



Branching and Looping

Dr. A. Malathi M.Sc.,M.Phil.,Ph.D.,M.Sc(Psych)
Assistant Professor
PG and Research Department of Computer Science
Government Arts College
Coimbatore - 641018

UNIT II

- Branching and Looping: If...Then, If-Then-Else, Else If, Nested If, Selected-Case, For-Next, Do-Loop, While-Wend, Nested Loops, Stop Statement. VB Controls: Textbox – Checkbox – List Box –Combo Box –Label – Command Button – Directory List –Box –Drive List Box. Assigning Values to Forms and Controls – Naming Controls and Forms – Executing Commands – Displaying O/P Data – Assigning Properties Collectively Using „With“ Block

RELATIONAL OPERATORS

- Greater Than $>$
- Less Than $<$
- Equal To $=$
- Not Equal To $<>$
- Greater Than or Equal To $>=$
- Less Than or Equal to $<=$

Examples

- $X = 27$
- $\text{Error} \leq \text{Abs}(x1 - x2)$
- $C < \text{Sqr}(A + B)$
- $\text{Profit} > (\text{Gross} - \text{Taxes})$

Logical Operators

- And, Or, Xor (exclusive Or), Not, Eqv (equivalent) and Imp (implies).
- The first three operators (And, Or and Xor) are used to combine logical expressions.
- Xor, however, will result in a condition that is true only if one of the expressions is true and the other is false.
- Eqv will result in a condition that is true if both expressions have the same logical value
- Imp will always result in a true condition unless the first expression is true and the second is false.

Logical Operators

Logical And Operator

Expression 1	Expression 2	Expression 1 And Expression 2
False	False	False
False	True	False
True	False	False
True	True	True

Logical Operators

- Logical Or Operator

Expression 1	Expression 2	Expression 1 Or Expression 2
False	False	False
False	True	True
True	False	True
True	True	True

Other Logical Operators

- Logical Xor operator

Expression 1	Expression 2	Expression 1 Xor Expression 2
False	False	False
False	True	True
True	False	True
True	True	False

- Logical Not operator
 - If Not (value = 0) Then ...

Rules of Operator Precedence

- \wedge
- $+$, $-$ (sign operations)
- $*$, $/$
- \backslash
- Mod
- $+$, $-$ (addition and subtraction)
- $\&$
- $=$, $\langle \rangle$, $<$, \leq , $>$, \geq (equality and relational)
- Not
- And, AndAlso
- Or, OrElse
- Xor
- $=$, $+=$, $-=$, $*=$, $/=$, $\backslash=$, $\wedge=$, $\&=$

