In this section you create your first Visual Basic application:
a window that displays a map and road directions to a hotel. In the process you learn how to place controls on a form and manipulate various properties.

Addison Wesley
is an imprint of

Define What the Application is To Do

- Purpose: Display a map to the Highlander Hotel

- Input: None

- Process: Display a form

- Output: Display on the form a graphic image showing a map

Visualize the Application and Design Its User Interface

- Below is a sketch of the application's form

Directions to the Highlander Hotel

Highway 101

Airport

I-89

Exit 125

Hotel

Determine the Controls Needed

Control Type	Control Name	Description
Form	Form1 (Default Name)	A small form that will serve as the window onto which the other controls will be placed
Label	Label1 (Default Name)	Displays the message "Directions to the Highlander Hotel"
PictureBox	PictureBox1 (Default Name)	Displays the graphic image showing the map to the hotel

Define Relevant Property Values for Each Control

- Form
  - Name: Form1
  - Text: "Directions"
- Label
  - Name: Label1
  - Text: "Directions to the Highlander Hotel"
  - TextAlign: MiddleCenter
  - Font: Microsoft sans serif, bold, 18 point
- PictureBox
  - Name: PictureBox1
  - Picture: HotelMap.jpg
  - SizeMode: StretchImage

Create the Forms and Other Controls Using Visual Basic

- Establish the Form and set its Text property
- Add a Label control
  - Position and resize it on the form
  - Set Text, TextAlign, and Font properties
- Add a PictureBox control
  - Position and resize it on the form
  - Set Image property to display HotelMap.jpg
- Run the application
- Close and save the application

Design Mode, Run Mode, and Break Mode

- Visual Basic has three modes in which it operates:
  - Design Mode
    - the mode in which you create the application
    - also known as design time
  - Run Mode
    - executes the application in the Visual Studio environment
    - also known as run time
  - Break Mode
    - momentarily suspends execution of a running application
    - for testing and debugging purposes
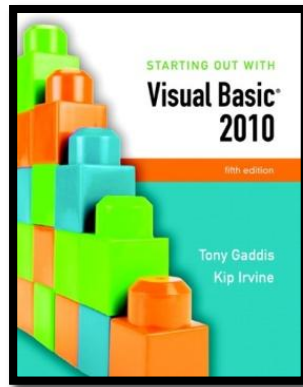
Project Organization On Disk

- User creates a new project in Visual Studio
  - A solution and a folder are created at the same time with the same name as the project
  - The project belongs to the solution
  - Multiple projects can be included in a solution
- The folder stores files related to the project including:
  - A solution file (.sln)
  - A project file (.vbproj)

Opening an Existing Project

- Use Recent Projects list on Start Page
  - Provided it hasn't been moved or deleted
- Use Open Project button on Start Page
  - Then browse using Open Project dialog box
- Use Open Project option on File menu
  - Then browse using Open Project dialog box

Properties Window

- Used to view and modify the property values of a given object

- Two views of the properties are available:
  - Alphabetic (across all properties)
  - Categorized (groups properties by logical use)

Section 2.2

FOCUS ON PROBLEM SOLVING: RESPONDING TO EVENTS

An application responds to events, such as mouse clicks and keyboard input, by executing code known as event handlers. In this section, you write event handlers for the Directions application.

Addison Wesley
is an imprint of

Modify the *Directions* Application

- The Highlander Hotel manager would like you to add the following items to the application:
  - A Label containing the written directions
  - A Button to display the directions
 - A Button to exit the application

Directions to the Highlander Hotel

Airport

Highway 101

I-89

Exit 125

Hotel

Traveling on I-89, take Exit 125 South. The hotel is on the left. Traveling on Highway 101 North, the hotel is on the right, just before the I-89 intersection.

