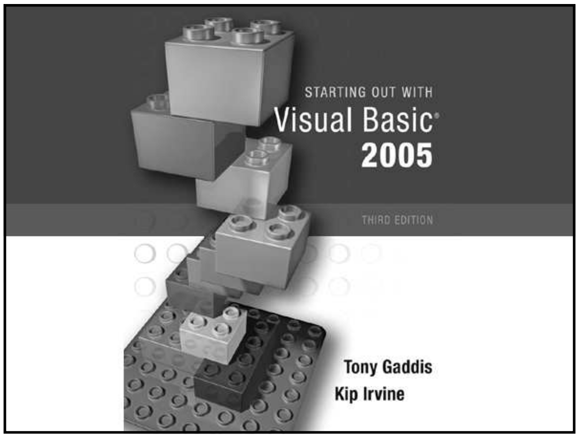
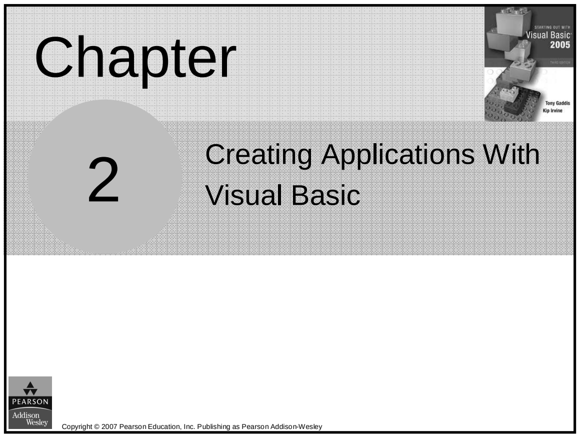


STARTING OUT WITH

Visual Basic® 2005

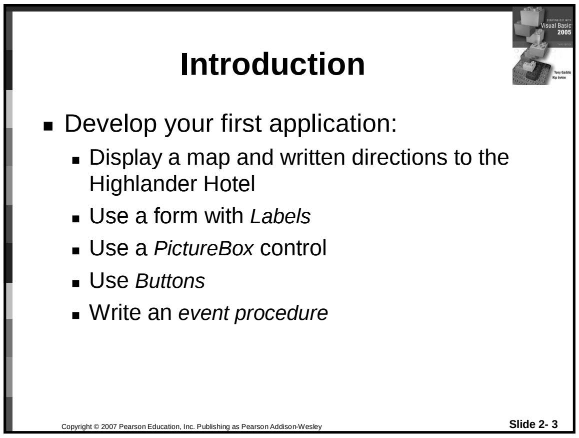
THIRD EDITION

Tony Gaddis
Kip Irvine



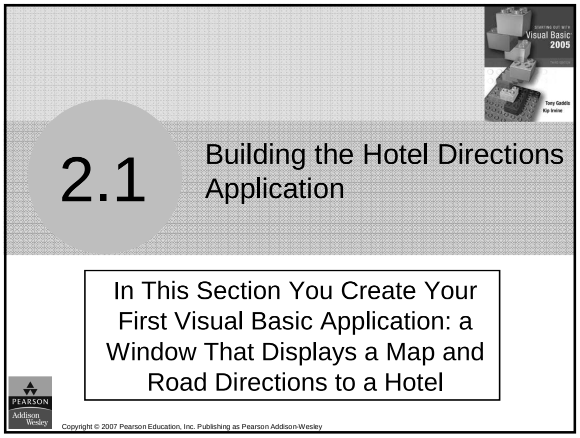
Chapter 2

Creating Applications With Visual Basic



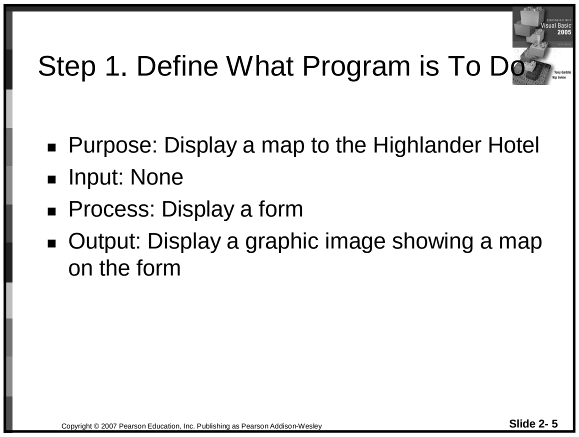
Introduction

- Develop your first application:
  - Display a map and written directions to the Highlander Hotel
  - Use a form with *Labels*
  - Use a *PictureBox* control
  - Use *Buttons*
  - Write an *event procedure*



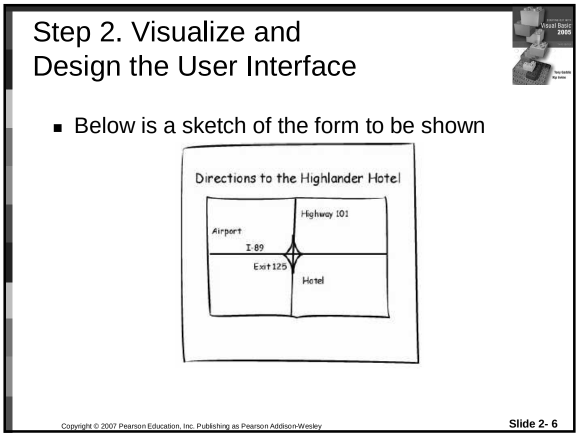
2.1 Building the Hotel Directions Application

In This Section You Create Your First Visual Basic Application: a Window That Displays a Map and Road Directions to a Hotel



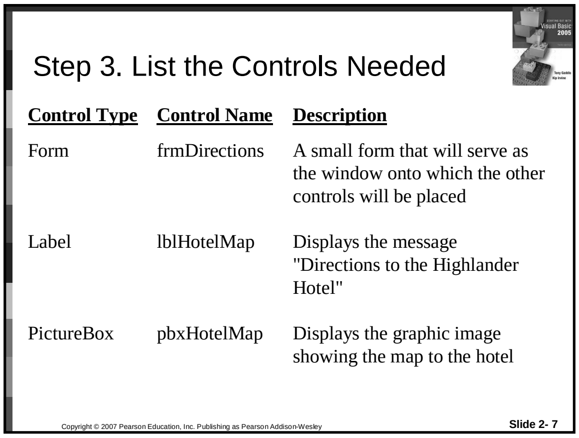
Step 1. Define What Program is To Do

- Purpose: Display a map to the Highlander Hotel
- Input: None
- Process: Display a form
- Output: Display a graphic image showing a map on the form