priority

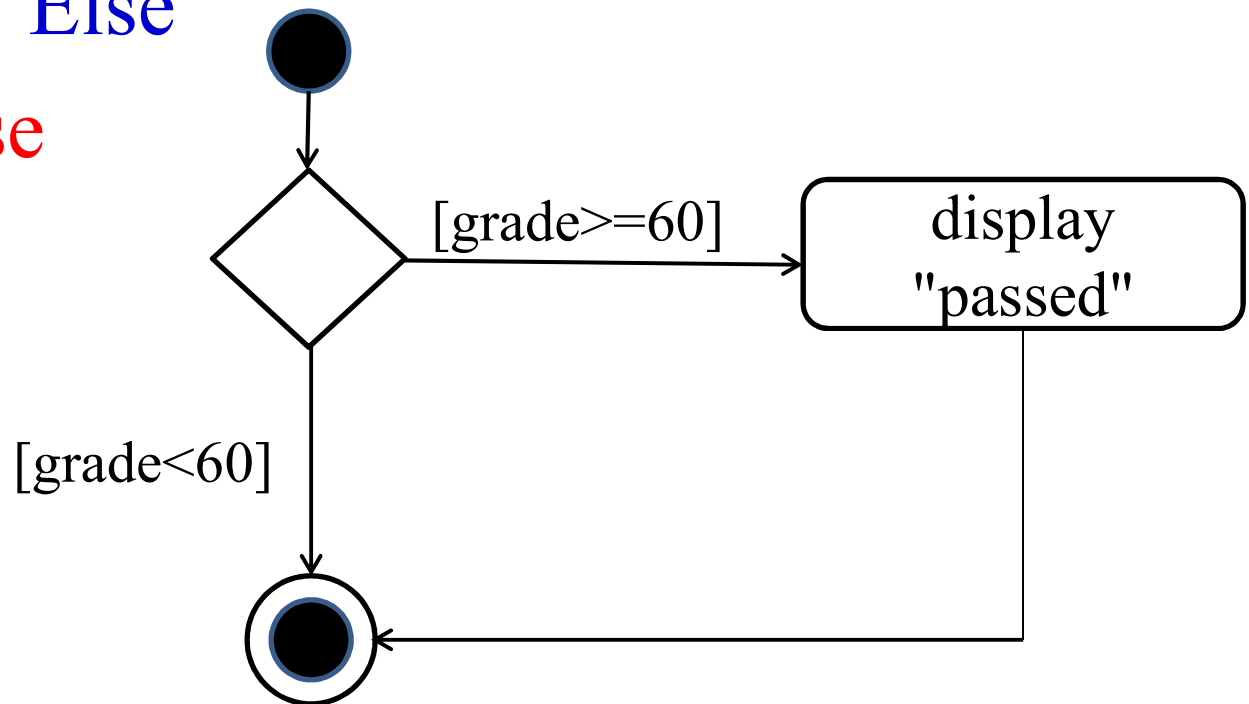
high



low

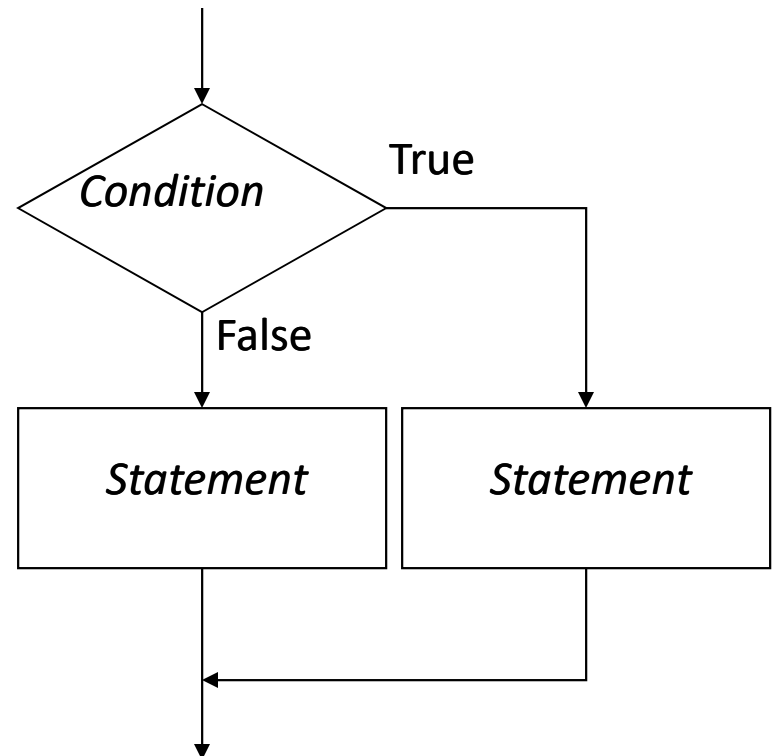
Selection Structure

- If ... Then
- If ... Then ... Else
- **Select ... Case**



If...Then...Else – General Form

If (condition) **Then**
 statement(s)
[Elseif (condition) **Then**
 statement(s)]
[Else
 statement(s)]
End If



If...Then...Else - Example

```
unitsDecimal = Decimal.Parse(unitsTextBox.Text)
If unitsDecimal < 32D Then
    freshmanRadioButton.Checked = True
Else
    freshmanRadioButton.Checked = False
End If
```

Exit Statements

- Exit Do
 - Terminate the repetition statements such as:
 - Do While...Loop, Do Until...Loop, Do...Loop While, Do...Loop Until
- Exit For
 - Terminate For...Next
- Exit While
 - Terminate While...End While
- Exit Select
 - Terminate Select...Case

