

# DECISIONS

**Robert Burtch**  
**Surveying Engineering Department**  
**Ferris State University**

## RELATIONAL AND LOGICAL OPERATORS

- Condition – expression involving relational operators that is either true or false
  - Conditions can also be incorporated into logical operators
- Less than (<) operator can be applied to both numbers and strings
  - String *a* is less than string *b* if *a* precedes *b* alphabetically
  - Digits precede uppercase letters which precede lowercase letters
- Conditions can involve variables, numeric operators, functions

# MATHEMATICAL RELATIONAL OPERATORS

Mathematical Notation	Visual Basic Notation	Numeric Meaning	String Meaning
=	=	Equal to	Identical to
≠	<>	Unequal to	Different from
<	<	Less than	Precedes alphabetically
>	>	Greater than	Follows alphabetically
≤	<=	Less than or equal to	Precedes alphabetically or is identical to
≥	>=	Greater than or equal to	Follows alphabetically or is identical to

# LOGICAL OPERATORS

- Programming often require more complex conditions
- 3 main logical operators are: And, Or, Not
- If *cond1* & *cond2* are conditions, then  

$$\text{cond1 And cond2}$$
is true if both conditions are true, otherwise it is false
- Condition  

$$\text{cond1 Or cond2}$$
is true is either *cond1* or *cond2* or both are true, otherwise it is false
- Condition  

$$\text{Not cond1}$$
is true if *cond1* is false

## LOGICAL OPERATORS ( n has value 4 and string answ has value "Y")

- a)  $(2 < n)$  And  $(n < 6)$
- b)  $(2 < n)$  Or  $(n = 6)$
- c) Not  $(n < 6)$
- d)  $(\text{answ} = \text{"Y"})$  Or  $(\text{answ} = \text{"y"})$
- e)  $(\text{answ} = \text{"Y"})$  And  $(\text{answ} = \text{"y"})$
- f) Not  $(\text{answ} = \text{"y"})$
- g)  $((2 < n)$  And  $(n = 5 + 1))$  Or  $(\text{answ} = \text{"No"})$
- h)  $((n = 2)$  And  $(n = 7))$  Or  $(\text{answ} = \text{"Y"})$
- i)  $(n = 2)$  And  $((n = 7)$  Or  $(\text{answ} = \text{"Y"}))$

## OPERATOR HIERARCHY FOR LOGICAL EXPRESSIONS

- All arithmetic operations carried out
  - All expressions involving  $>$ ,  $<$ , and  $=$  are evaluated to true or false
  - Logical operators next applied, in order: Not, then And, and finally Or
- Like mathematical expressions, parenthesis can be used to modify the hierarchy

## RELATIONAL AND LOGICAL OPERATORS

- Conditions involving numeric variables is different from an algebraic truth
  - EX:  $(a + b) < 2 * a$  is not valid algebraic truth because it is not valid for all values of a and b, but VB will evaluate relationship
- Conditions evaluate to either True or False
  - Often called truth values of condition
- Do not use equation like  $2 < n < 5$  because VB will not evaluate it as intended
  - Use  $(2 < n)$  And  $(n < 5)$
- Not  $(n < m)$  is not the same as  $(n > m)$ . Instead, use  $(n >= m)$

## IF BLOCK

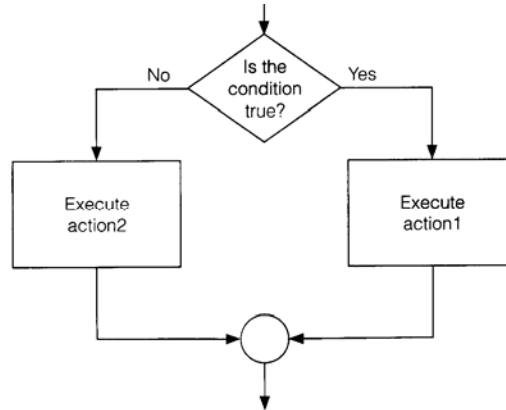
- Program decides course of action based on whether a condition is true or false
- Form:

```
If condition Then
    action1
Else
    action2
End if
```
- Action1 performed if condition is true and action 2 performed if the condition is false

# IF BLOCK

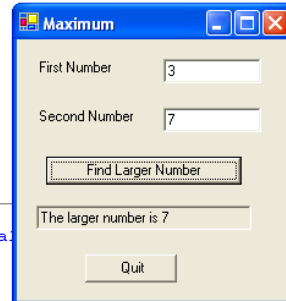
- Pseudocode and flowchart of If block:

```
If condition is true Then  
    action1  
Else  
    action2  
End If
```



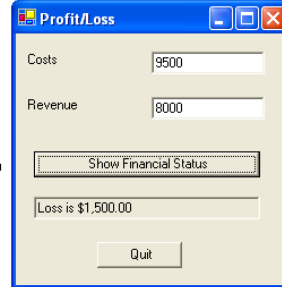
# EXAMPLE IF BLOCK

```
Private Sub btnFindLarger_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
    Dim num1, num2, largerNum As Double  
    num1 = Cdbl(txtFirstNum.Text)  
    num2 = Cdbl(txtSecondNum.Text)  
    If num1 > num2 Then  
        largerNum = num1  
    Else  
        largerNum = num2  
    End If  
    txtResult.Text = "The larger number is " & largerNum  
End Sub  
  
Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
    End  
End Sub  
End Class
```