Display Directions

Exit

Controls to be Added

Control Type	Control Name	Description
Label	lblDirections	Displays written directions to the hotel
Button	btnDisplayDirections	When clicked, causes lblDisplayDirections text to appear on the form
Button	btnExit	Stops the application when clicked

Control Properties

- Label
  - Name: lblDirections
  - Text: "Traveling on I-89,…etc"
  - Visible: False
- Button
  - Name: btnDisplayDirections
  - Text: "Display Directions"
- Button
  - Name: btnExit
  - Text: "Exit"

The Code Window

- Double-clicking a control in design mode:
  - opens the code window
  - creates a code template for the control's event handler where you fill in the code for the event

Begin typing code here
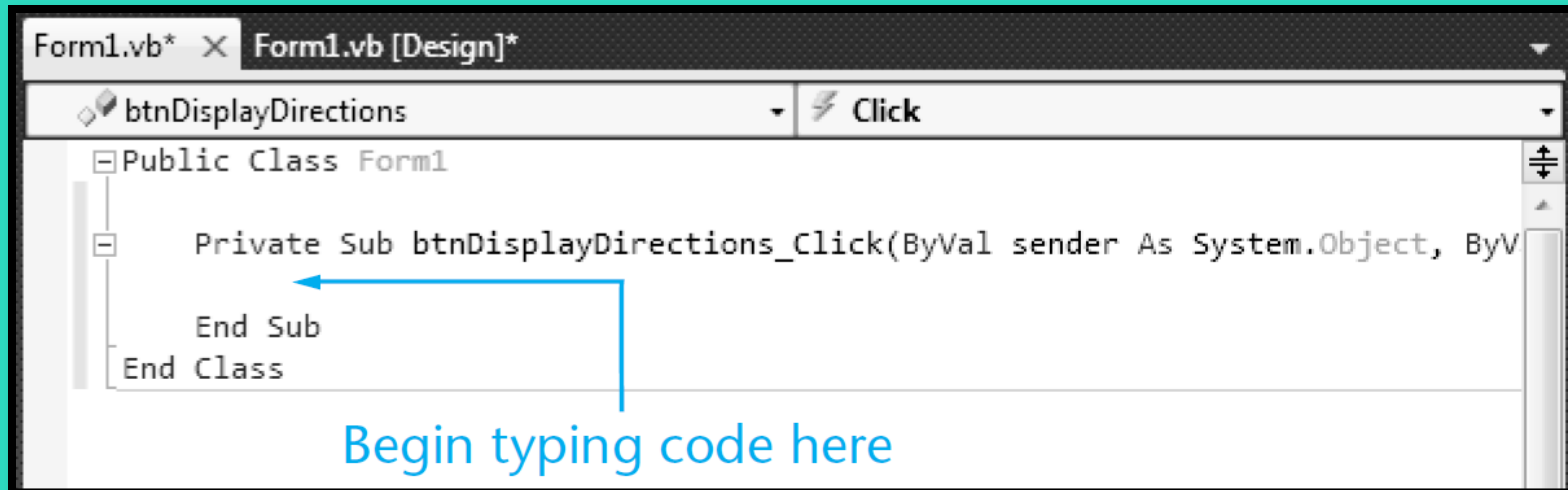
The Click Event Handler for btnDisplayDirections

Marks the beginning of this event procedure

Name of the control that owns the event procedure

Name of the event the procedure responds to

```
Private Sub btnDisplayDirections_Click(…)Handles btnDisplayDirections.Click
    ' Make the directions visible.
    lblDirections.Visible = True
End Sub
```

Event handled by this procedure

Makes the control lblDirections visible:
Assigns the value True to the Visible Property
of the lblDirections control.

Changing a Control's Visible Property in Code

- Specify the control name **`(lblDirections)`**
- Then a dot
- Then the PropertyName **`(Visible)`**
- For example:
  - **`lblDirections.Visible`**
  - refers to the Visible property of the lblDirections control
  - The visible property values may only be true or false

The Assignment Statement

- Specify the item to receive the value
- Then the equal symbol
- Then the value to be assigned
- For example:
  - **lblDirections.Visible = True**
  - Assigns the value True to the Visible property of the lblDirections control
  - Causes the text of the lblDirections control to become visible to the user

