

Module 1: Input Validation

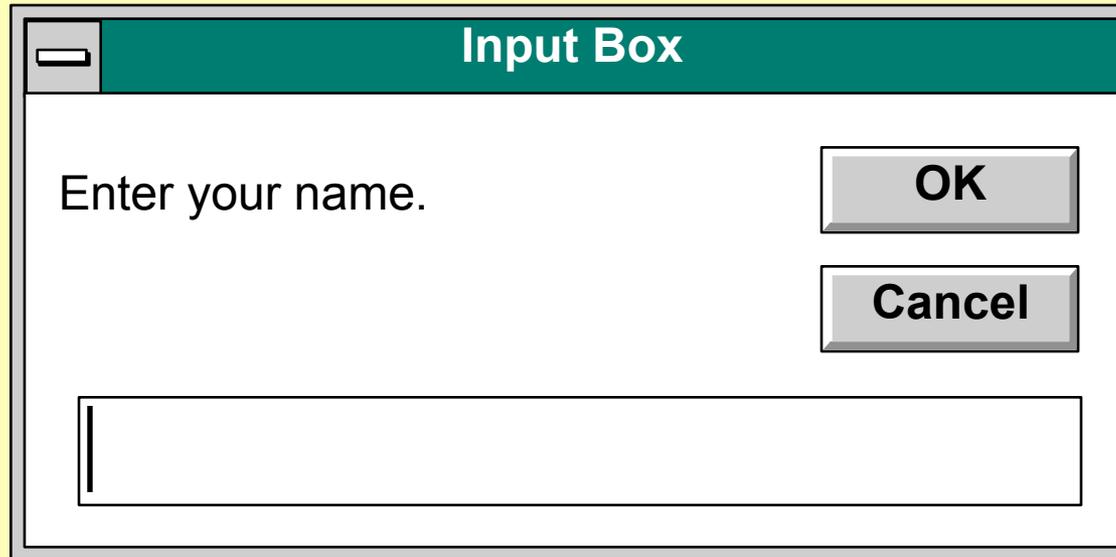
1

Overview

Getting the Input
Validating the Input

1

Using the InputBox\$ Function

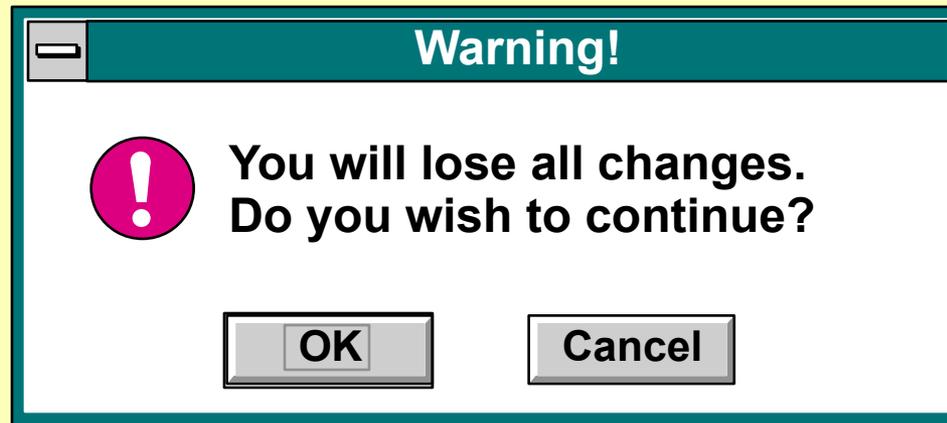


```
UserTypeIn = InputBox$ (MsgText, MsgTitle, Default)
```

Displaying Text with a Label or Message Box

Hello

```
Label1.Caption = "Hello"
```



```
Response = MsgBox (MsgText, MsgType, MsgTitle)
```

Controlling the Tab Order

TabIndex Values

The diagram illustrates a form window titled "Form1" with a tab order sequence. The controls and their corresponding TabIndex values are:

- Form Window: 0
- Name label: 1
- Name text box: 2
- Number label: 3
- Number text box: 4
- City label: 5
- City text box: 6
- OK button: 7
- Cancel button: 8

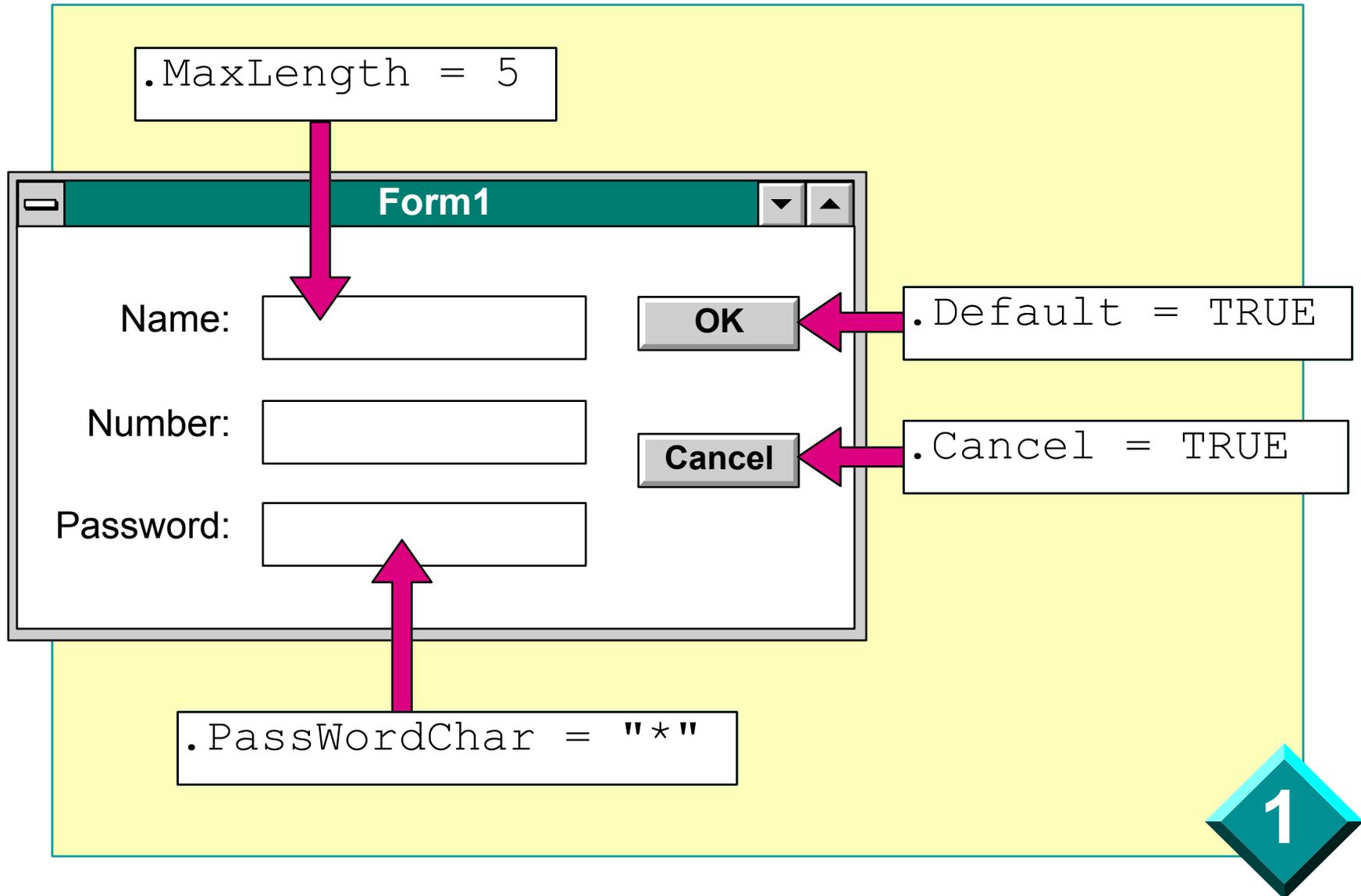
Assigning Access Keys

The screenshot shows a standard Windows application window titled "Form1". Inside the window, there are three text boxes arranged vertically. The first text box is preceded by the label "Name:" with an underlined "N". The second text box is preceded by the label "Number:" with an underlined "N". The third text box is preceded by the label "City:" with an underlined "C". To the right of the text boxes are two buttons: "OK" and "Cancel".

