Chapter 4 – Decisions

4.1 Relational and Logical Operators
4.2 If Blocks
4.3 Select Case Blocks
4.4 Input via User Selection
4.1 Relational and Logical Operators

- ANSI Values
- Relational Operators
- Logical Operators
- Boolean Data Type
- Two Boolean-Valued Methods
- A Boolean-Valued Function
Condition

• **A condition** is an expression involving relational and/or logical operators
• The value of the condition is Boolean – that is, True or False
ANSI Character Set

A numeric representation for every key on the keyboard and for other assorted characters.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>(space)</td>
<td>48</td>
<td>0</td>
</tr>
<tr>
<td>33</td>
<td>!</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>34</td>
<td>&quot;</td>
<td>57</td>
<td>9</td>
</tr>
<tr>
<td>35</td>
<td>#</td>
<td>65</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>66</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97</td>
<td>a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>98</td>
<td>b</td>
</tr>
<tr>
<td></td>
<td></td>
<td>122</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td></td>
<td>123</td>
<td>{</td>
</tr>
<tr>
<td></td>
<td></td>
<td>125</td>
<td>}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>126</td>
<td>~</td>
</tr>
</tbody>
</table>
A numeric representation for every key on the keyboard and for other assorted characters.

<table>
<thead>
<tr>
<th>162  ×</th>
<th>177  ±</th>
<th>181  μ</th>
<th>190  ¼</th>
</tr>
</thead>
<tbody>
<tr>
<td>169  ©</td>
<td>178  ²</td>
<td>188  ¼</td>
<td>247  ÷</td>
</tr>
<tr>
<td>176  °</td>
<td>179  ³</td>
<td>189  ½</td>
<td>248  ø</td>
</tr>
</tbody>
</table>
Chr Function

For $n$ between 0 and 255,

$$\text{Chr}(n)$$

is the string consisting of the character with ANSI value $n$.

Examples:  \(\text{Chr}(65) \text{ is } A\)

\(\text{Chr}(162) \text{ is } \$\)
Asc Function

For a string \( str \),

\[
\text{Asc}(str)
\]

is ANSI value of the first character of \( str \).

Examples: \[
\text{Asc}("A") \text{ is } 65
\]

\[
\text{Asc}("\&25") \text{ is } 162
\]
Relational Operators

<  less than
\le  less than or equal to
>  greater than
\ge  greater than or equal to
=  equal to
<>  not equal to

ANSI values are used to decide order for strings.
A condition is an expression involving relational and/or logical operators.

Result of the condition is True or False.
Example

When $a = 3$, $b = 4$

$$(a + b) < 2 \times a$$

\[
\begin{align*}
3 + 4 &= 7 \\
2 \times 3 &= 6
\end{align*}
\]

7 is NOT less than 6 and so the value of the expression is False
Another Example

\[ a = 4 \quad b = 3 \quad c = "hello" \]

\[ (c.\text{Length} - b) = \left(\frac{a}{2}\right) \]

\[
\begin{align*}
5 - 3 &= 2 \\
4 \div 2 &= 2
\end{align*}
\]

**True** because 2 equals 2
Relational Operator Notes

- Relational operators are binary – they require an operand on both sides of the operator
- Value of a relational expression will always be True or False
Logical Operators

Used with Boolean expressions

- **Not** – makes a False expression True and vice versa
- **And** – will yield a True if and only if both expressions are True
- **Or** – will yield a True if at least one of both expressions are True
Example 4.3

\[ n = 4, \ \text{answ} = "Y" \] Are the following expressions true or false?

- \( \text{Not} (n < 6) \)
- \( (\text{answ} = "Y") \) Or \( (\text{answ} = "y") \)
- \( (\text{answ} = "Y") \) And \( (\text{answ} = "y") \)
- \( \text{Not}(\text{answ} = "y") \)
Boolean Expression

• An expression that evaluates to either True or False is said to have Boolean data type.

• Example:
  The statement
  
  `txtBox.Text = CStr((2 + 3) < 6)`

  displays True in the text box.
A variable declared with a statement of the form

```vbnet
Dim var As Boolean
```

Has Boolean data type. It can assume just the two values True and False.

Example:  

```vbnet
Dim boolVar As Boolean
boolVar = 2 < 6
textBox.Text = CStr(boolVar)
```

displays True in the text box.
Syntax Error

The following is NOT a valid way to test whether $n$ falls between 2 and 5:

$$2 < n < 5$$
Correction to Syntax Error

To test if \( n \) falls between 2 and 5 use:

\[(2 < n) \text{ And } (n < 5)\]

A complete relational expression must be on either side of the logical operators And and Or.
Common Error in Boolean Expressions

• A common error is to replace the condition Not ( 2 < 3 ) with the condition ( 2 > 3 ).