Continue Statements

- **Continue** statement only *terminates the current iteration*
- **Continue Do**
 - Terminate the repetition statements such as:
 - Do While...Loop, Do Until...Loop, Do...Loop While, Do...Loop Until
- **Continue For**
 - Terminate For...Next
- **Continue While**
 - Terminate While...End While

Repetitive and selection statements

- Repetition statements
 - For ... Next, Do ... Loop While, Do ... Loop Until
- Selection
 - Select ... Case
- Know how to use Exit and Continue statements to break or terminate the current iteration
- Learn how to use logical operators

Selection case

```
Select Case expression  
Case value1  
    executable statements  
Case value2  
    executable statements  
.....  
Case Else  
    executable statements  
End Select
```

Multiple-Selection Statement: Select ... Case

```
Select Case grade
Case 100
    perfectScoreCount += 1 ' increment perfectScoreCount
    aCount += 1 ' increment aCount
Case 90 To 99 ' grade was between 90 and 99
    aCount += 1 ' increment aCount
Case 80 To 89 ' grade was between 80 and 89
    bCount += 1 ' increment bCount
Case 70 To 79 ' grade was between 70 and 79
    cCount += 1 ' increment cCount
Case 60 To 69 ' grade was between 60 and 69
    dCount += 1 ' increment dCount
Case Else ' grade was less than 60
    fCount += 1 ' increment fCount
End Select
```

Multiple-Selection Statement: Select ... Case

- **Case Else** is optional
- If no case matches and there is no **Case Else**, then program control continues with the first statement after **Select ... Case**
- **End Select** terminates the **Select ... Case** statement

Looping with For... Next

- General Form of a For ... Next Statement
- *For initialization To final Value Step increment statement*

Next

Examples of For ... Next

- Declaring the control variable before a For...Next Statement

- Dim counter As Integer

- For counter = 2 To 10 Step 2

- outputLabel.Text &=counter & " "

- Next

- Using expressions in the For...Next statement

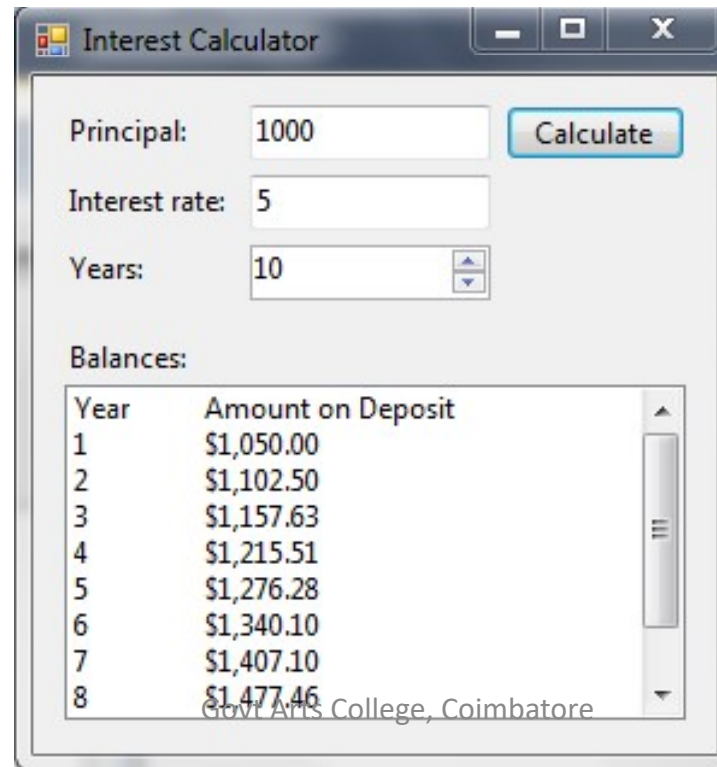
- For j As Integer = x To 4*x*y Step y \ x

Examples of For ... Next

- Default increment: 1
 - For i = 1 To 100
 - For i = 1 To 100 Step 1
- Decrement
 - For i = 100 To 1 Step -1

InterestCalculator

http://media.pearsoncmg.com/ph/esm/deitel/vb_http_2010/codeexamples.html



Interest Calculator

Principal: 1000

Interest rate: 5

Years: 10

Balances:

Year	Amount on Deposit
1	\$1,050.00
2	\$1,102.50
3	\$1,157.63
4	\$1,215.51
5	\$1,276.28
6	\$1,340.10
7	\$1,407.10
8	\$1,477.46

Nested Repetition Statements

- For i = 1 To 10 Step 1
 For j = 1 To 20 Step 2
 ...
 Next i
Next j

Looping with Do-Loop

1. Do While logical expression

.....

executable statements

.....