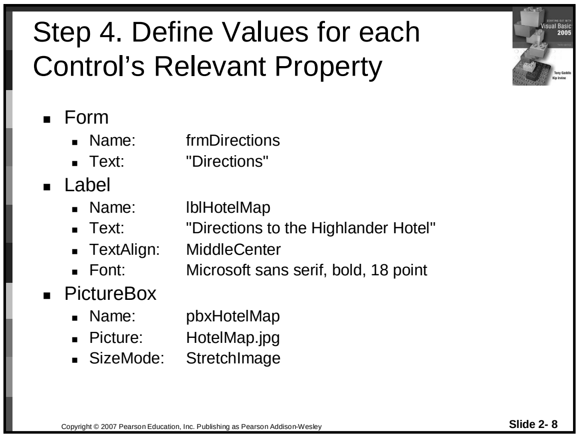
Step 2. Visualize and Design the User Interface

- Below is a sketch of the form to be shown



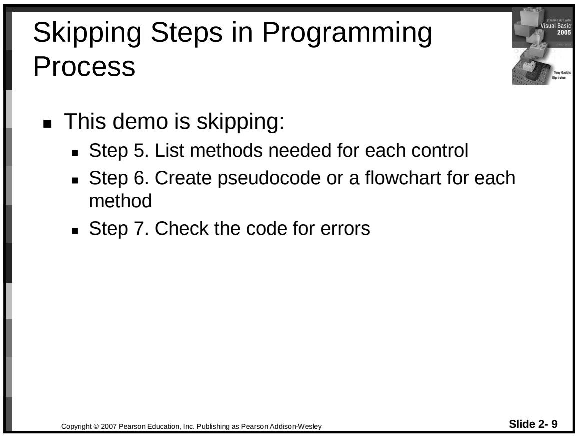
Step 3. List the Controls Needed

Control Type	Control Name	Description
Form	frmDirections	A small form that will serve as the window onto which the other controls will be placed
Label	lblHotelMap	Displays the message "Directions to the Highlander Hotel"
PictureBox	pbxHotelMap	Displays the graphic image showing the map to the hotel



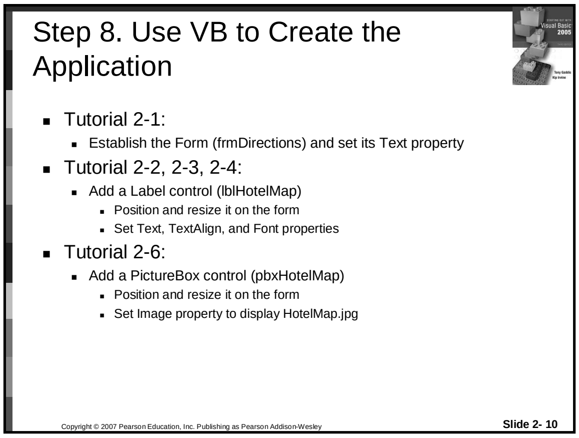
Step 4. Define Values for each Control's Relevant Property

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



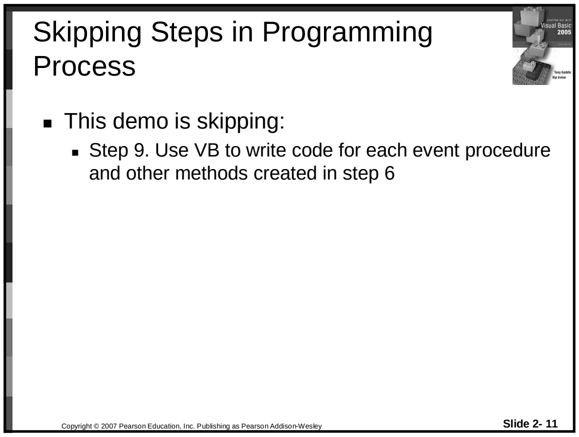
Skipping Steps in Programming Process

- This demo is skipping:
  - Step 5. List methods needed for each control
  - Step 6. Create pseudocode or a flowchart for each method
  - Step 7. Check the code for errors



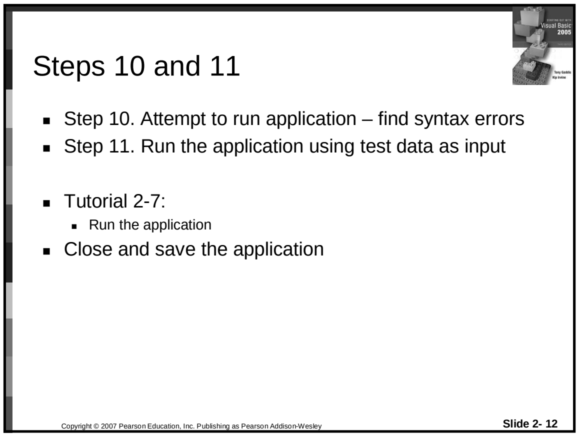
Step 8. Use VB to Create the Application

- Tutorial 2-1:
  - Establish the Form (frmDirections) and set its Text property
- Tutorial 2-2, 2-3, 2-4:
  - Add a Label control (lblHotelMap)
    - Position and resize it on the form
    - Set Text, TextAlign, and Font properties
- Tutorial 2-6:
  - Add a PictureBox control (pbxHotelMap)
    - Position and resize it on the form
    - Set Image property to display HotelMap.jpg



Skipping Steps in Programming Process

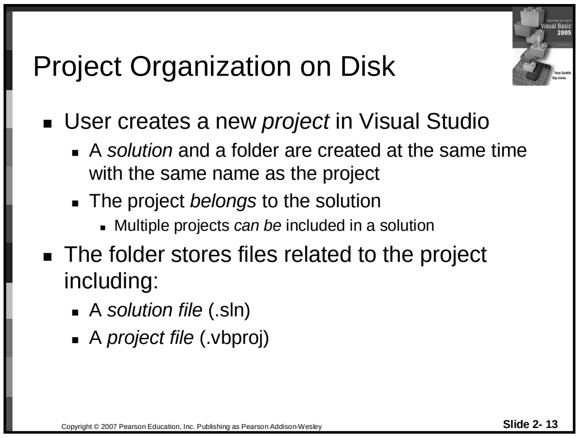
- This demo is skipping:
  - Step 9. Use VB to write code for each event procedure and other methods created in step 6