# EXAMPLE IF BLOCK

```
Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnST
    Dim costs, revenue, profit, loss As Double
    costs = Cdbl(txtCosts.Text)
    revenue = Cdbl(txtRev.Text)
    If costs = revenue Then
        txtResult.Text = "Break even"
    Else
        If costs < revenue Then
            profit = revenue - costs
            txtResult.Text = "Profit is " & FormatCurrency(profit)
        Else
            loss = costs - revenue
            txtResult.Text = "Loss is " & FormatCurrency(loss)
        End If
    End If
End Sub
```

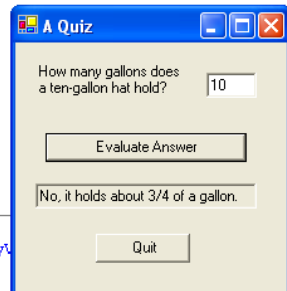


# EXAMPLE IF BLOCK

- Logical operator in condition

```
Private Sub btnEvaluate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    'Evaluate answer
    Dim answer As Double
    answer = Cdbl(txtAnswer.Text)
    If (answer >= 0.5) And (answer <= 1) Then
        txtSolution.Text = "Good, "
    Else
        txtSolution.Text = "No, "
    End If
    txtSolution.Text &= "it holds about 3/4 of a gallon."
End Sub

Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    End
End Sub
```

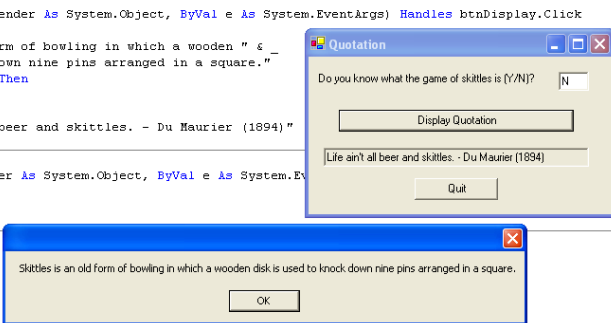


# EXAMPLE IF BLOCK

- Program offers assistance to user before presenting quotation

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnDisplay.Click
    Dim message As String
    message = "Skittles is an old form of bowling in which a wooden " & _
        "disk is used to knock down nine pins arranged in a square."
    If txtAnswer.Text.ToUpper = "N" Then
        MsgBox(message, , "")
    End If
    End If
    txtQuote.Text = "Life ain't all beer and skittles. - Du Maurier (1894)"
End Sub

Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnQuit.Click
    End
End Sub
End Class
```



# ElseIf Clauses

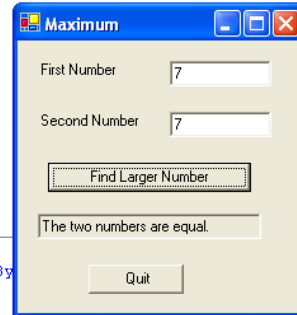
- Extension: two possible alternatives with use of Elseif clause
- Form:

```
If condition Then
    action1
Elseif condition2 Then
    action2
Elseif condition3 Then
    action3
Else
    action4
End If
```

# IF BLOCK EXAMPLE WITH ELSEIF CLAUSE

```
Private Sub btnFindLarger_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim num1, num2, largerNum As Double
    num1 = CDb1(txtFirstNum.Text)
    num2 = CDb1(txtSecondNum.Text)
    If num1 > num2 Then
        txtResult.Text = "The larger number is " & num1
    ElseIf (num2 > num1) Then
        txtResult.Text = "The larger number is " & num2
    Else
        txtResult.Text = "The two numbers are equal."
    End If
End Sub

Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
End Sub
```

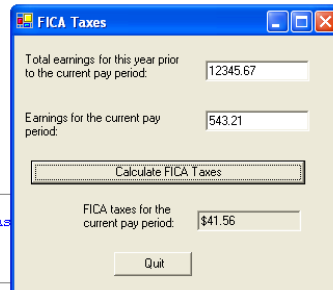


# IF BLOCK WITH FUNCTION

```
Private Sub btnCalculate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnCalcul
    Dim FicaTaxes As Double
    FicaTaxes = CalculateFICA(CDb1(txtToDate.Text), CDb1(txtCurrent.Text))
    txtTax.Text = FormatCurrency(FicaTaxes)
End Sub

Function CalculateFICA(ByVal ytdEarnings As Double, ByVal curEarnings As Double) As Double
    'Calculate Social Security benefits tax and Medicare tax
    'for a single pay period in 2002
    Dim socialSecurityBenTax, medicareTax As Double
    If (ytdEarnings = curEarnings) <= 84900 Then
        socialSecurityBenTax = 0.062 * curEarnings
    ElseIf ytdEarnings < 84900 Then
        socialSecurityBenTax = 0.062 * (84900 - ytdEarnings)
    End If
    medicareTax = 0.0145 * curEarnings
    Return socialSecurityBenTax + medicareTax
End Function