• The correct replacement is ( 2 >= 3 ) because >= is the opposite of <, just as <= is the opposite of >.
Two Boolean-Valued Methods

• The expression `strVar1.EndsWith(strVar2)` is true if the value of the first variable ends with the value of the second variable.

• The expression `strVar1.StartsWith(strVar2)` is true if the value of the first variable begins with the value of the second variable.

• **Note:** String literals can be used instead of string variables.
Examples

After the following code is executed each text box will contain the word True.

```
Dim firstName As String = "William"
TextBox1.Text = CStr(firstName.EndsWith("am"))
TextBox2.Text = CStr(firstName.StartsWith("Will"))
```
A Boolean-Valued Function

• The expression `IsNumeric(strVar)` is true if the value of `strVar` can be converted to a number with `CInt` or `CDbl`.

  **Note:** The string variable can be replaced with a string literal.

• Examples:
  
  - `IsNumeric("123")` is true
  - `IsNumeric("$123")` is true
  - `IsNumeric("3 - 2")` is false
4.2 If Blocks

- If Block
- Nested If Blocks
- ElseIf Clauses
- Input Validation with If Blocks
If Block

The following program will take a course of action based on whether a condition is true.

If condition Then
  action 1
Else
  action 2
End If

Will be executed if condition is true
Will be executed if condition is false
Another example If Block

\[ \text{If condition Then} \]
\[ \quad \text{action 1} \]
\[ \text{End If} \]
\[ \text{Statement 2} \]
\[ \text{Statement 3} \]

Regardless of whether the condition in the If statement is true or false, these statements will be executed.
Pseudocode and Flowchart for an If Block

If condition is true Then
    Execute action 1
Else
    Execute action 2
End If

Is the condition true?

No

Execute action 2

Yes

Execute action 1
Example 1: Form

- `txtFirstNum`
- `txtSecondNum`
- `txtResult`
Example 1: Code

Private Sub btnFindLarger_Click(...)  _
Handles btnFindLarger.Click

Dim num1, num2, largerNum As Double
num1 = CDb1(txtFirstNum.Text)
num2 = CDb1(txtSecondNum.Text)
If num1 > num2 Then
    largerNum = num1
Else
    largerNum = num2
End If
txtResult.Text = "Larger number: " & largerNum
End Sub
Example 1: Output

![Maximum](image)

First number: 3
Second number: 7

Find Larger Number

The larger number is 7.
Example 2: Form

How many gallons does a ten-gallon hat hold?

Evaluate Answer

txtAnswer

txtSolution
Example 2: Code

Private Sub btnEvaluate_Click(...) _
Handles btnEvaluate.Click

    Dim answer As Double
    answer = CDbI(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 gals."
End Sub
Example 2: Output

![Quiz window]

**Quiz**

How many gallons does a ten-gallon hat hold? 10

Evaluate Answer

No, it holds about 3/4 of a gallon.
Example 3: Form

Quotation

Do you know what the game of skittles is (Y/N)?

Display Quotation

- mtbAnswer
- txtQuote
Private Sub btnDisplay_Click(...) _
    Handles btnDisplay.Click

    Dim msg As String
    msg = "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
        MessageBox.Show(msg, "")
    End If

    txtQuote.Text = "Life ain't all beer " & 
    and skittles. – Du Maurier (1894)."
End Sub
Example 3: Output

Image of a computer window showing a question: "Do you know what the game of skittles is (Y/N)?" with a response of "N".

Image of another computer window showing a definition: "Skittles is an old form of bowling in which a wooden disk is used to knock down nine pins arranged in a square."
Example 3: Output (continued)

Do you know what the game of skittles is (Y/N)? N

Life ain't all beer and skittles. - Du Maurier (1894)
Nested If Blocks

When one If block is contained inside another If block, the structure is referred to as nested If blocks.
Example 5: Form

![Profit/Loss Form Image]
Example 5: Partial Code

If costs = revenue Then
  txtResult.Text = "Break even"
Else
  If costs < revenue Then
    profit = revenue - costs
    txtResult.Text = "Profit is " & profit.ToString("C") & "."
  Else
    loss = costs - revenue
    txtResult.Text = "Loss is " & loss.ToString("C") & "."
  End If
End If
Example 5: Output

![Profit/Loss Window]

- Costs: 9500
- Revenue: 8000

Show Financial Status

Loss is $1,500.00.
ElseIf Clause

If condition 1 Then
   action 1
ElseIf condition 2 Then
   action 2
ElseIf condition 3 Then
   action 3
Else
   action 4
End If
Example 6: Form