Steps 10 and 11

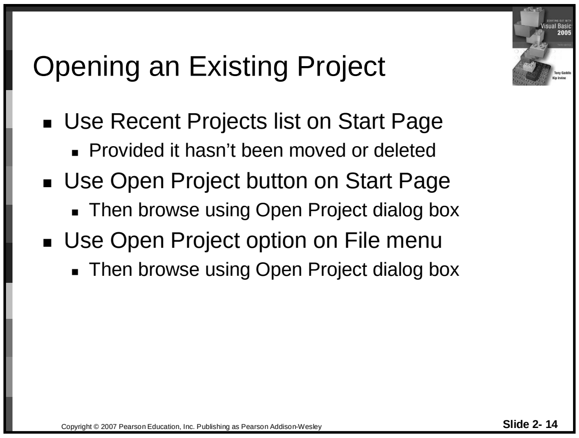
- Step 10. Attempt to run application – find syntax errors
- Step 11. Run the application using test data as input

- Tutorial 2-7:
  - Run the application
- Close and save the application



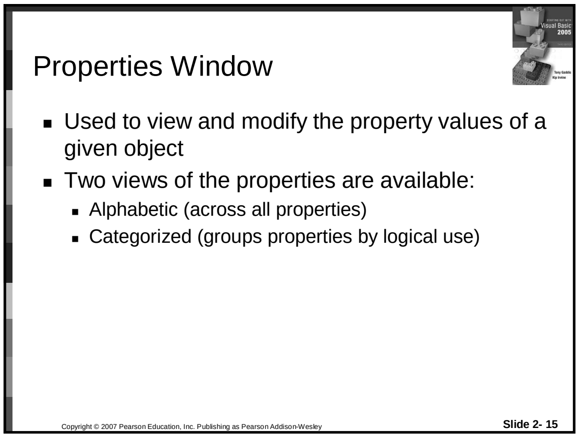
Project Organization on Disk

- 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)



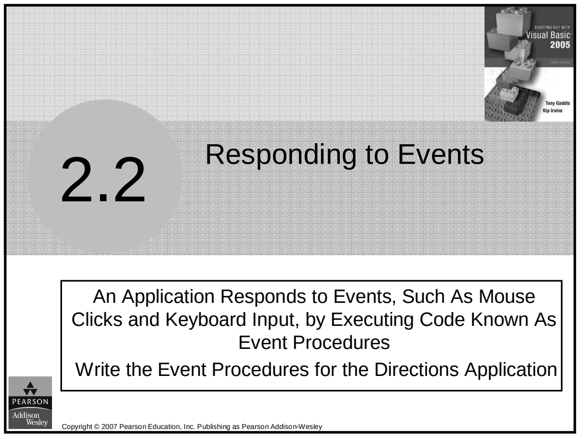
Opening an Existing Project

- 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

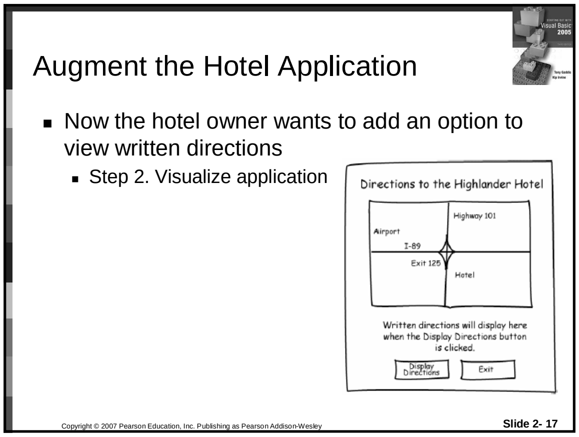
- Used to view and modify the property values of a given object
- Two views of the properties are available:
  - Alphabetic (across all properties)
  - Categorized (groups properties by logical use)



2.2 Responding to Events

An Application Responds to Events, Such As Mouse Clicks and Keyboard Input, by Executing Code Known As Event Procedures

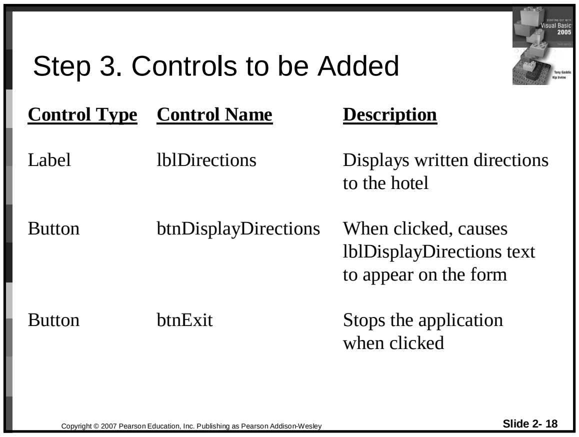
Write the Event Procedures for the Directions Application



Augment the Hotel Application

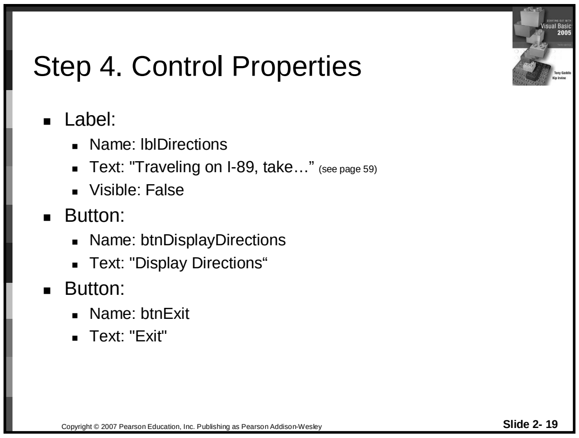
- Now the hotel owner wants to add an option to view written directions
  - Step 2. Visualize application

Directions to the Highlander Hotel
Highway 101
Airport
I-89
Exit 125
Hotel
Written directions will display here when the Display Directions button is clicked.
Display Directions
Exit