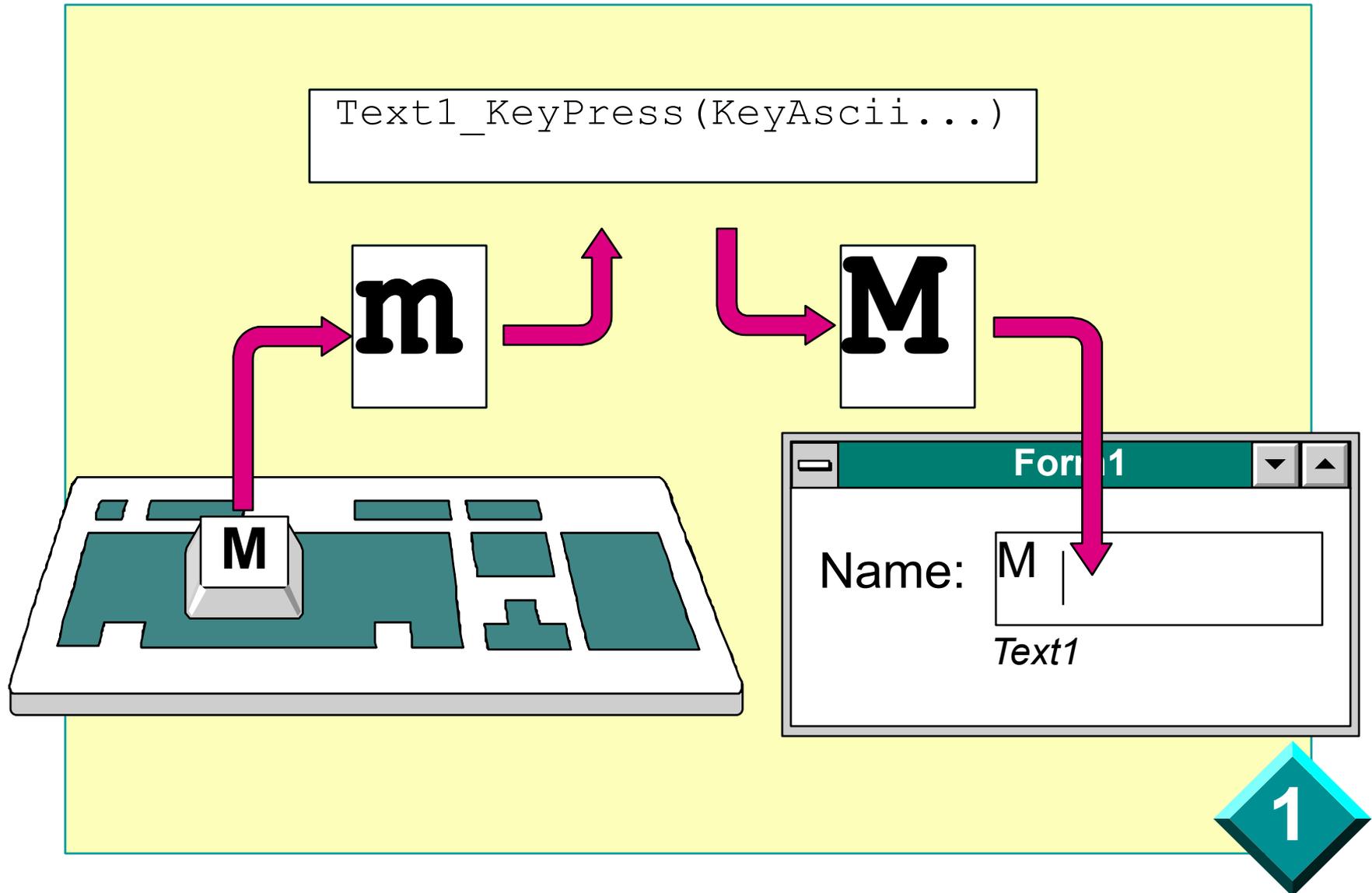
`Label1.TabIndex = 4`

`Text3.TabIndex = 5`

Using Properties to Restrict Data Entry

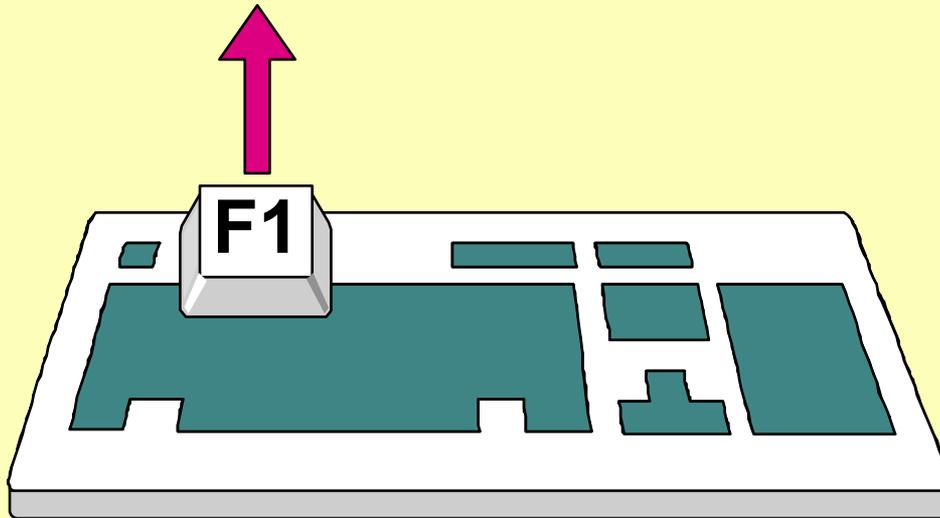


Using KeyPress to Restrict Keystrokes



Using KeyUp and KeyDown to Trap Keystrokes

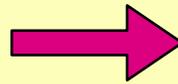
```
Text1_KeyDown(KeyCode...) 'Check Key
```



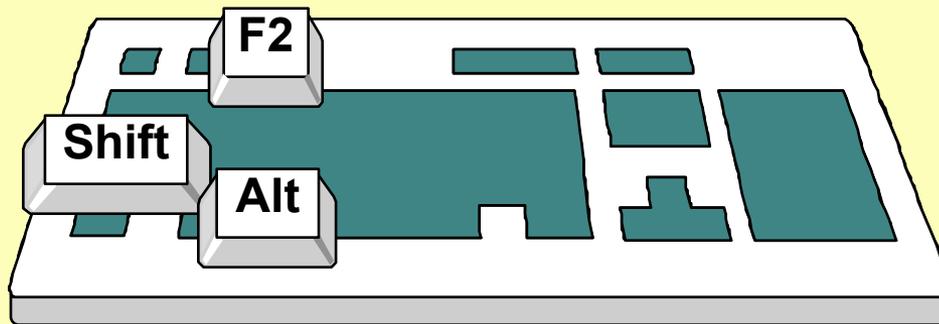
A Form-Level Keyboard Handler

```
Form1.KeyPreview = True
```

```
Form_KeyDown
```



```
Text1_KeyDown
```



Using ENTER Key to Tab Between Controls

Form1

0 Name: 1

2 Number: 3

```
Form_KeyPress  
'If ENTER key pressed  
'Check TabIndex  
'Set new TabIndex  
'Set focus
```

The Masked Edit Control

The diagram illustrates a form titled "Form1" with several input fields and buttons. Pink arrows point from code snippets on the right to the corresponding fields on the form:

- `.Mask = &&&&&` points to the "Name:" field.
- `.Mask = ??` points to the "State" field.
- `.Mask = ##` points to the "Number:" field.
- `.Mask = ###-##-####` points to the "Soc-Sec-No:" field.
- `lblError.Caption
↪ = "Error Message"` points to an empty text box below the "Soc-Sec-No:" field.