- txtFirstNum
- txtSecondNum
- txtResult
Example 6: Code

Private Sub btnFindLarger_Click(...) _
    Handles btnFindLarger.Click
    Dim num1, num2 As Double
    num1 = CDbl(txtFirstNum.Text)
    num2 = CDbl(txtSecondNum.Text)
    If (num1 > num2) Then
        txtResult.Text = "Larger number is " & num1
    ElseIf (num2 > num1) Then
        txtResult.Text = "Larger number is " & num2
    Else
        txtResult.Text = "The two are equal."
    End If
End Sub
Example 7: Form

![FICA Tax Form Image]
Const WAGE_BASE As Double = 113700 'year 2013
Const SOCIAL_SECURITY_RATE As Double = 0.062
Const MEDICARE_RATE As Double = 0.0145
Dim ytdEarnings, curEarnings As Double
Dim socSecBenTax, medicareTax, ficaTaxes As Double
ytdEarnings = CDb1(txtToDate.Text)
curEarnings = CDb1(txtCurrent.Text)
If (ytdEarnings + curEarnings) <= WAGE_BASE Then
  socSecBenTax = SOCIAL_SECURITY_RATE * curEarnings
ElseIf ytdEarnings < WAGE_BASE Then
  socSecBenTax = SOCIAL_SECURITY_RATE * (WAGE_BASE - ytdEarnings)
End If
medicareTax = MEDICARE_RATE * curEarnings
ficaTaxes = socSecBenTax + medicareTax
txtText.Text = ficaTaxes.ToString("C")
Example 7: Output

FICA Tax

Total earnings for this year prior to the current pay period: $12345.67

Earnings for the current pay period: $543.21

Calculate FICA Tax

FICA tax for the current pay period: $41.56
Input Validation

The statement

\[
\text{If } (\text{IsNumeric(txtBox.Text)} = \text{True}) \text{ Then}
\]

is commonly used to validate that input is numeric. It can be condensed to

\[
\text{If } \text{IsNumeric(txtBox.Text)} \text{ Then}
\]
Simplified Nested If Statement

Care should be taken to make If blocks easy to understand.

Confusing

If \( \text{cond1} \) Then
Then
If \( \text{cond2} \) Then
action
End If
End If

Clear

If \( \text{cond1} \) And \( \text{cond2} \) action
End If
Some programs call for selecting among many possibilities. Although such tasks can be accomplished with complicated nested If blocks, the Select Case block (discussed in Section 4.3) is often a better alternative.
4.3 Select Case Block

• A decision-making structure that simplifies choosing among several actions.
• Avoids complex nested If constructs.
• If blocks make decisions based on the truth value of a condition. Select Case choices are determined by the value of an expression called a selector.
Select Case Terminology

Each of the possible actions is preceded by a clause of the form

```
Case valueList
```

where `valueList` itemizes the values of the `selector` for which the action should be taken.
Example 1: Form

![DIagram of a form](image)

- **txtPosition**: Input field for the finishing position.
- **txtOutcome**: Output field for the evaluation result.
Example 1: Code

Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click
    Dim position As Integer = CInt(txtPosition.Text)
    Select Case position
        Case 1
            txtOutcome.Text = "Gold medalist"
        Case 2
            txtOutcome.Text = "Silver medalist"
        Case 3
            txtOutcome.Text = "Bronze medalist"
        Case 4, 5
            txtOutcome.Text = "You almost won a medal."
        Case Else
            txtBox.Text = "Nice try."
    End Select
End Sub
Example 1: Output

![Image of Olympics output showing finishing position 2 and silver medalist]
Example 2: Form

Olympics

Finishing position (1, 2, 3, ...):

Evaluate Position

- txtPosition
- txtOutcome
Example 2: Code

Private Sub btnEvaluate_Click(...) _
    Handles btnEvaluate.Click

    Dim position As Integer = CInt(txtPosition.Text)

    Select Case position
        Case 1 To 3
            txtOutcome.Text = "Olympic medalist"
        Case Is >= 4
            txtOutcome.Text = "No medal this time."
    End Select

End Sub
Example 2: Output
Select Case Syntax

The general form of the Select Case block is

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

Each value list contains one or more of the following types of items separated by commas.

1. a literal
2. a variable
3. an expression
4. an inequality sign preceded by Is and followed by a literal, variable, or expression
5. a range given in the form $a \text{ To } b$, where $a$ and $b$ are literals, variables, or expressions.
Flowchart for Select Case
Example 4: Form

Quiz

What was President Wilson's first name?