Loop

2. Do until logical expression

.....

executable statements

.....

Loop

Looping with Do-Loop

3. Do

.....

executable statements

.....

Loop While logical expression

4. Do

.....

executable statements

.....

Loop until logical expression

Example

1. sum = 0
count = 1
Do
 sum = sum + count
 count = count + 1
Loop While count <= 10
2. sum = 0
count = 1
Do While count <= 10
 sum = sum + count
 If sum >= 10 Then
 Exit Do
count = count + 1
Loop

Looping with While-Wend

The general form

While logical expression

.....

executable statements

.....

Wend

Example:

sum = 0

count = 1

While count <= 10

 sum = sum + count

 count = count + 1

Wend

Do...Loop While and Do...Loop Until

- The loop body is always executed at least once

```
Dim product As Integer = 1
```

```
Do
```

```
    product = product * 3
```

```
Loop While product <=100
```

```
Dim product As Integer = 1
```

```
Do
```

```
    product = product * 3
```

```
Loop Until product >100
```

The Stop Statement

- The Stop statement is used to terminate the execution at any point in the program.
- The statement consists simply of the keyword Stop.
- This statement may appear anywhere in a Visual Basic program except at the very end.
- Multiple Stop statements may appear in the same program, as dictated by the program logic.
- However, modern programming practice tends to avoid the use of the Stop statement.

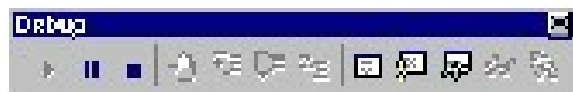
The Toolbox, Toolbars, and Controls

- Visual Basic Integrated Development Environment includes:
 - Standard toolbar
 - Form Editor toolbar (optional)
 - Edit toolbar (optional)
 - Debug toolbar (optional)
 - Toolbox