At the bottom of the form are "OK" and "Cancel" buttons. A blue diamond with the number "1" is located in the bottom right corner of the slide.

The ValidationError Event

ValidationError Event

MaskedEdit1_ValidationError

↳ (InvalidText as String, StartPosition as Integer)

Analyze InvalidText

Analyze StartPosition

1

String Functions

Function	Purpose
LTrim\$	Deletes left trailing spaces
RTrim\$	Deletes right trailing spaces
Trim\$	Deletes left and right trailing spaces
Mid\$	Returns a string that is part of another string
Left\$	Returns a string consisting of the n leftmost characters of a string
Right\$	Returns a string consisting of the n rightmost characters of a string

String Functions *(continued)*

Function	Purpose
Chr\$	Converts ASCII to single character
Val	Converts String to Numeric
Format\$	Converts Numeric to Formatted String as specified
Str\$	Converts Numeric to String
Instr	Searches Target String for matching Search String: Returns Position
Hex\$/Oct\$	Returns a string consisting of the converted numerical value

Enabling the OK Button

```
Sub ValidateOKButton  
    'Verify Controls
```

Form1

Name:

Number:

OK Cancel

```
mskName_Change  
    ValidateOKButton mskName
```

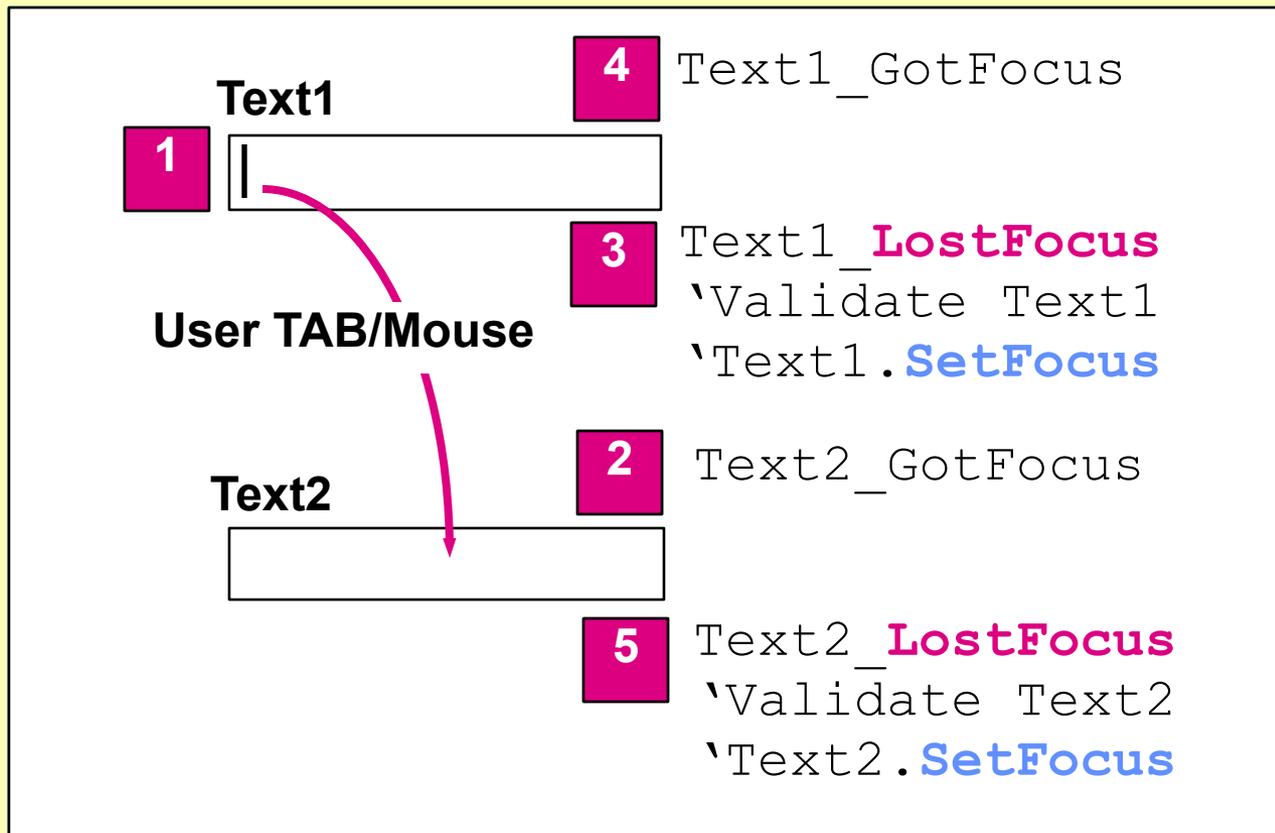
```
mskNumber_Change  
    ValidateOKButton mskNumber
```

