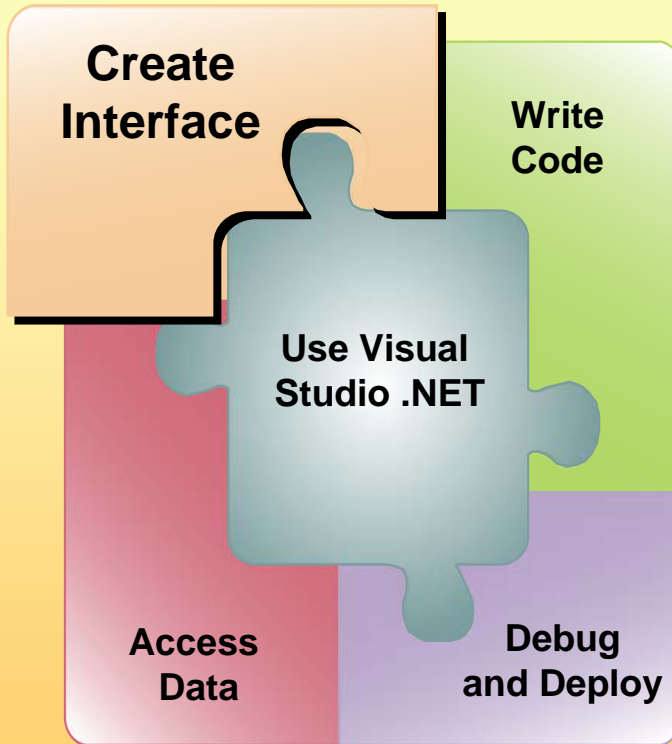


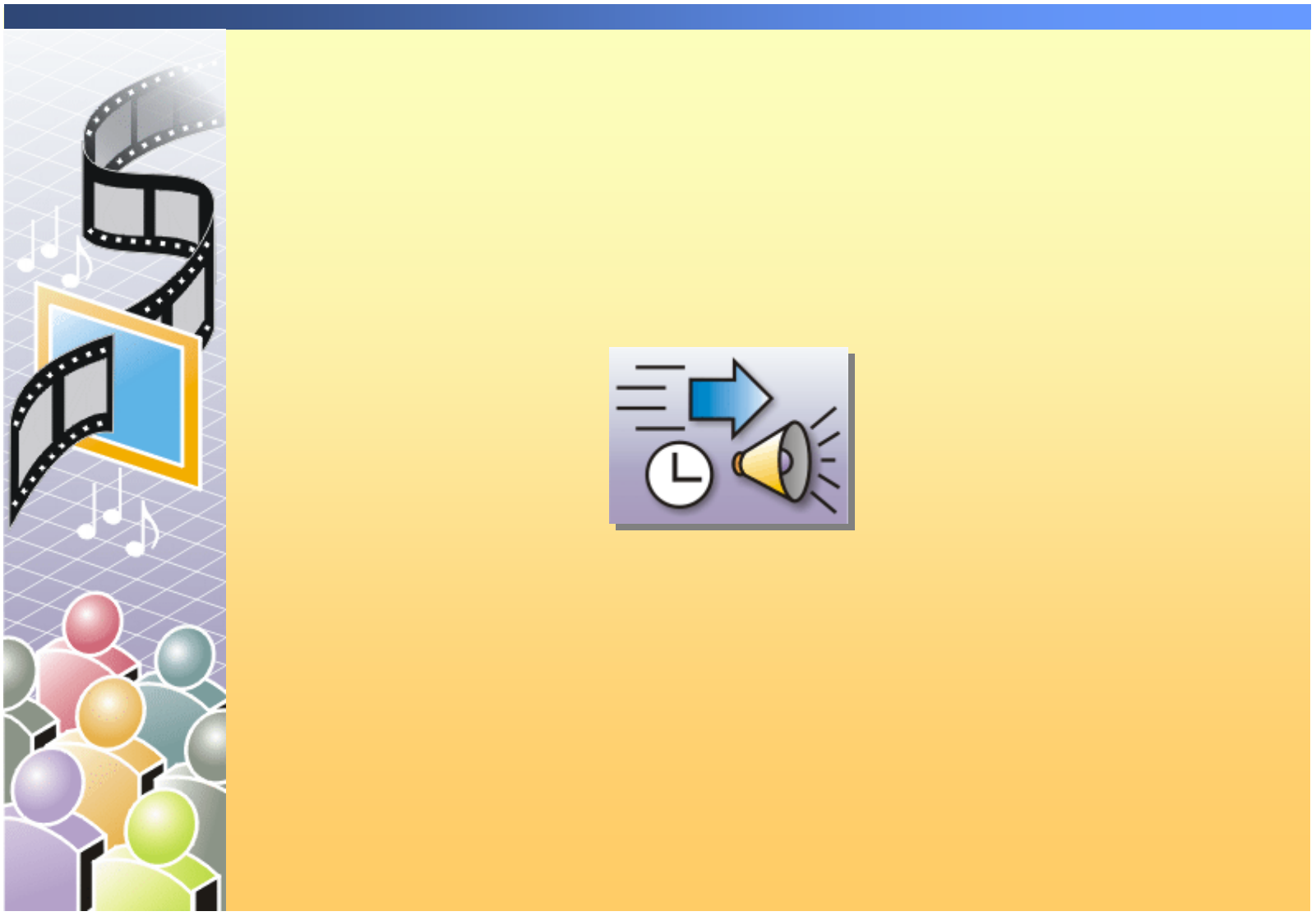
**Module 2:
Working with Forms
and Controls**

Overview



- Understanding Programming Concepts
- Working with Windows Forms
- Working with Controls
- Styling Your Code