Displaying the Toolbars



Standard Toolbar



Debug Toolbar

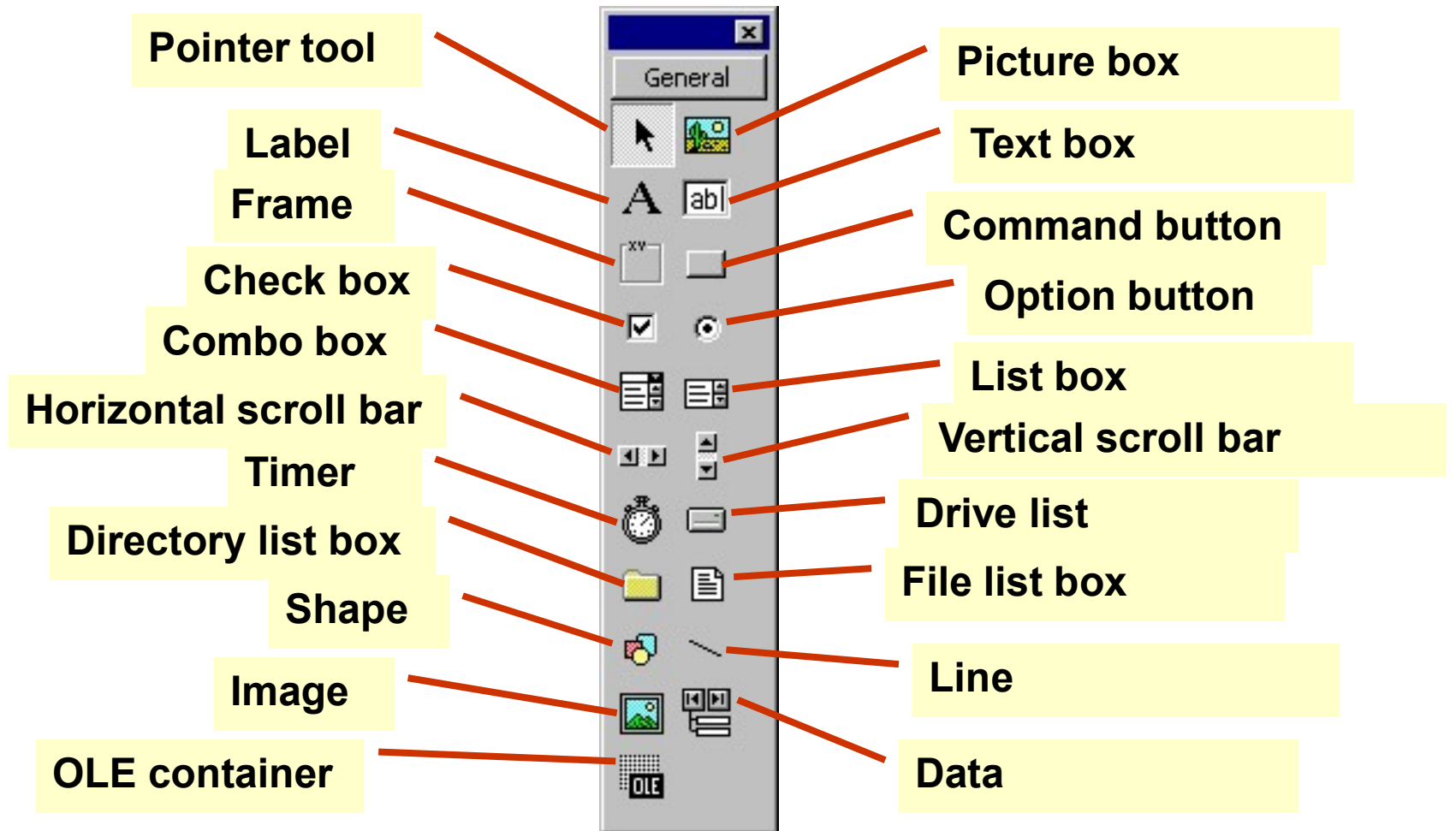


Edit Toolbar



Form Editor Toolbar

Controls



Controls...

Pointer - use to move or change the size of a control

Picture box - use to display graphics

Image - use to display a graphic, uses fewer resources than a picture box

Label - use for text that the user cannot change

Text box - use for text that the user can enter or change

Frame - use to group controls together

Command button - use to create a clickable button

Controls...

Check box - use for choices where more than one choice can be selected

Option button - use for choices where choices are mutually exclusive

Combo box - user can either select an item from the list or enter a value

List box - use to display a list of items where only one choice is allowed

Horizontal or Vertical Scroll bars - use to quickly navigated through a long list of items or a large amount of information

Controls...

Timer - use like a stopwatch to trigger events

Drive list box - displays list of disk drives

Directory list box - displays list of folders

File List box - displays a list of files

Shape - use to draw a rectangle, square, oval or circle

Line - use to draw a variety of lines

OLE Container - use to link or embed OLE objects

Data - use to create applications from many types of databases

Naming Controls and forms

Object - combination of code and data that is treated as a unit

Class - formal definition of an object that defines the properties and methods of an object

Method - a procedure that acts on an object

Instance - one of a set of object that belong to the same class

Naming Controls

Follow these rules when naming objects:

- Name must begin with a letter
- Name must contain only letters, numbers, and the underscore character
- Neither punctuation marks nor spaces are allowed
- Name can be a maximum of 40 characters

Control	Prefix	Control	Prefix
Check Box	chk	MS Chart	ch
Combo Box	cbo	MS Tab	mst
Command Button	cmd	OLE Container	ole
Common Dialog	dlg	Option Button	opt
Date Picker	dtp	Picture Box	pic
Form	frm	ProgressBar	prg
Frame	fra	RichTextBox	rtf
Grid	grd	Shape	shp
Hierarchical Flexgrid	flex	Slider	sld
Horizontal Scroll Bar	hsb	StatusBar	sta
Image	img	SysInfo	sus
ImageList	ils	TabStrip	tab
Label	lbl	Text Box	txt
Line	lin	Timer	tmr
List Box	lst	Toolbar	tlb
List View	lww	TreeView	tre
Menu	mnu	UpDown	upd
Month View	mww	Vertical Scroll Bar	vsb