Switching Between the Code Window and the Designer Window

- Window Tabs

- Solution Explorer Icons

- Menu Bar

- Keyboard Shortcuts
  - F7 opens the Code Window
  - Shift + F7 opens the Designer Window
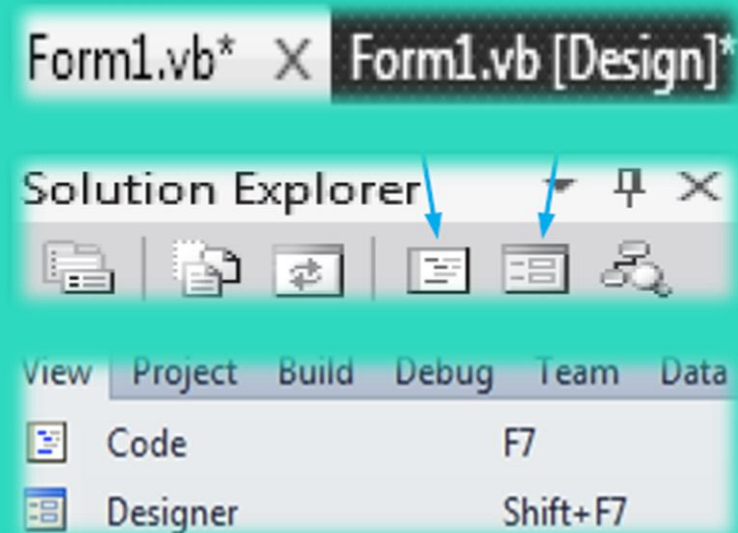
The Click Event Handler for btnExit

Marks the beginning of this event procedure

Name of the control that owns the event procedure

Name of the event the procedure responds to

```
Private Sub btnExit_Click(…) Handles btnExit.Click
    ' Close the form.
    Me.Close()
End Sub
```

Event handled by this procedure

Closes the current form, referred to as *Me*, and ends the program

Use Visual Basic to Update the Application

- Place the label and the buttons on the form

- Enter the code for the two procedures

- Test the application

Changing Text Colors

- Color properties for a control:
  - BackColor: Sets the background (fill) color
  - ForeColor: Sets the foreground (text) color
- In the properties window:
  - click on the down arrow
  - select a color from the list

Custom	Web	System

ActiveBorder
ActiveCaption
ActiveCaptionText
AppWorkspace
ButtonFace
ButtonHighlight
ButtonShadow
Control
ControlDark
ControlDarkDark
ControlLight
ControlLightLight
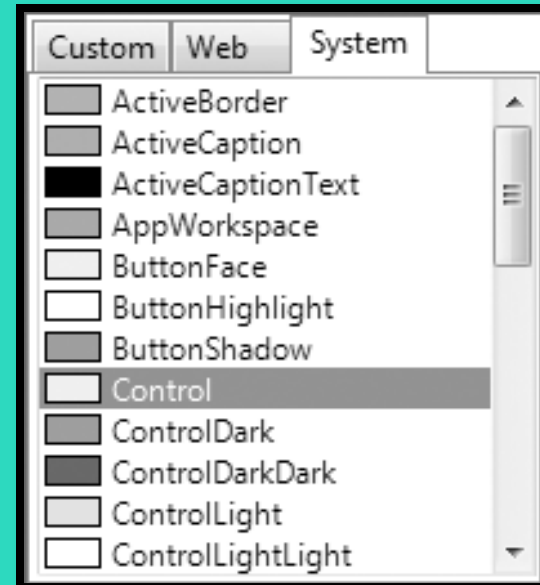
Setting the FormBorderStyle Property

- Border style properties for a form:
  - Sizable: (Default) Has min, max, and close buttons; can be resized by dragging edges
  - Fixed3D: Has a 3D look; min, max, and close buttons; cannot be resized
  - FixedSingle: Has single line border; min, max, and close buttons; cannot be resized

Locking the Controls

- Locking controls prevents them from being moved around during design time
- To Lock Controls:
  - right-click an empty space on the form
  - select *Lock Controls* from the menu
- To Unlock Controls:
  - right-click an empty space on the form
  - select *Lock Controls* from the menu
- Control Locking has a Toggle Effect:
  - Follow the same steps to lock/unlock controls
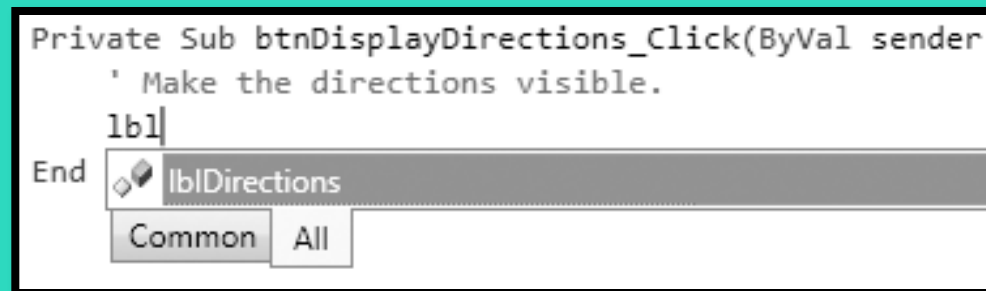
Printing Your Code