Disable at Design Time

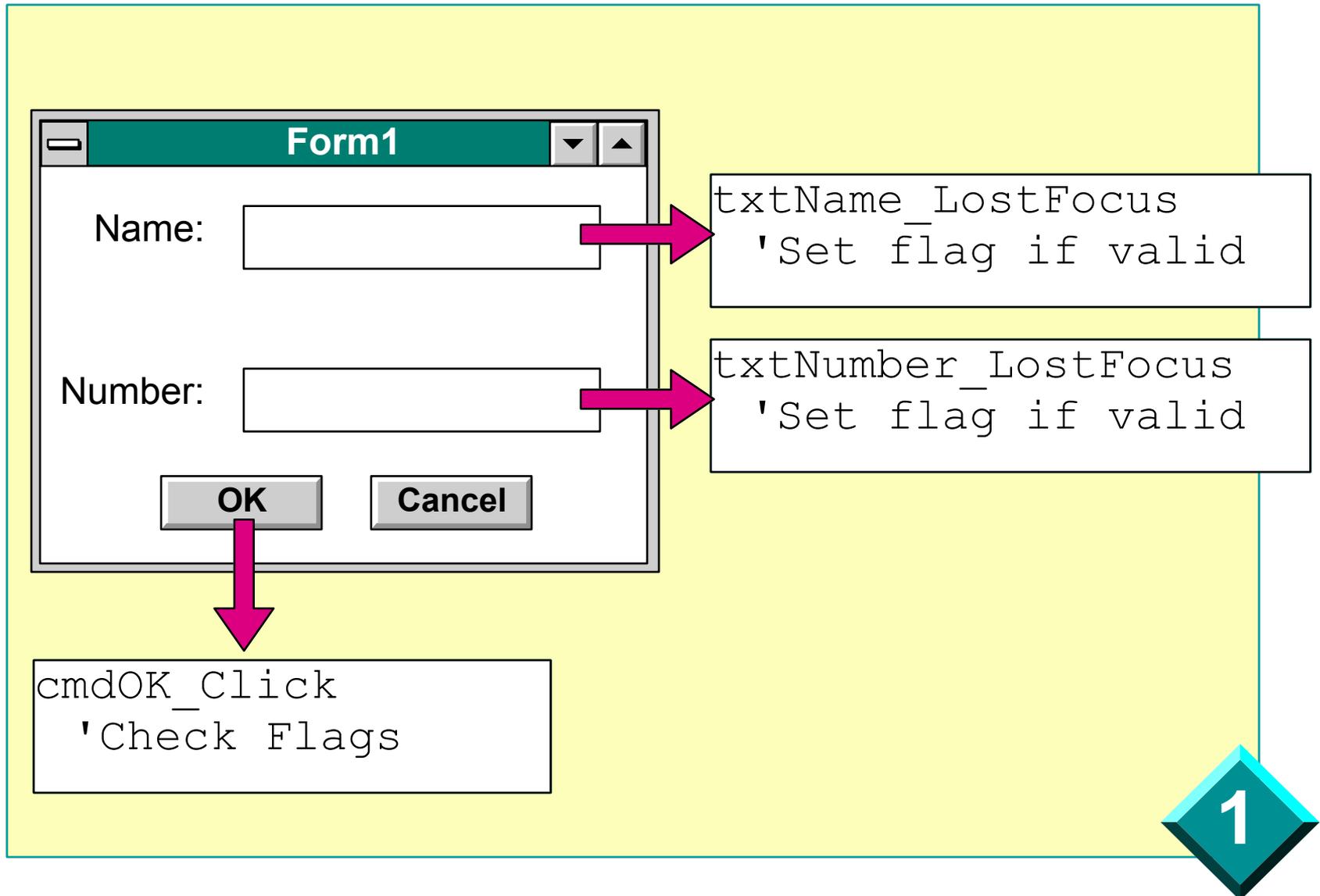
1

Competing for Focus

Do Not Use SetFocus in LostFocus Event



Validate Controls



Detecting Form Focus

**A Form Receives a LostFocus Event Only If
There Are No Controls on the Form**

A Form Receives an Activate or Deactivate Event

Input Validation Hierarchy