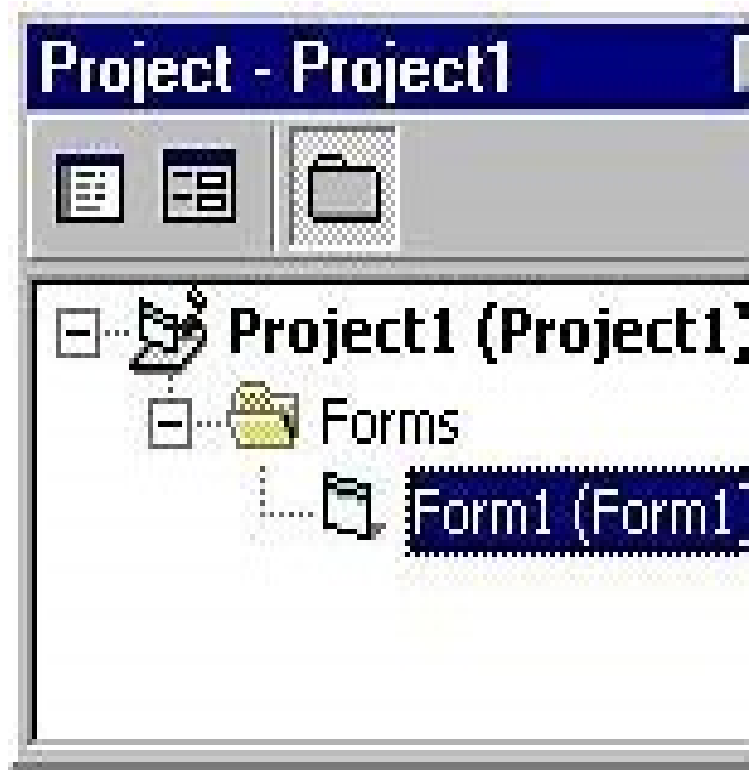
Assigning Property Values To Forms & Controls

- In general terms, a property assignment is written as `object_name.property = value`
- Each of the following commands assigns a run-time value to a text box property.
`Text1.Text = "Welcome to Visual Basic"`
`txtMessage.Text = "Welcome to Visual Basic"`
`txtMessage.Height = 300`

Executing Commands

- An event occurs when the user takes some action, such as clicking on a control icon, or dragging an icon to another location.
- Each event procedure begins with a Sub statement, such as `Private Sub Command1_Click()`, and ends with an End Sub statement.
- Between the Sub and End Sub statements is a group of instructions.
- The parentheses in the Sub statement may contain arguments – special variables that are used to transfer information between the event procedure and the “calling” routine.
- Command buttons are often used to execute Visual Basic event procedures.
- The statements within the event procedure may involve the properties of controls other than the command button.

The Visual Basic Project Window



Displaying Output Data

- The most straightforward way to display output data is with a label or a text box.
- A label can only display output data, though a text box can accept input data as well as display output data.
- Both of these controls process information in the form of a string.
- This is not a serious limitation, however, because numeric values can easily be converted to strings via the Str function.
- To display output using a label, the basic idea is to assign a string containing the desired output information to the label's Caption property.
- when displaying output using a text box, a string containing the desired output information is assigned to the text box's Text property.

Examples

- Private Sub Form_Load()
With Label1 .Caption = "Say Hello, in . . ." .Font.Size = 10
End With
- With Combo1 .List(0) = "French" .List(1) = "German" .List(2) = "Hawaiian" .List(3) = "Hebrew" .List(4) = "Italian" .List(5) = "Japanese" .List(6) = "Spanish" .Text = "Language . . ."
.Font.Size = 10
End With
- With Text1 .Text = "" .BackColor = &H8000000A 'Gray
.BorderStyle = 0 .Font.Size = 14
End With

References

- SCHAUM'S OUTLINE OF THEORY AND PROBLEMS OF PROGRAMMING WITH VISUAL BASIC
- [http://media.pearsoncmg.com/ph/esm/deitel/vb htp 2010/odeexamples.html](http://media.pearsoncmg.com/ph/esm/deitel/vb_htp_2010/odeexamples.html)

The End