Private Sub btnQuit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
End Sub
End Sub
Class
```



## IF BLOCKS

- Actions of If block and words Else and Elself do not have to be indented – but does improve readability
- If blocks contained within another If block are called nested If blocks
- Make If blocks easier to follow – e.g. code to right

```
If cond1 Then
  If cond2 Then
    action
  End If
End If

If cond1 And cond2 Then
  action
End If
```

## SELECT CASE BLOCKS

- Efficient decision-making structure when choosing among several actions
- Select Case choices determined by value of expression called a selector
- Each action preceded by clause:

*Case valuList*

– Where *valuList* itemizes values of selector for which action should be taken

# EXAMPLE SELECT CASE BLOCK

```
Private Sub btnEvaluate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim position As Integer 'Selector
    position = CInt(txtPosition.Text)
    Select Case position
        Case 1
            txtOutcome.Text = "Win"
        Case 2
            txtOutcome.Text = "Place"
        Case 3
            txtOutcome.Text = "Show"
        Case 4, 5
            txtOutcome.Text = "You almost placed in the money."
        Case Else
            txtOutcome.Text = "Out of the money."
    End Select
End Sub
```



# EXAMPLE SELECT CASE BLOCK

- Can specify range of values as:

```
Private Sub btnEvaluate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    'Describe finishing positions in a horse race
    Dim position As Integer
    position = CInt(txtPosition.Text)
    Select Case position
        Case 1 To 3
            txtOutcome.Text = "In the money. Congratulations."
        Case Is >= 4
            txtOutcome.Text = "Not in the money."
    End Select
End Sub
```



## SELECT CASE BLOCKS

- Typical form of Select Case block:

```
Select Case selector
  Case valueList1
    action1
  Case valueList2
    action2
  :
  Case Else
    action of last resort
End Select
```

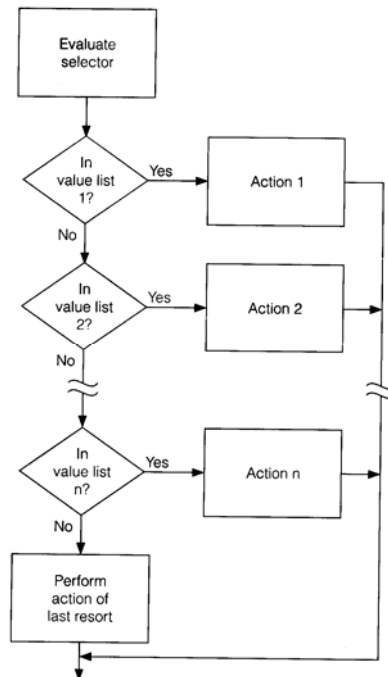
- Case Else is optional

## SELECT CASE BLOCKS

- Each value list contains one or more of:
  - Constant
  - Variable
  - Expression
  - Inequality sign preceded by *Is* and followed by a constant, variable, or expression
  - Range expressed in form *a To b*, where *a* and *b* are constants, variables or expressions
- Different items in same list must be separated by commas

# SELECT CASE BLOCKS

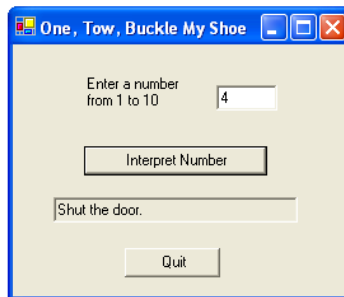
- Basic looks for first value-list including value of selector and carries out associated action
  - If selector appears in 2 different value list, only action with first action list will be carried out
- If no Case Else clause, program will continue with statement following Select Case block



# SELECT CASE BLOCKS

```

Private Sub btnInterpret_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    'One, Two, Buckle My Shoe
    Dim x As Integer = 2, y As Integer = 3
    Dim num As Integer
    num = CInt(txtNumber.Text)
    Select Case num
        Case y - x, x
            txtPhrase.Text = "Buckly my shoe."
        Case Is <= 4
            txtPhrase.Text = "Shut the door."
        Case x + y To x * y
            txtPhrase.Text = "pick up sticks."
        Case 7, 8
            txtPhrase.Text = "Lay them straight."
        Case Else
            txtPhrase.Text = "start all over again."
    End Select
End Sub
  
```



# SELECT CASE BLOCK USING STRING VARIABLES

```
Private Sub btnInterpret_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnInterp  
    'Quiz  
    Dim firstName As String  
    firstName = txtName.Text  
    Select Case firstName  
        Case "Thomas"  
            txtReply.Text = "Correct."  
        Case "Woodrow"  
            txtReply.Text = "Sorry, his full name was Thomas Woodrow Wilson."  
        Case "President"  
            txtReply.Text = "Are you for real?"  
        Case Else  
            txtReply.Text = "Nice try, but no cigar."  
    End Select  
End Sub
```