Keystroke Validation

Masked Edit Control - Mask Setting;
ValidationError

KeyPress, KeyUp, KeyDown, Change Events

Input Field Validation

MaxLength Property

ValidationError Event; LostFocus Event

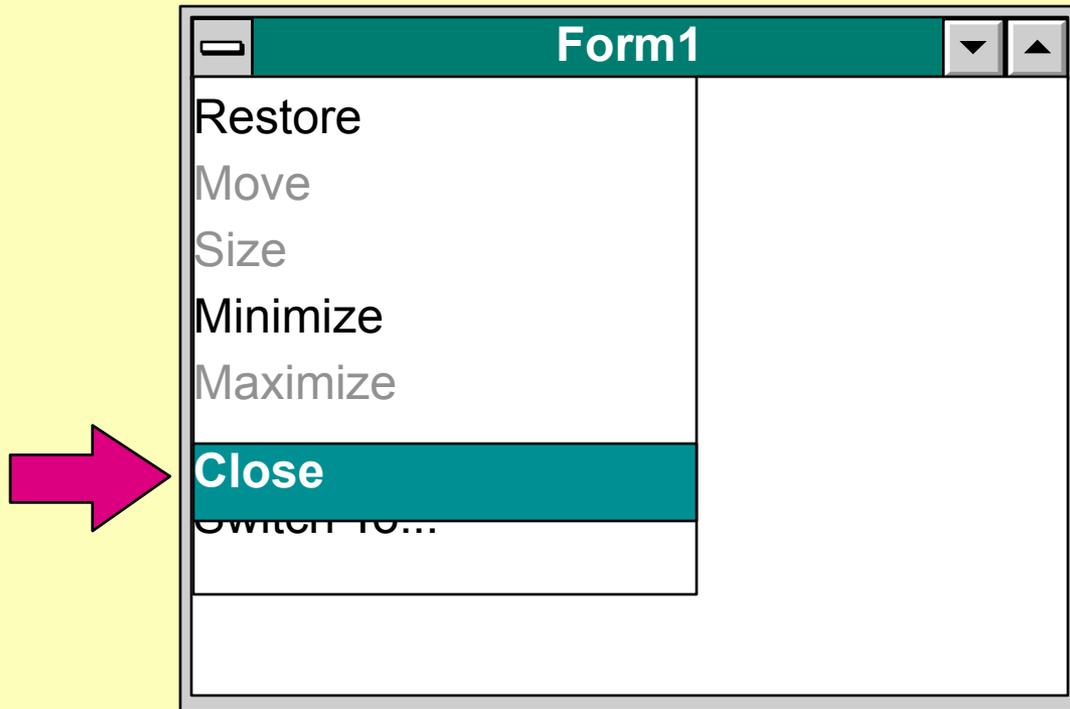
String Analysis

Ensemble Validation

Flags Array

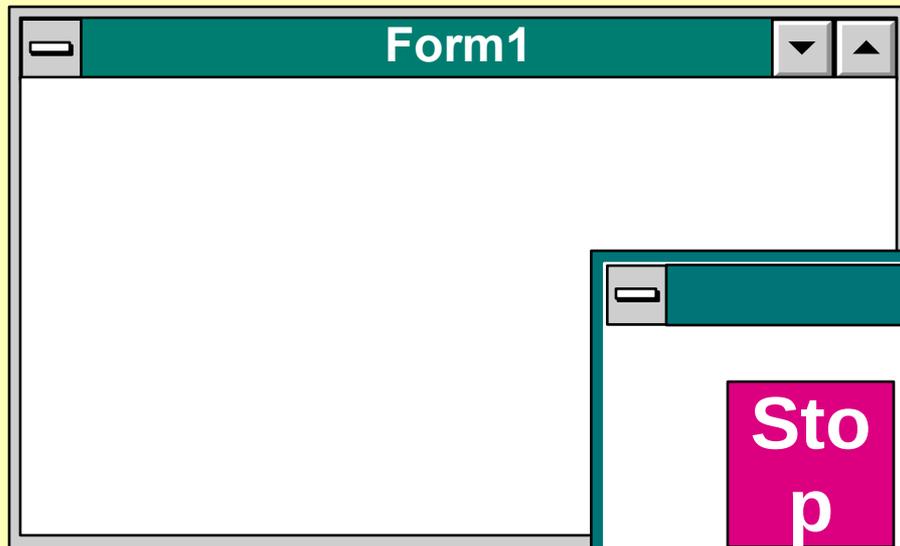
Using QueryUnload