Step 3. Controls to be Added

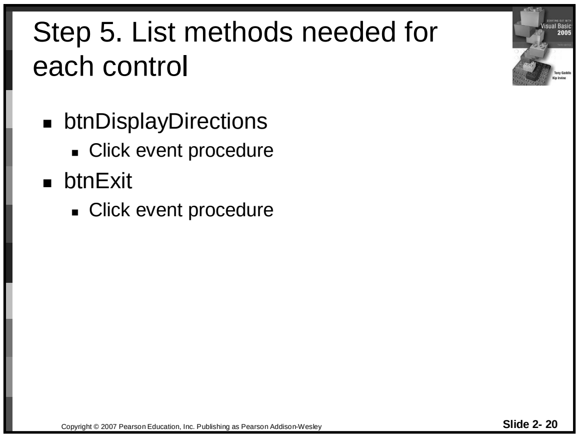
Control Type	Control Name	Description
Label	lblDirections	Displays written directions to the hotel
Button	btnDisplayDirections	When clicked, causes lblDisplayDirections text to appear on the form
Button	btnExit	Stops the application when clicked



Step 4. Control Properties

- Label:
  - Name: lblDirections
  - Text: "Traveling on I-89, take…" (see page 59)
  - Visible: False
- Button:
  - Name: btnDisplayDirections
  - Text: "Display Directions“
- Button:
  - Name: btnExit
  - Text: "Exit"

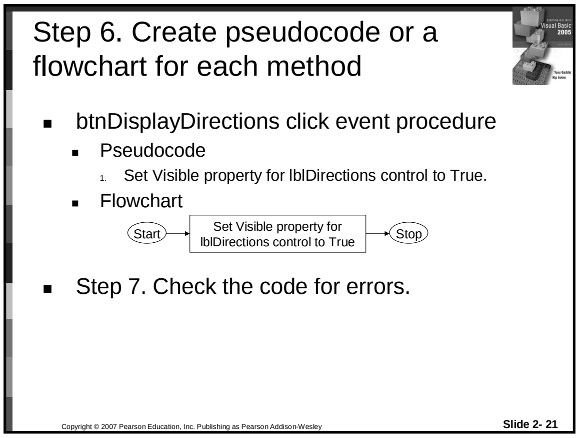
Slide 2- 19



Step 5. List methods needed for each control

- btnDisplayDirections
  - Click event procedure
- btnExit
  - Click event procedure

Slide 2- 20



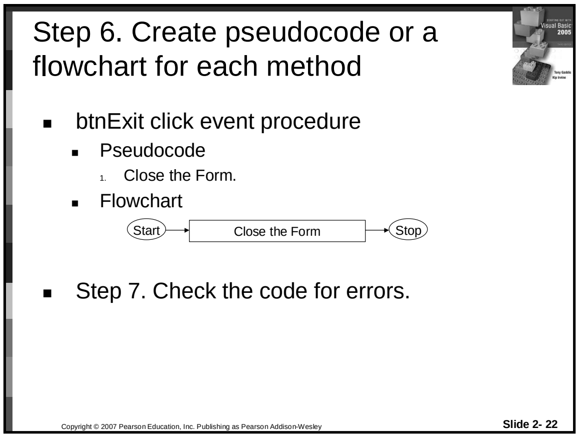
Step 6. Create pseudocode or a flowchart for each method

- btnDisplayDirections click event procedure
  - Pseudocode
    1. Set Visible property for lblDirections control to True.
  - Flowchart

Start → Set Visible property for lblDirections control to True → Stop

- Step 7. Check the code for errors.

Slide 2- 21



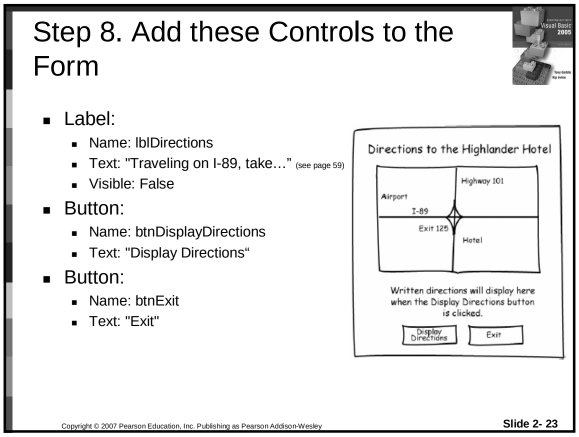
Step 6. Create pseudocode or a flowchart for each method

- btnExit click event procedure
  - Pseudocode
    1. Close the Form.
  - Flowchart

Start → Close the Form → Stop

- Step 7. Check the code for errors.

Slide 2- 22

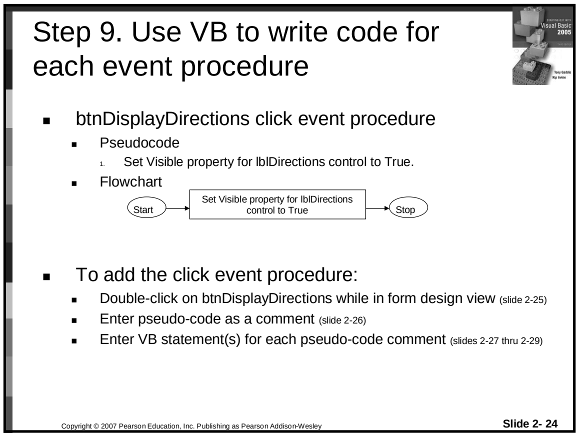


Step 8. Add these Controls to the Form

- Label:
  - Name: lblDirections
  - Text: "Traveling on I-89, take…" (see page 59)
  - Visible: False
- Button:
  - Name: btnDisplayDirections
  - Text: "Display Directions“
- Button:
  - Name: btnExit
  - Text: "Exit"

Directions to the Highlander Hotel
Highway 101
Airport
I-89
Exit 125
Hotel
Written directions will display here when the Display Directions button is clicked.
Display Directions
Exit