- To print a project's code:
    - open the *Code* window
    - click *File* on the menu bar
    - click on the *Print* command
- Using the keyboard shortcut:
    - open the *Code* Window
    - Ctrl + P to print

Using IntelliSense

- IntelliSense is a feature that provides automatic code completion as you type programming statements
- Press the Tab key to use IntelliSense
  - For Example:

```
Private Sub btnDisplayDirections_Click(ByVal sender
    ' Make the directions visible.
    lbl
End   ◇ lblDirections
      Common  All
```

Section 2.3

MODIFYING A CONTROL'S TEXT PROPERTY WITH CODE

Quite often, you will need to change a control's Text property with code. This is done with an assignment statement.

Addison Wesley
is an imprint of

Modifying the Text Property in Code

- Suppose a form is established with a label lblMessage whose Text property is:

    1 Kilometer = ?

- And on a btnFeet button click, we want to change the value of the text property to:

    1 Kilometer = 3,281 feet

Modifying the Text Property in Code

```
Private Sub btnFeet_Click(...) Handles btnFeet.Click
    ' Display the conversion to feet.
    lblMessage.Text = "1 Kilometer = 3,281 feet"
End Sub
```

Assigns the string to the right of the equal sign to the text property of lblMessage

This replaces the previous text property of lblMessage with the new value shown
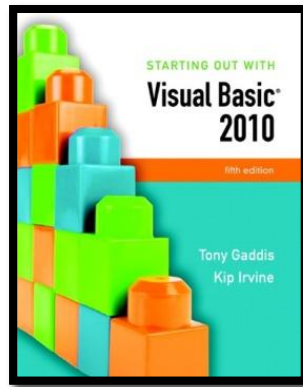
Section 2.4

THE AUTOSIZE, BORDERSTYLE, AND TEXTALIGN PROPERTIES

The Label control's AutoSize property determines whether a label will change size automatically to accommodate the amount of text in its Text property, or remain a fixed size. The BorderStyle property allows you to set a border around a Label control. The TextAlign property determines how the text is aligned within the label.

Addison Wesley
is an imprint of

The AutoSize Property

- AutoSize is a Boolean (either True or False) Property of labels

- False (the default) means the box size will not change, regardless of the amount of text assigned to it

- True means the box will automatically resize itself to fit the amount of text assigned to it

The BorderStyle Property

- BorderStyle determines the look of the box
- None (the default) means no border
- FixedSingle results in a border one pixel wide
- Fixed3D gives the border a recessed 3-dimensional look

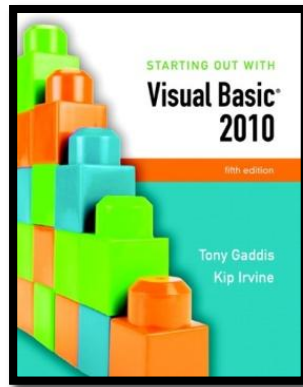| BorderStyle set to None	BorderStyle set to FixedSingle	BorderStyle set to Fixed3D