```
Sub Form_QueryUnload (Cancel As Integer,  
    ↪ UnloadMode As Integer)  
    If UnloadMode = FORM_CONTROLMENU Then...
```

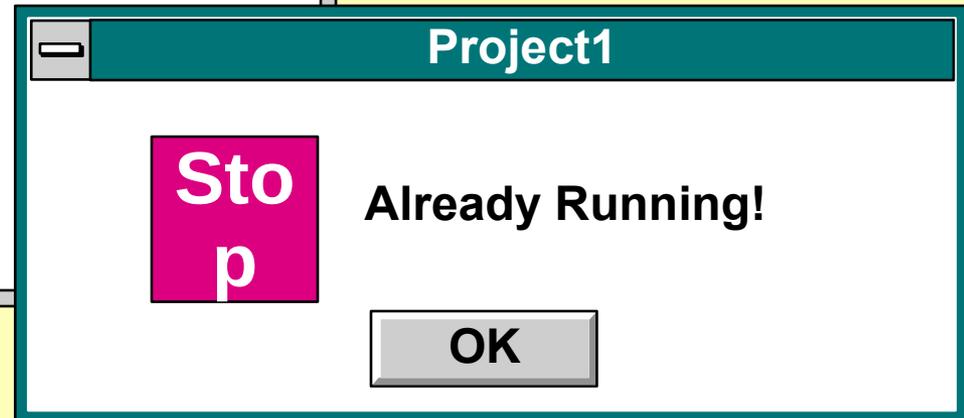


Checking the App Object

```
Sub Form_Load ()  
  If App.PrevInstance...
```



MYPROG.EXE



MYPROG.EXE

1

Summary

Getting the Input
Validating the Input

1