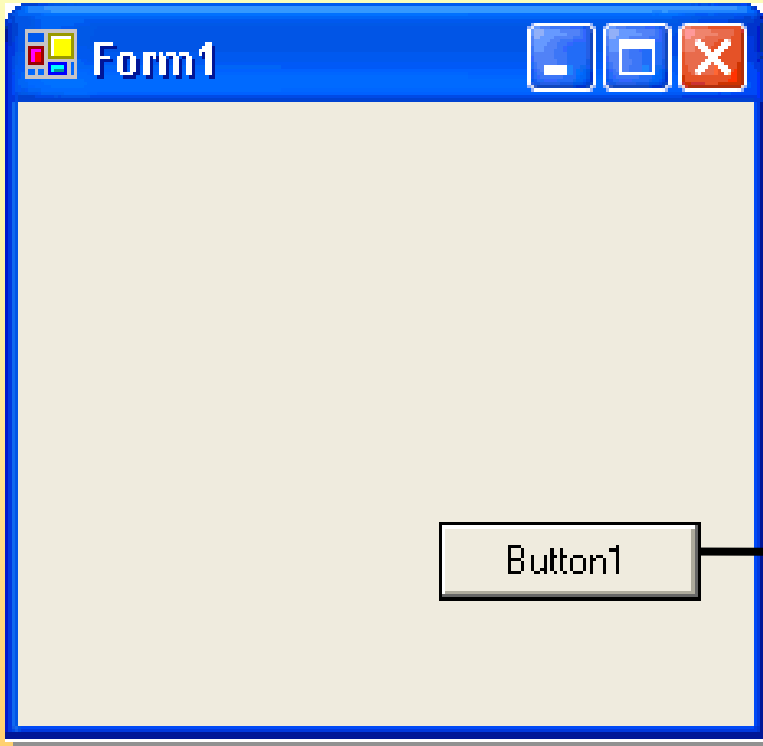
Multimedia: What Are Objects, Properties, Methods, and Events?



Lesson: Understanding Programming Concepts

- Event-driven programming
- Classes
- Events
- Methods
- Objects
- Properties

What Is Event-Driven Programming?



```
Sub Button1_Click (...)  
    'Insert code for event  
End Sub
```

Classes: Blueprints for Objects

Class

A symbolic representation of an object.

Analogy: A blueprint.



Object

An instance of a class.

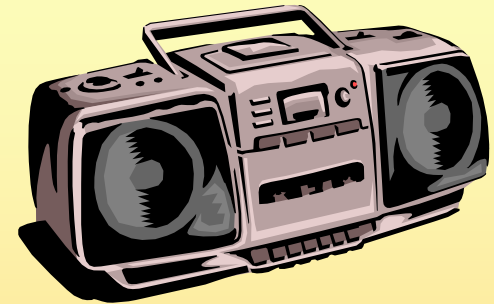
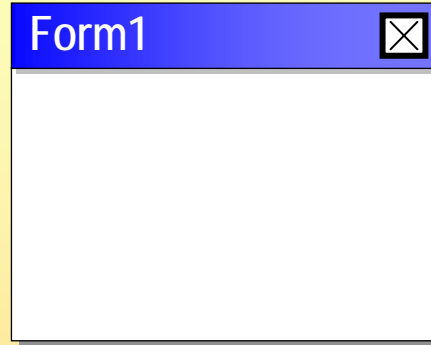
Analogy: A building based on the blueprint.



Example

Each form in a Visual Basic project is an object.
Each form is an instance of the Form class.

What Are Properties, Methods, and Events?



Properties	Size Text	Volume Bass
Methods	Close Hide	Tune
Events	Click	Low Battery

Practice: Properties, Methods, and Events



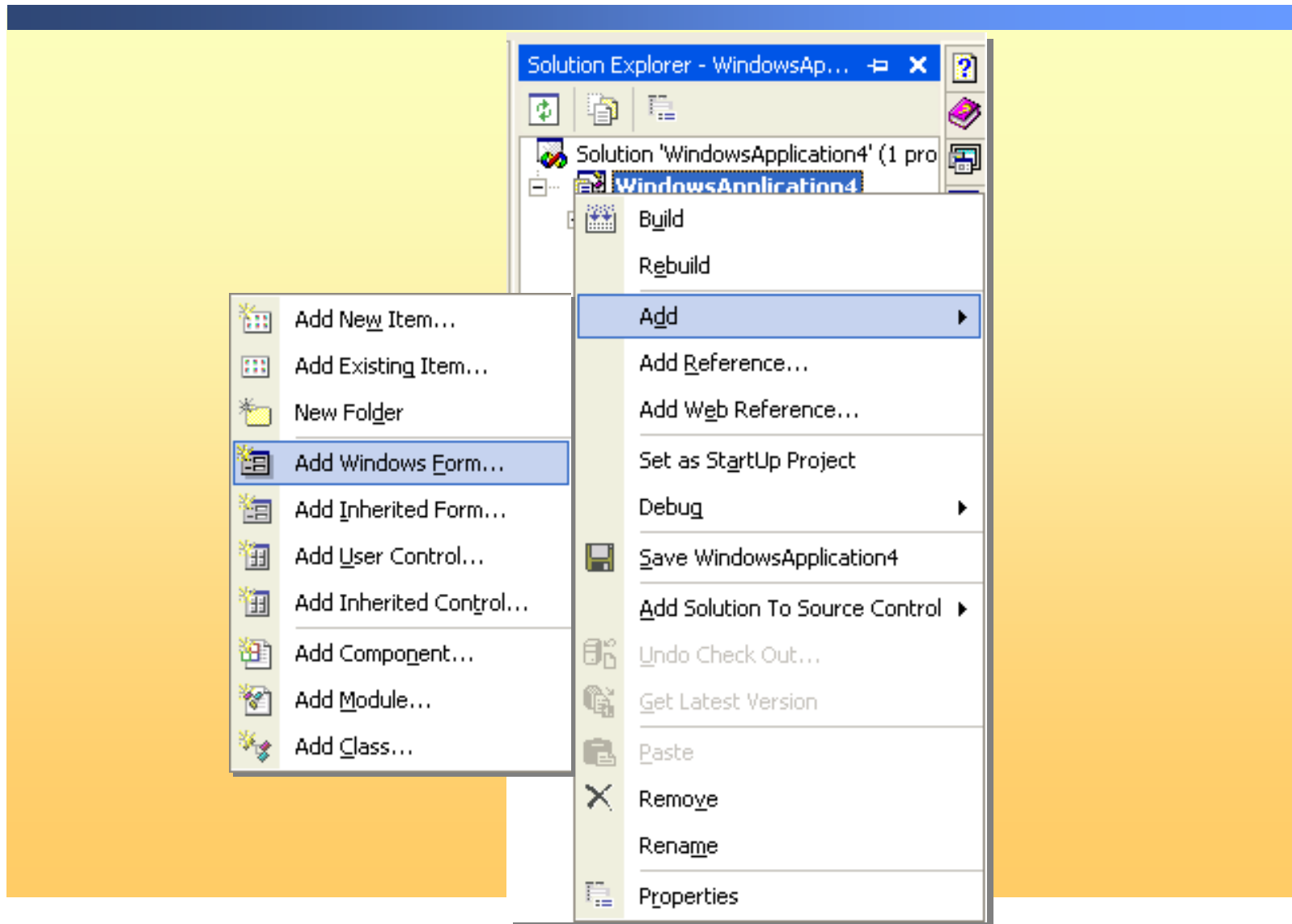
For each of the following objects, suggest associated properties, methods, and events:

- Door
- Car
- *Example: Telephone*
 - *Properties: Width, Color, Volume*
 - *Methods: Ring, VolumeChange*
 - *Events: VolumeChanged, DialCompleted*

Lesson: Working with Windows Forms

- How to Create a Form
- How to Set Form Properties
- How to Call Methods
- How to Handle Form Events
- Modal and Modeless Forms
- How to Manage Multiple Forms

How to Create a Form



How to Set Form Properties

The screenshot shows the Visual Studio Properties window for a form named **Form1**. The window title is "Properties" and the object is identified as **Form1** of type `System.Windows.Forms.Form`. The window contains a toolbar with icons for categorized, alphabetic, and description views. Below the toolbar is a list of properties, with **AcceptButton** selected. A description pane at the bottom provides details for the **AcceptButton** property.

Object Name

Categorized Button

Alphabetic Button

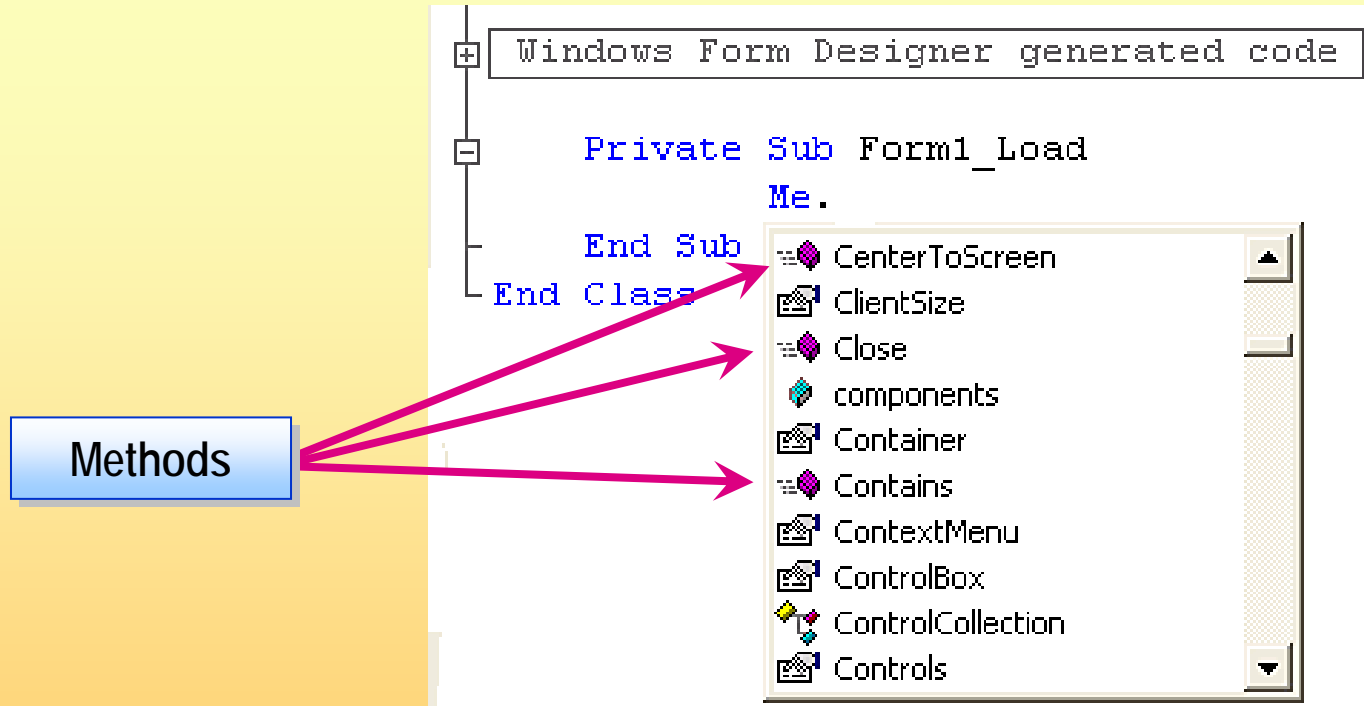
Description Pane

(DataBindings)	
(DynamicProperti	
(Name)	Form1
AcceptButton	(none)
AccessibleDescrip	
AccessibleName	
AccessibleRole	Default
AllowDrop	False

AcceptButton
The accept button of the form. If this is set, the button is 'clicked' whenever the user presses the 'ENTER' key.

- If you change the Name property of Form1, you must also change the startup object for your project

How to Call Methods



The screenshot shows a code editor window titled "Windows Form Designer generated code". The code contains a private sub procedure:

```
Private Sub Form1_Load  
    Me.  
End Sub  
End Class
```

A blue box labeled "Methods" has three red arrows pointing to the dropdown menu that appears after the dot in the code. The dropdown menu lists the following methods:

- CenterToScreen
- ClientSize
- Close
- components
- Container
- Contains
- ContextMenu
- ControlBox
- ControlCollection
- Controls

```
Sub Form1_Click  
    Me.CenterToScreen( )  
End Sub
```

How to Handle Form Events

The image shows a screenshot of the Visual Studio IDE. The main window displays the code for a class named `Form1`, which inherits from `System.Windows.Forms.Form`. The code is as follows:

```
Public Class Form1
    Inherits System.Windows.Forms.Form
    Windows Form Designer generated code.
End Class
```

On the right side of the IDE, the **(Declarations)** window is open, showing a list of events for the `Form1` class. The `Click` event is currently selected. The list of events includes:

- Activated
- BackColorChanged
- BackgroundImageChanged
- BindingContextChanged
- CausesValidationChanged
- ChangeUICues
- Click
- Closed
- Closing
- ContextMenuChanged
- CursorChanged

Two callouts are present:

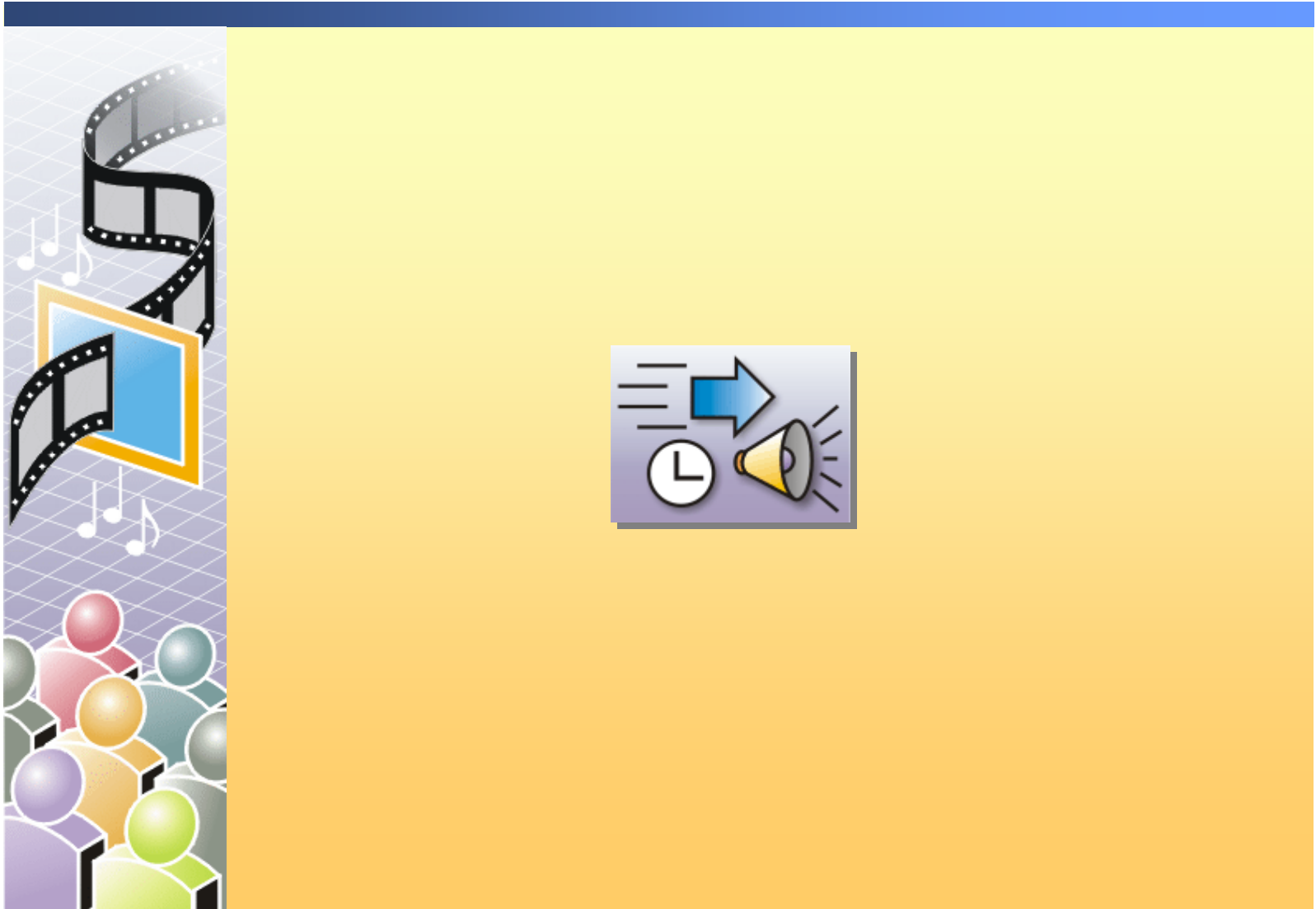
- A blue box labeled **Class Name List Box** with a pink arrow pointing to the `Form1` class name in the code.
- A blue box labeled **Events** with a pink arrow pointing to the `Click` event in the declarations list.