The TextAlign Property

- The value of TextAlign establishes the alignment (or justification) of the text:
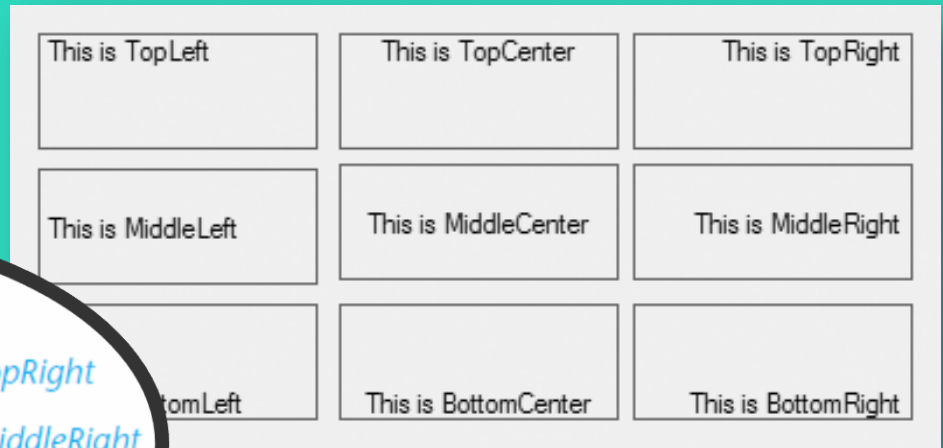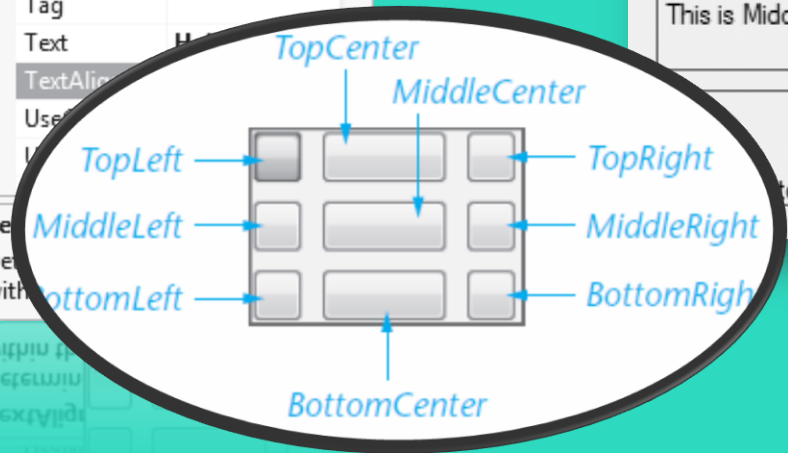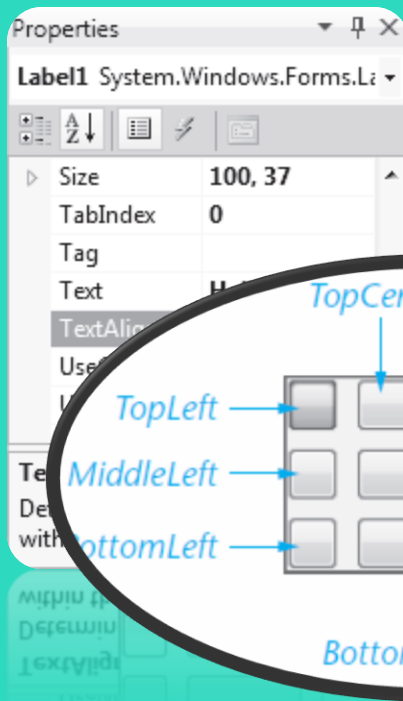
The TextAlign Property

- You can change a label's TextAlign property with code at runtime

- For example:
  - Assume an application uses a Label control named lblReportTitle
  - The following statement aligns the control's text with the middle and center of the control's bounding box

lblReportTitle.TextAlign = ContentAlignment.MiddleCenter

Section 2.5

DISPLAYING MESSAGE BOXES

You can use the MessageBox.Show method to display a message box, which is a dialog box that pops up, showing a message to the user.
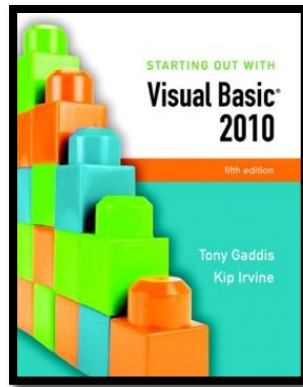
The Message Box

- A message box is
  - a small window
  - sometimes referred to as a dialog box
  - a convenient way to display output to the user
  - displayed until the OK button is clicked

- For example:

MessageBox.Show("Hello World!")

MessageBox   dot   Show   string "Message"
enclosed in parentheses

Section 2.6

CLICKABLE IMAGES

Controls other than buttons can have Click event handlers. In this section, you learn to create PictureBox controls that respond to mouse clicks.