Slide 2- 23



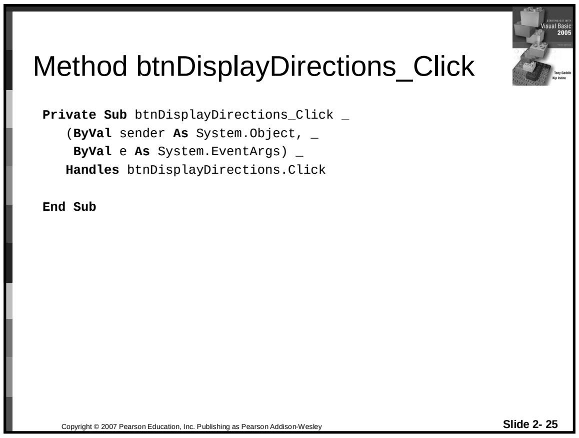
Step 9. Use VB to write code for each event procedure

- btnDisplayDirections click event procedure
  - Pseudocode
    1. Set Visible property for lblDirections control to True.
  - Flowchart

Start → Set Visible property for lblDirections control to True → Stop

- To add the click event procedure:
  - Double-click on btnDisplayDirections while in form design view (slide 2-25)
  - Enter pseudo-code as a comment (slide 2-26)
  - Enter VB statement(s) for each pseudo-code comment (slides 2-27 thru 2-29)

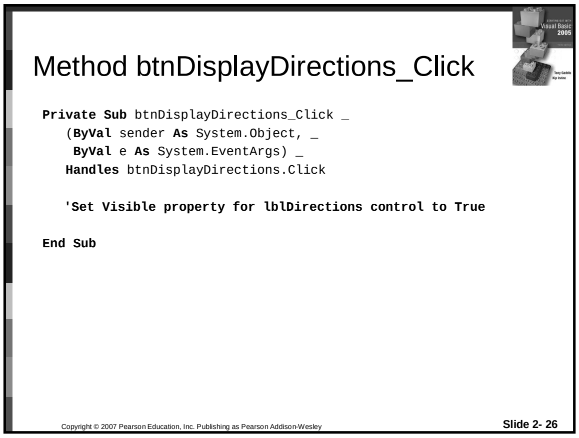
Slide 2- 24



Method btnDisplayDirections_Click

```
Private Sub btnDisplayDirections_Click _
   (ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
   Handles btnDisplayDirections.Click

End Sub
```

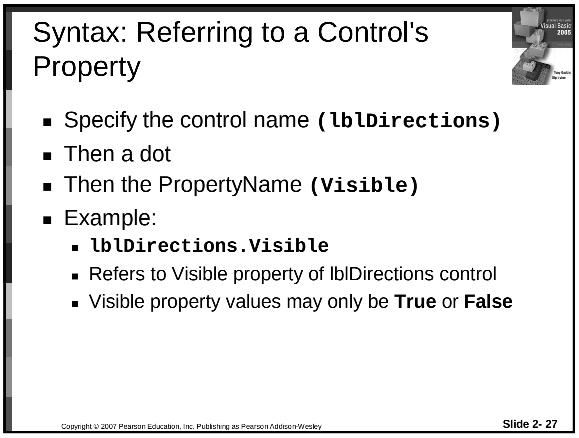
Slide 2- 25



Method btnDisplayDirections_Click

```
Private Sub btnDisplayDirections_Click _
   (ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
   Handles btnDisplayDirections.Click

   'Set Visible property for lblDirections control to True

End Sub
```

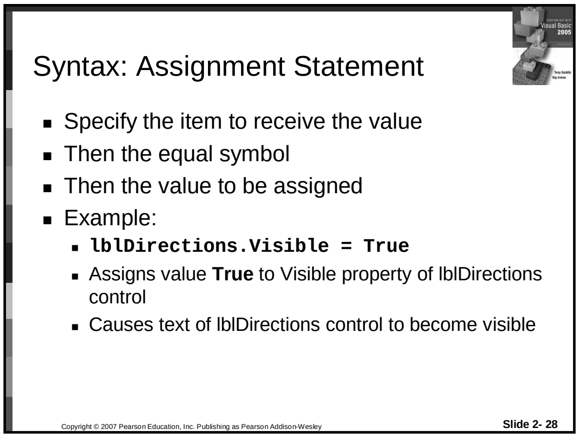
Slide 2- 26



Syntax: Referring to a Control's Property

- Specify the control name **(lblDirections)**
- Then a dot
- Then the PropertyName **(Visible)**
- Example:
  - **lblDirections.Visible**
  - Refers to Visible property of lblDirections control
  - Visible property values may only be **True** or **False**

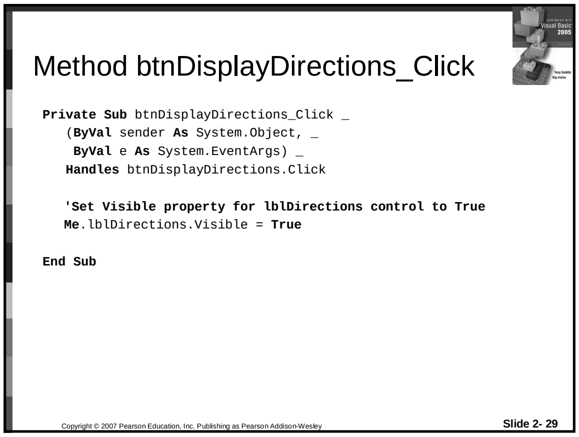
Slide 2- 27



Syntax: Assignment Statement

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

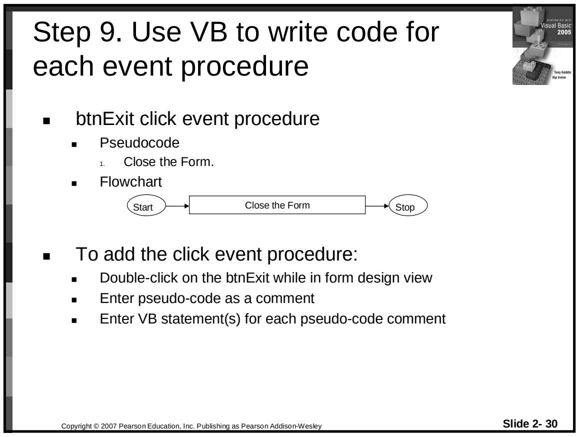
Slide 2- 28



Method btnDisplayDirections_Click

```
Private Sub btnDisplayDirections_Click _
   (ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
   Handles btnDisplayDirections.Click

   'Set Visible property for lblDirections control to True
   Me.lblDirections.Visible = True

End Sub
```