Practice: Writing Code for Form Events

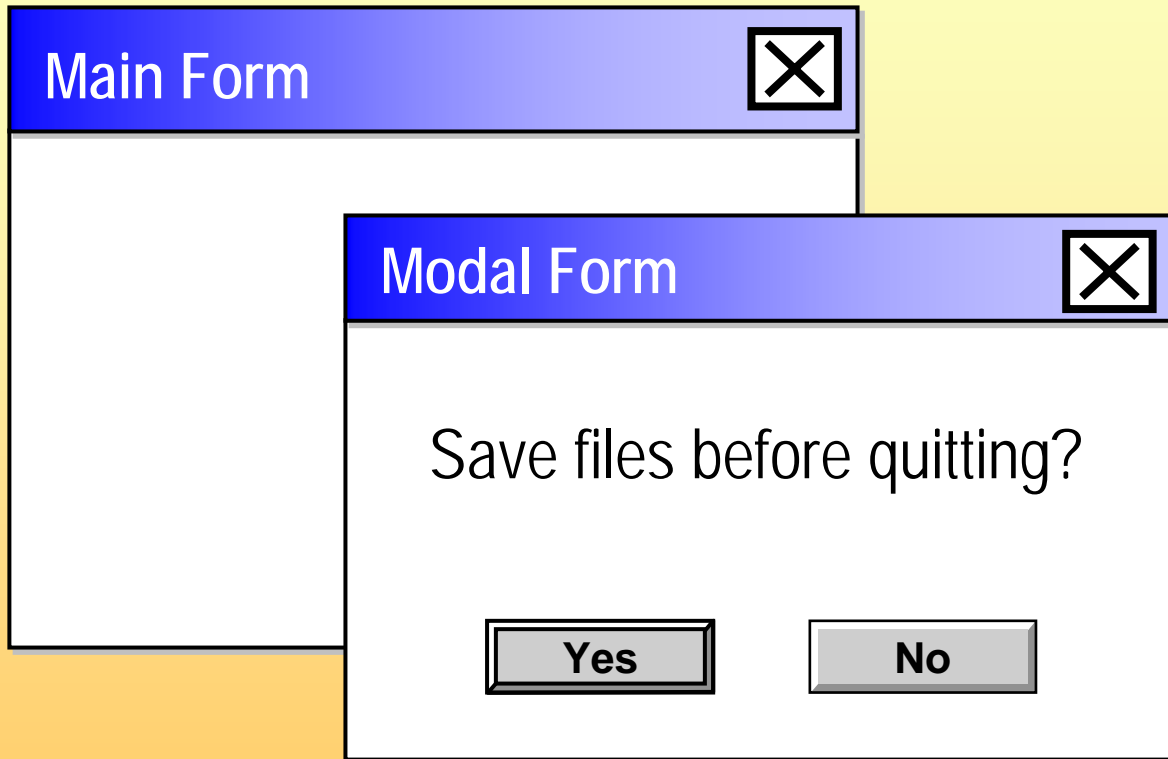


- 1** Open a new Windows application in Visual Basic .NET
- 2** Open the Code Editor for the form
- 3** Open the Form1_Click event handler
- 4** Add code to the event handler
- 5** Run the application and test your code