PictureBox Control

- As we saw earlier the Image Property can be set to a graphic image of some sort

- The flag images in Tutorial 2-16 are clickable

- The click event can be handled by code to take whatever action is desired

PictureBox Click Event code

- When PictureBox picUSA is clicked, the lblMessage text property is set to display *United States of America*

```
Private Sub picUSA_Click(…) Handles picUSA.Click
    ' Display United States of America.
    lblMessage.Text = "United States of America"
End Sub
```
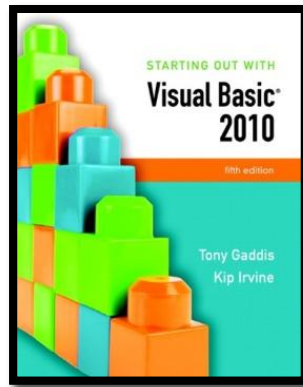
Section 2.7

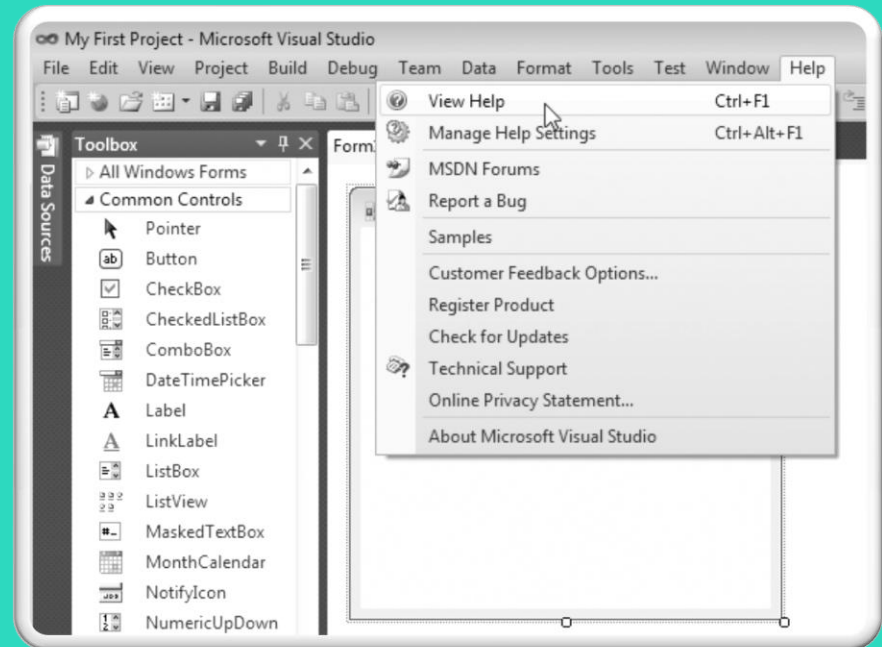# USING VISUAL STUDIO HELP

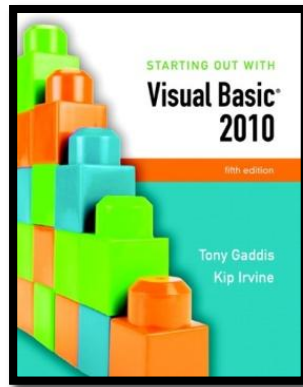Learn to use the Visual Studio Help System

Accessing the Help System

- You access the Visual Studio Documentation and the MSDN library by clicking *Help on the menu bar.*

Context-Sensitive Help (F1 Key)

- Displays information about whatever feature the user is currently focused on
- For example:
  - Click on a Button control
  - Press F1
  - Help explains all about the Button control
  - Click on a Label control
  - Press F1
  - Help explains all about the Label control

Section 2.8

DEBUGGING YOUR APPLICATION

At some point, most applications contain bugs (errors) that prevent the application from operating properly. In this section, you learn fundamental debugging techniques.

Compile Errors

- These are errors in the syntax (form) of your program
- Visual Basic will inform you of these errors as soon as the code is entered
- The area of the error will be underlined with a jagged blue line
- A description of the error will be given in the Error List window
- Display this window by selecting Error List from the View menu option

Runtime Errors

- Some errors occur as your program runs
- These are different from syntax errors which occur as the code is entered by the programmer
- Runtime errors occur when Visual Basic attempts to perform an operation that cannot be executed