Slide 2- 29



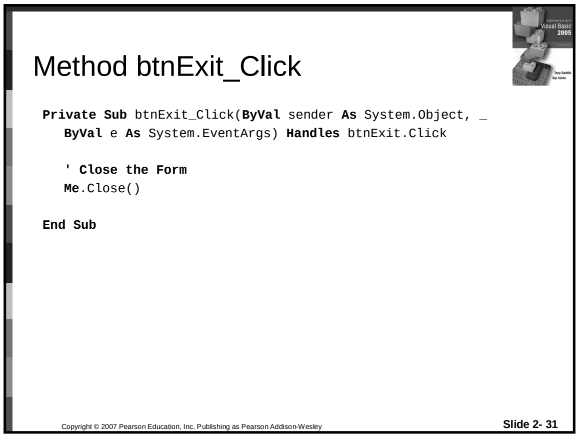
Step 9. Use VB to write code for each event procedure

- btnExit click event procedure
  - Pseudocode
    1. Close the Form.
  - Flowchart

Start
Close the Form
Stop

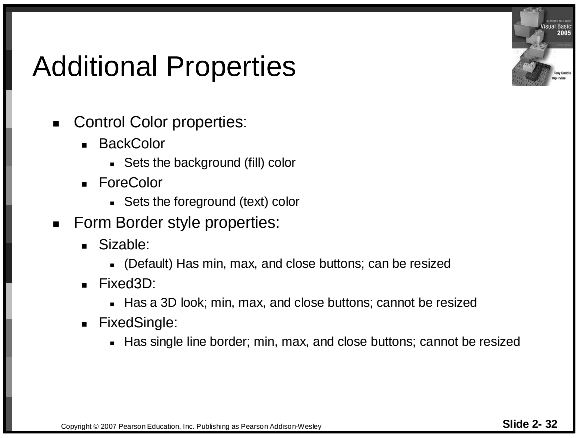
- To add the click event procedure:
  - Double-click on the btnExit while in form design view
  - Enter pseudo-code as a comment
  - Enter VB statement(s) for each pseudo-code comment

Slide 2- 30



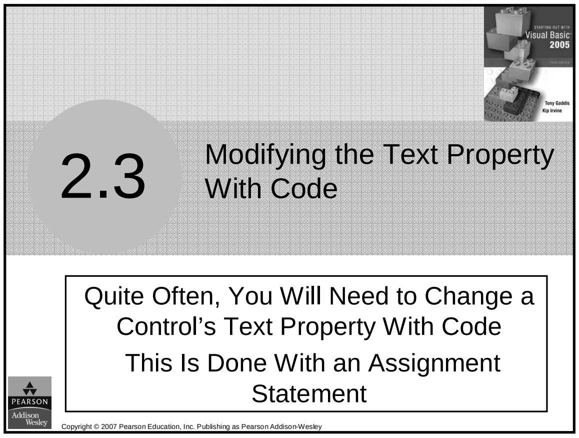
Method btnExit_Click

```
Private Sub btnExit_Click(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles btnExit.Click

   ' Close the Form
   Me.Close()

End Sub
```



Additional Properties

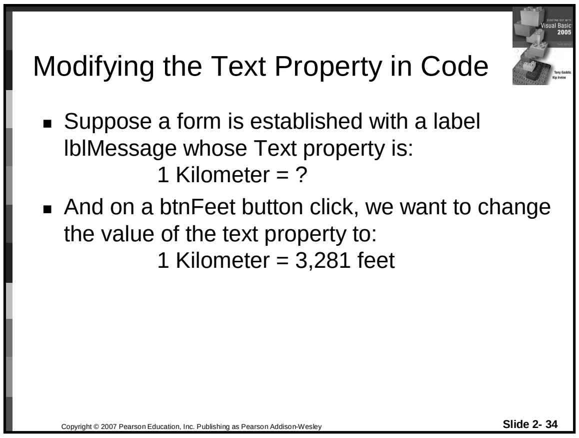
- Control Color properties:
  - BackColor
    - Sets the background (fill) color
  - ForeColor
    - Sets the foreground (text) color
- Form Border style properties:
  - Sizable:
    - (Default) Has min, max, and close buttons; can be resized
  - Fixed3D:
    - Has a 3D look; min, max, and close buttons; cannot be resized
  - FixedSingle:
    - Has single line border; min, max, and close buttons; cannot be resized



2.3 Modifying the Text Property With Code

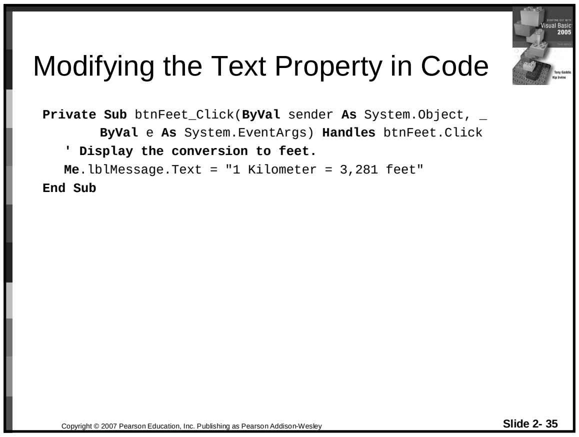
Quite Often, You Will Need to Change a Control's Text Property With Code