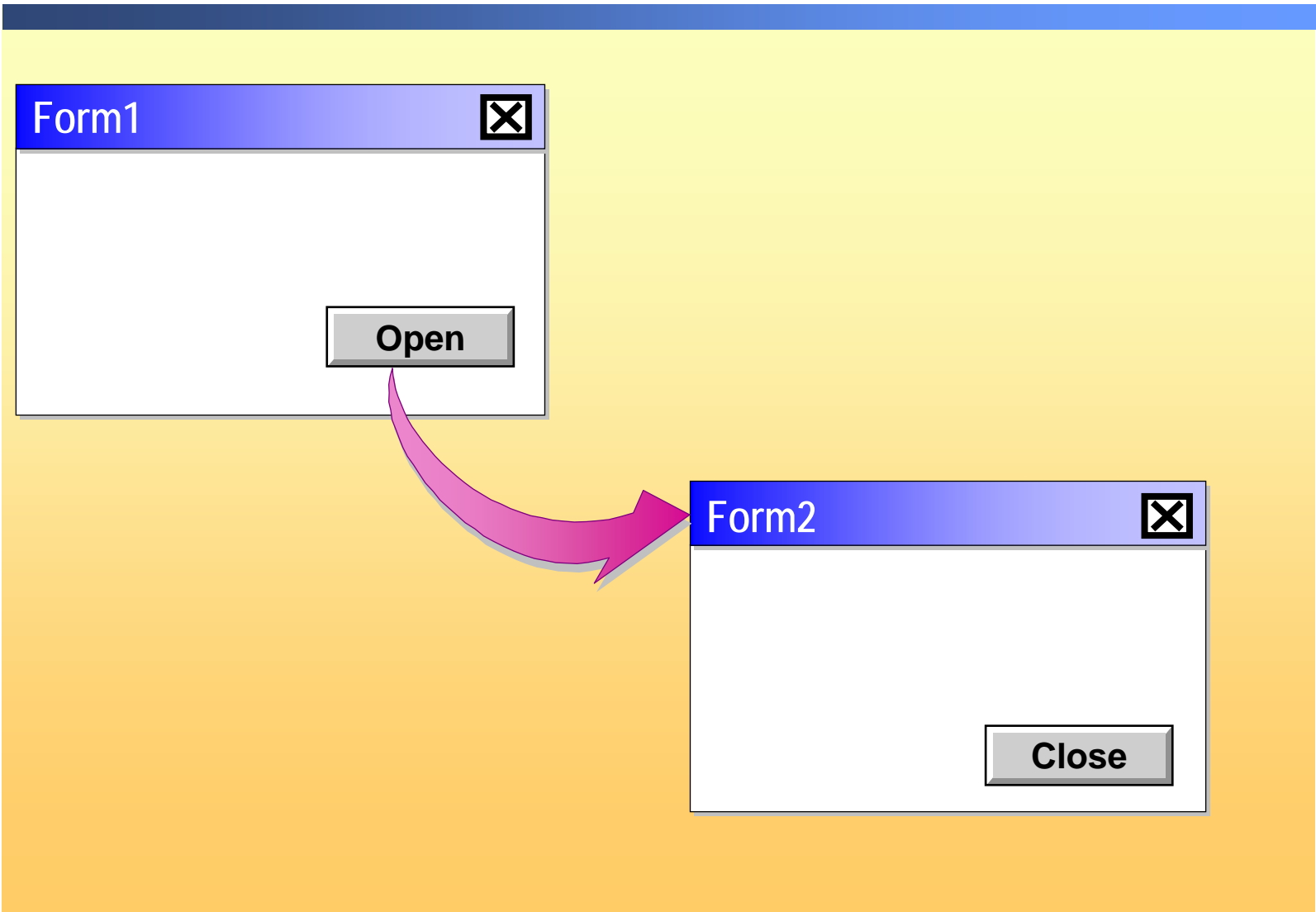
Multimedia: Managing Forms



Modal and Modeless Forms



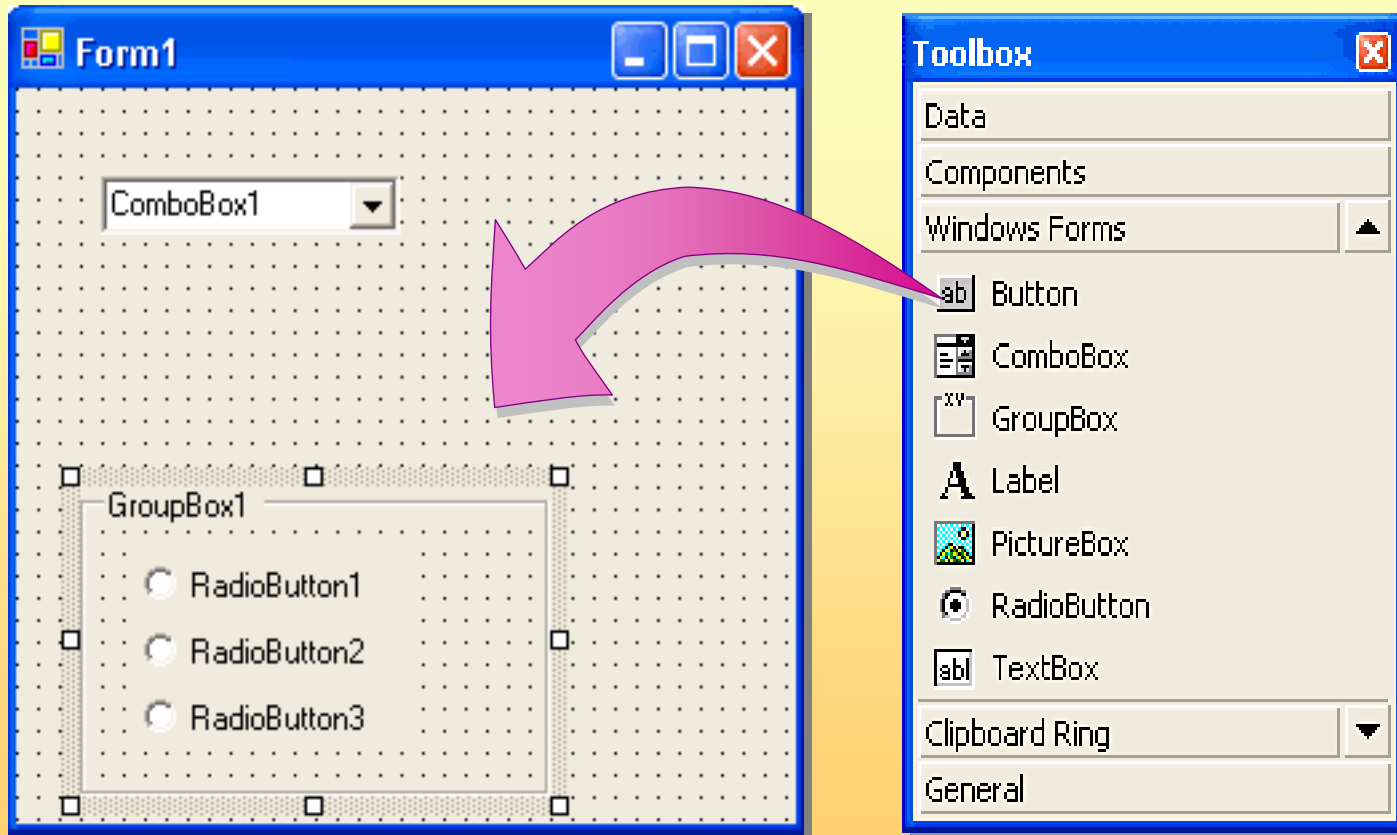
How to Manage Multiple Forms



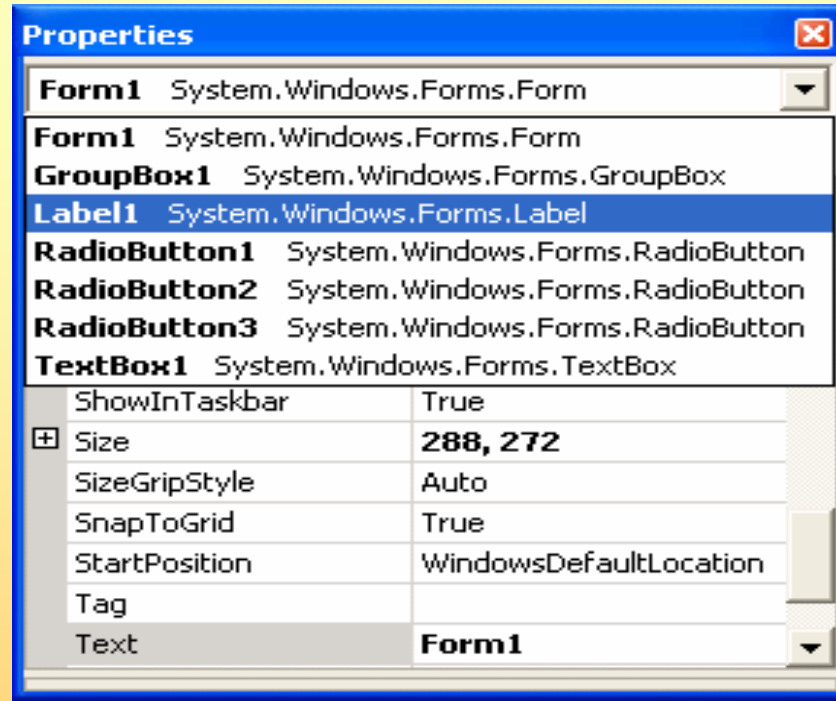
Lesson: Working with Controls

- How to Add Controls to a Form
- How to Set Control Properties
- How to Add Code to Control Events
- How to Use the MessageBox Function