This Is Done With an Assignment Statement



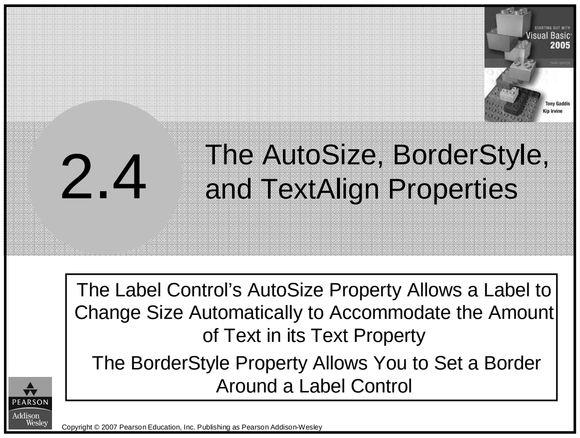
Modifying the Text Property in Code

- 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

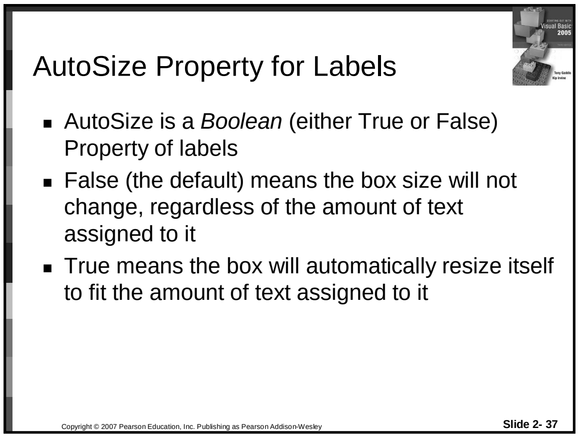
```
Private Sub btnFeet_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnFeet.Click
   ' Display the conversion to feet.
   Me.lblMessage.Text = "1 Kilometer = 3,281 feet"
End Sub
```



2.4 The AutoSize, BorderStyle, and TextAlign Properties

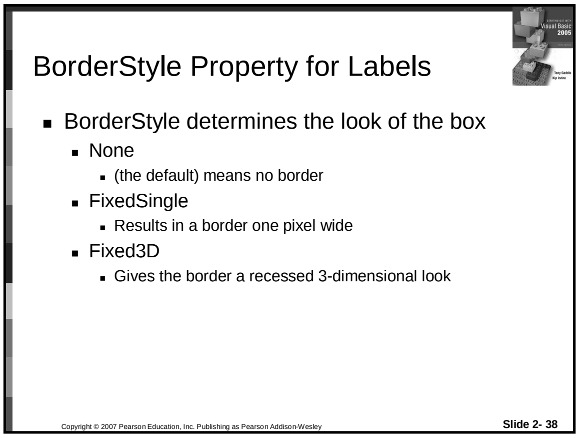
The Label Control's AutoSize Property Allows a Label to Change Size Automatically to Accommodate the Amount of Text in its Text Property

The BorderStyle Property Allows You to Set a Border Around a Label Control



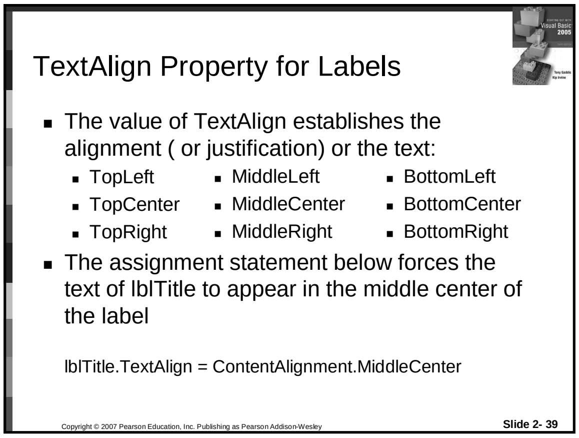
AutoSize Property for Labels

- 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



BorderStyle Property for Labels

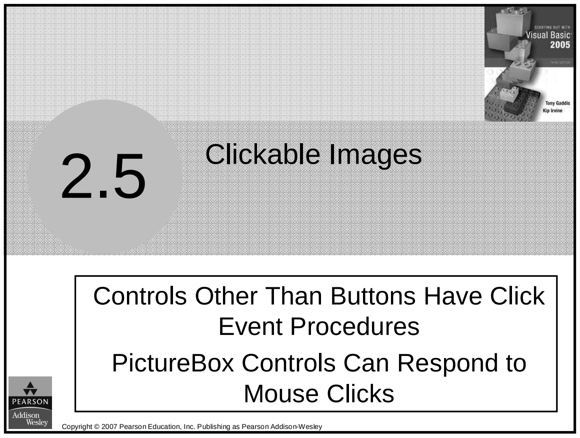
- BorderStyle determines the look of the box
  - None
    - (the default) means no border
  - FixedSingle
    - Results in a border one pixel wide
  - Fixed3D
    - Gives the border a recessed 3-dimensional look



TextAlign Property for Labels

- The value of TextAlign establishes the alignment (or justification) or the text:
  - TopLeft
  - TopCenter
  - TopRight
  - MiddleLeft
  - MiddleCenter
  - MiddleRight
  - BottomLeft
  - BottomCenter
  - BottomRight
- The assignment statement below forces the text of lblTitle to appear in the middle center of the label

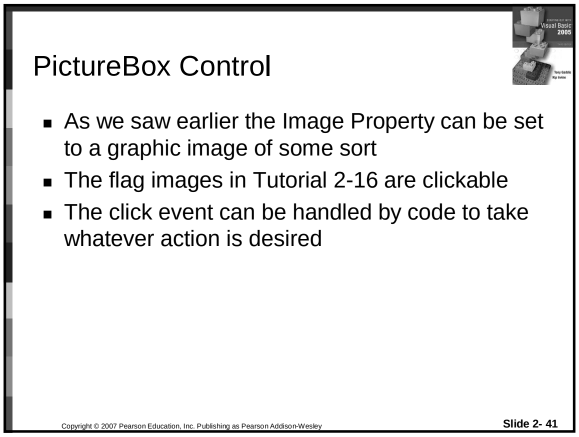
lblTitle.TextAlign = ContentAlignment.MiddleCenter



2.5 Clickable Images

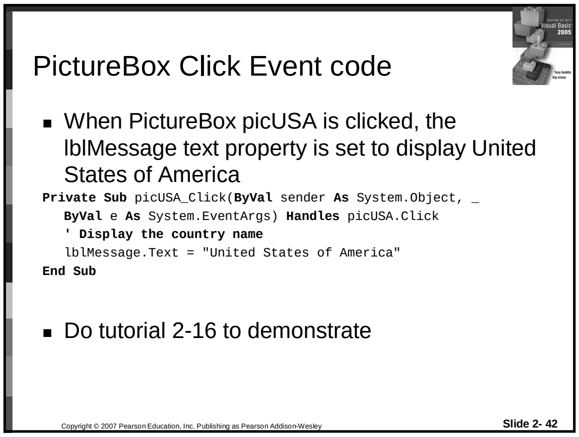
Controls Other Than Buttons Have Click Event Procedures

PictureBox Controls Can 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

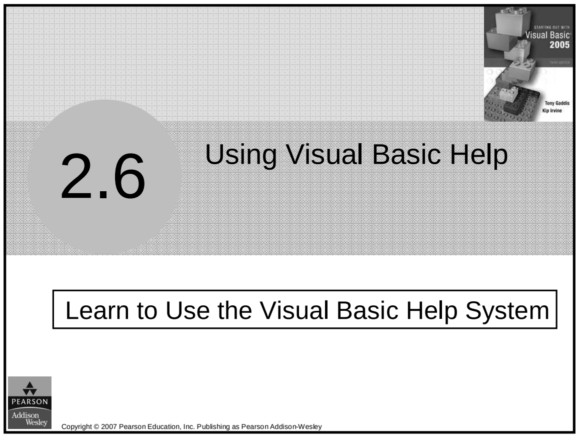


PictureBox Click Event code

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

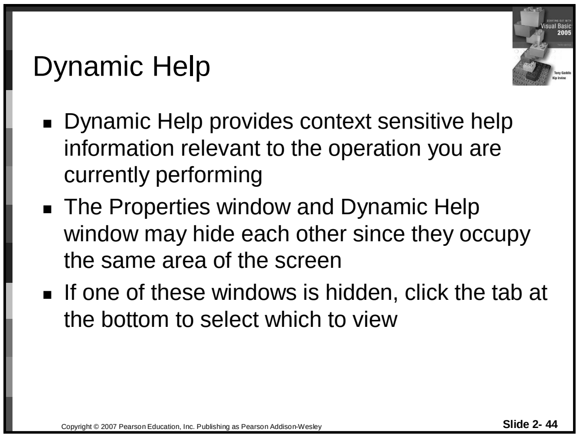
```
Private Sub picUSA_Click(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles picUSA.Click
   ' Display the country name
   lblMessage.Text = "United States of America"
End Sub
```

- Do tutorial 2-16 to demonstrate



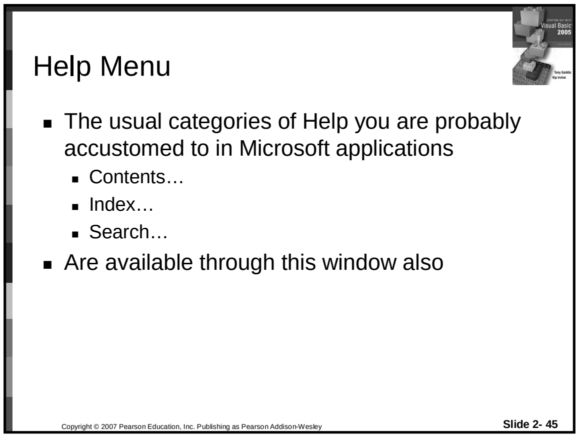
2.6 Using Visual Basic Help

Learn to Use the Visual Basic Help System



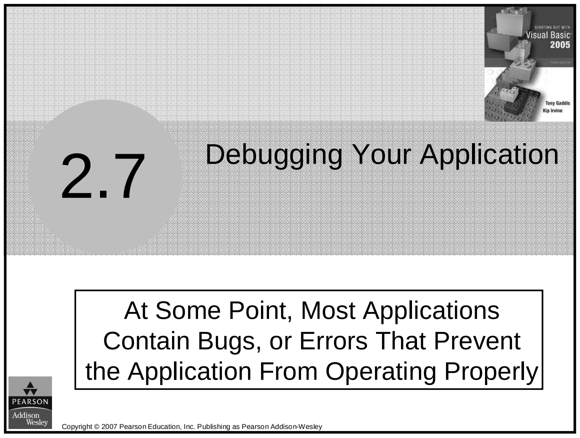
Dynamic Help

- Dynamic Help provides context sensitive help information relevant to the operation you are currently performing
- The Properties window and Dynamic Help window may hide each other since they occupy the same area of the screen
- If one of these windows is hidden, click the tab at the bottom to select which to view



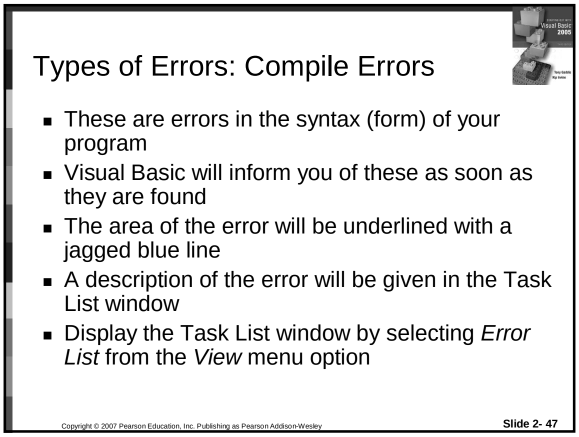
Help Menu

- The usual categories of Help you are probably accustomed to in Microsoft applications
  - Contents…
  - Index…
  - Search…
- Are available through this window also



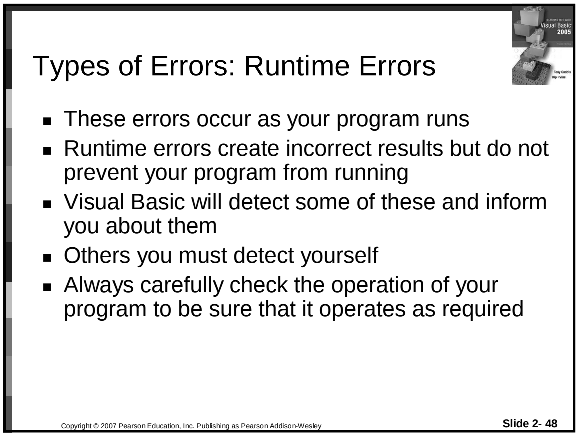
2.7 Debugging Your Application

At Some Point, Most Applications Contain Bugs, or Errors That Prevent the Application From Operating Properly



Types of Errors: Compile Errors

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



Types of Errors: Runtime Errors

- These errors occur as your program runs
- Runtime errors create incorrect results but do not prevent your program from running
- Visual Basic will detect some of these and inform you about them
- Others you must detect yourself
- Always carefully check the operation of your program to be sure that it operates as required