How to Add Controls to a Form

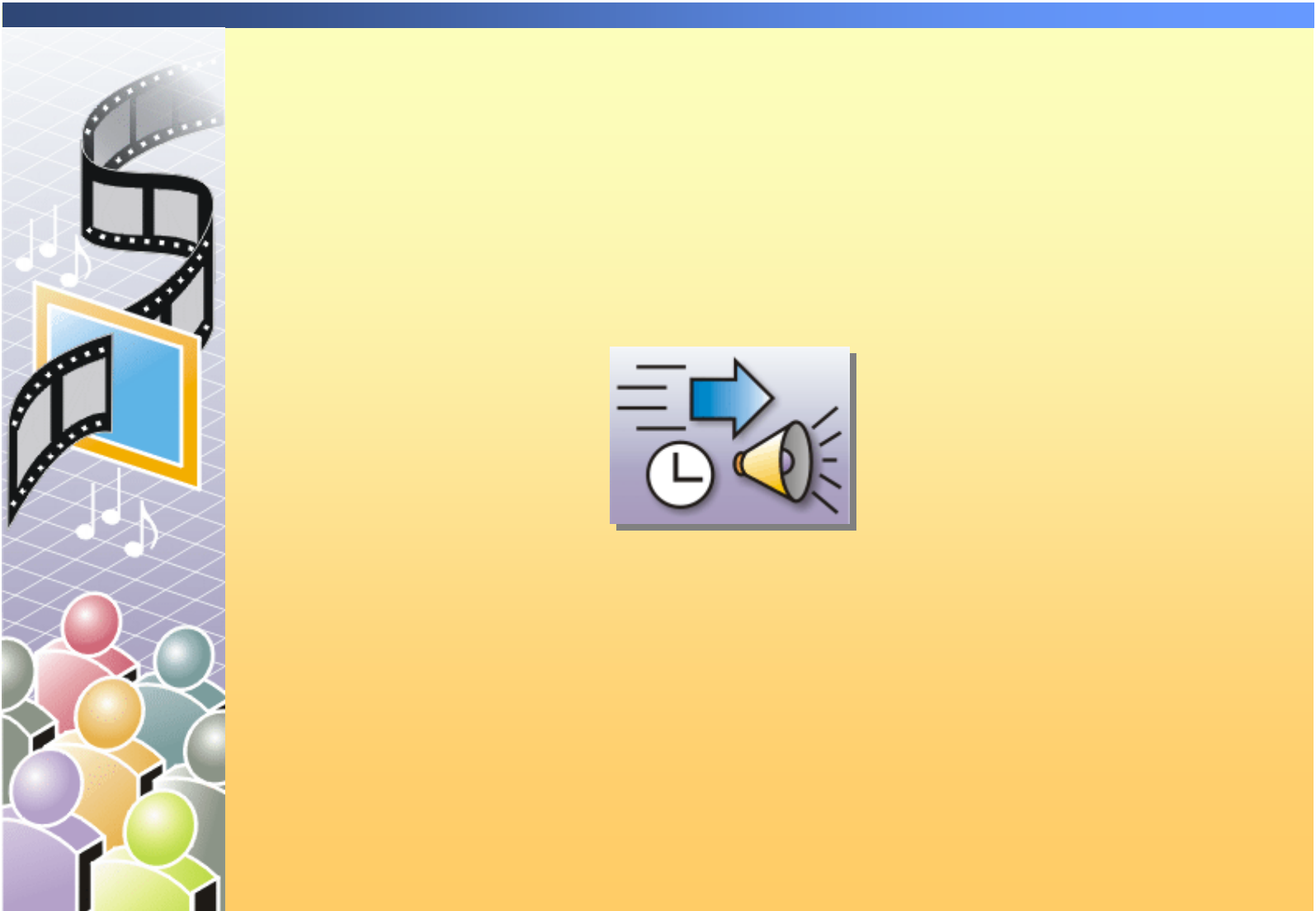


How to Set Control Properties

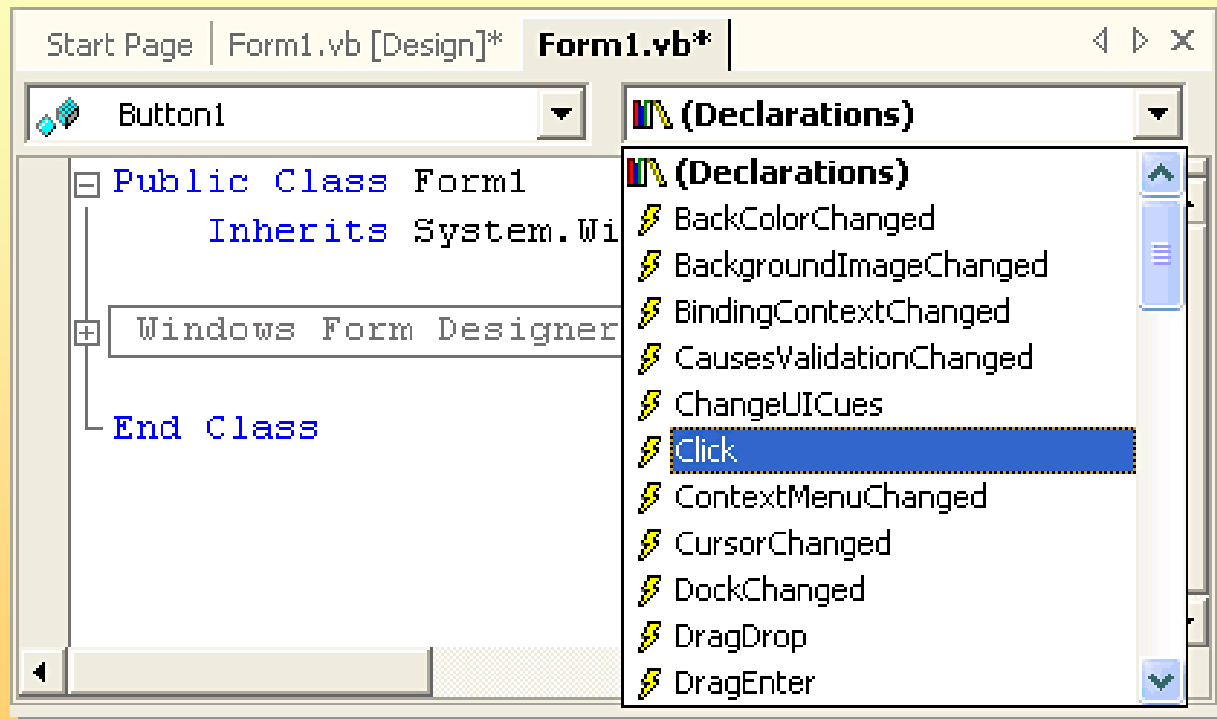


- You can set the same property value for multiple controls at the same time
- Set properties of individual controls to support accessibility guidelines

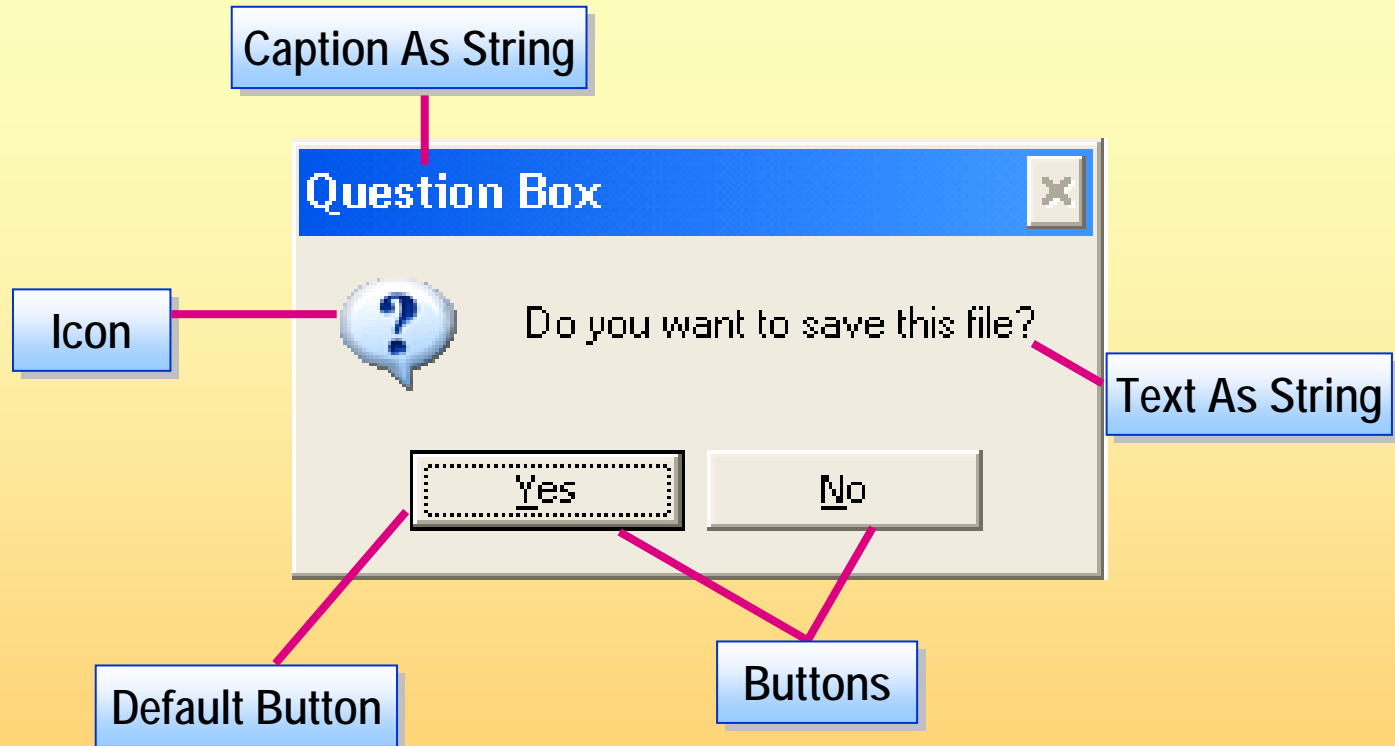
Multimedia: Form and Control Events



How to Add Code to Control Events



How to Use the MessageBox Function



```
MessageBox.Show( )
```


Lesson: Styling Your Code

- Naming Conventions
- How to Format and Document Code

Naming Conventions

■ Rules

- Use only letters, digits, and underscores (`_`)
- Begin with a letter or underscore
- Do not use keywords

Answer42
42Answer



OpenButton
True



■ Guidelines

- **Case:** Use PascalCasing or camelCasing, depending on the element you are naming
- **Mechanics:** Use nouns for objects, verbs for methods, and so on
- **Word choice:** Use terms consistently across code segments

BADSTYLE
_poorstyle
BestStyle



How to Format and Document Code

- Indenting

```
Sub Button1_Click
    Me.Close
End Sub
```

- Line continuation and concatenation

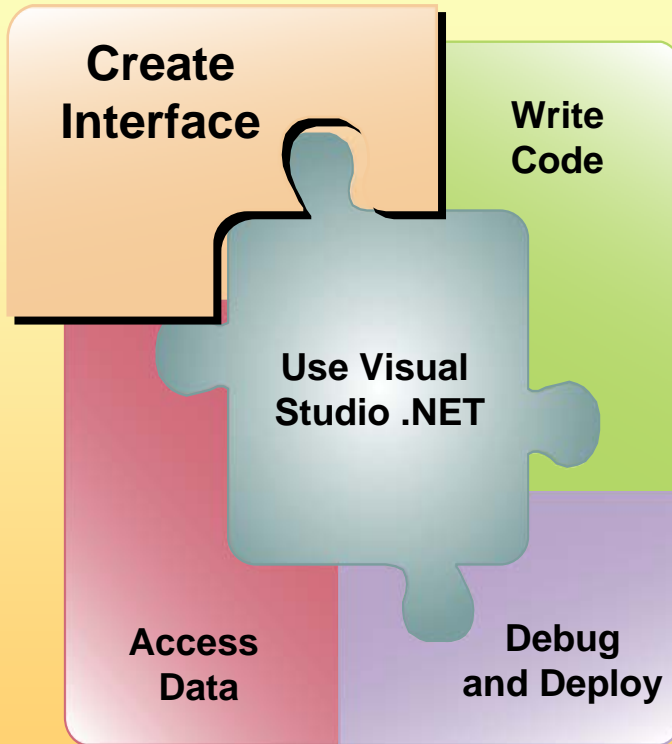
```
MessageBox.Show("User Name = " & UserName.Text & _
    ", Password = " & Password.Text)
```

- Adding comments

```
'Make CalculationForm visible
Dim CalculationForm as new Form2( )
CalculationForm.Show( )
```

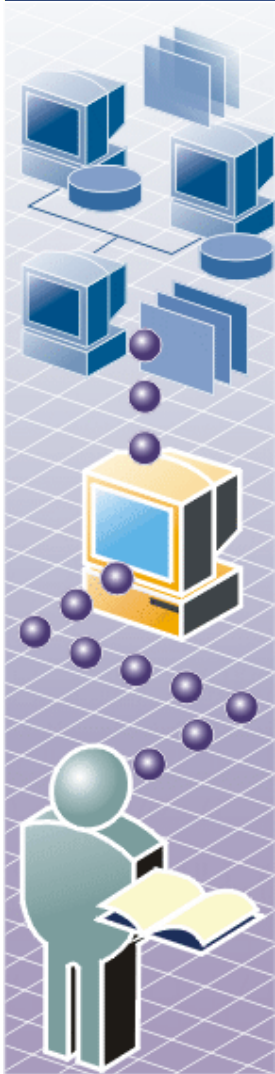
- Add comments to your code to make it more readable and easier to maintain

Review



- Understanding Programming Concepts
- Working with Windows Forms
- Working with Controls
- Styling Your Code

Lab 2.1: Creating the User Interface



- Exercise 1: Creating the Main Form