Interpret Answer

txtReply
Example 4: Partial Code

```vbnet
Select Case firstName
    Case "THOMAS"
        txtReply.Text = "Correct."
    Case "WOODROW"
        txtReply.Text = "Sorry, his name" & 
                        "  was Thomas Woodrow Wilson."
    Case "PRESIDENT"
        txtReply.Text = "Are you for real?"
    Case Else
        txtReply.Text = "Nice try."
End Select
```
Example 4: Output

![Quiz interface](image)

What was President Wilson's first name? [Woodrow]

Interpret Answer

Sorry, his full name was Thomas Woodrow Wilson.
Comments

- In a Case clause of the form Case $b$ To $c$, the value of $b$ should be less than or equal to the value of $c$.
- The word “Is” should precede an inequality sign in a value list.
- If the word “Is” is accidentally omitted where required, the editor will automatically insert it when checking the line.
Data Type Comment

- The items in the value list must evaluate to a literal of the same data type as the selector.
- For instance, if the selector evaluated to a string value, as in

```vbnet
Dim firstName As String = txtBox.Text
Select Case firstName
```

then the clause

```vbnet
Case firstName.Length
```

would be meaningless.
Block-Level Scope

• A variable declared inside an If or Select Case block has **block-level scope**.

• The variable cannot be referred to outside of the block.
4.4 Input via User Selection

- Using a List Box for Input
- Group Box Control
- Using Radio Buttons for Input
- Using Check Boxes for Input
- Events Raised by Selections
The Three Types of Controls Used for Selection

- list box
- radio buttons
- check boxes
Fill a List Box at Design Time via its String Collection Editor

Tasks button

click here to invoke string collection editor
String Collection Editor

Fill by direct typing or by copying and pasting from a text editor or a spreadsheet.
List Box at Run Time

The value of lstMonths.Text is the string consisting of the selected item.
Example 1: Code for btnDetermine.Click

```vbnet
Dim daysInMonth As String
Select Case lstMonths.Text
    Case "September", "April", "June", "November"
        daysInMonth = "30"
    Case "February"
        daysInMonth = "28 or 29"
    Case Else
        daysInMonth = "31"
End Select
txtDays.Text = daysInMonth
```
The Group Box Control

- Group boxes are passive objects used to group other objects together.
- When you drag a group box, the attached controls follow as a unit.
- To attach controls to a group box, create the group box and then place or drag the controls into the group box.
Group Box Example

Three attached controls:
Button1
Button2
Button3
Radio Button Properties

• To determine if the button is on or off

    radioButton.Checked

    has value True if button is on.

• To turn a radio button on

    radioButton.Checked = True

Note: At most one radio button in a group box can be checked at one time.
Example 3: Form

![Form Diagram]

- radChild
- radMinor
- radAdult
- radSenior
Example 3: Code for Button

If radChild.Checked Then
    txtFee.Text = (0).ToString("C")
ElseIf radMinor.Checked Then
    txtFee.Text = (5).ToString("C")
ElseIf radAdult.Checked Then
    txtFee.Text = (10).ToString("C")
ElseIf radSenior.Checked Then
    txtFee.Text = (7.5).ToString("C")
Else
    MessageBox.Show("Must make a selection.")
End If
Example 3: Output
The Check Box Control

- Consists of a small square and a caption
- Presents the user with a Yes/No choice
- During run time, clicking on the check box toggles the appearance of a check mark.
- Checked property has value True when check box is checked and False when not
- CheckedChanged event is raised when the user clicks on the check box
- Several check boxes in a group can be checked at the same time.
Example 4: Form

chkDrug
chkDental
chkVision
chkMedical
Example 4: Code for Button

```vbnet
Dim sum As Double = 0
If chkDrugs.Checked Then
    sum += 39.15
End If
If chkDental.Checked Then
    sum += 10.81
End If
If chkVision.Checked Then
    sum += 2.25
End If
If chkMedical.Checked Then
    sum += 55.52
End If
txtTotal.Text = sum.ToString("C")
```
Example 4: Output

![Benefits Menu](image)

- Prescription Drug Plan ($39.15)
- Vision Plan ($2.25)
- Medical Plan ($55.52)

Determine Total Monthly Cost

Total monthly cost: $96.92
Events Raised by a Selection

- **SelectedIndexChanged** – raised when a new item of a list box is selected
- **CheckedChanged** - raised when the user clicks on an unchecked radio button or a check box; that is, when the value of the Checked property is changed.
Example 5: Code

Private Sub checkBox_Changed(...) Handles _
  chkDrugs.CheckedChanged,
  chkDental.CheckedChanged,
  chkVision.CheckedChanged,
  chkMedical.CheckChanged

  Dim sum As Double = 0
  If chkDrugs.Checked Then
    sum += 39.15
  End If

(continued on next slide)
Example 5: Code (continued)

If chkDental.Checked Then
    sum += 10.81
End If
If chkVision.Checked Then
    sum += 2.25
End If
If chkMedical.Checked Then
    sum += 55.52
End If

txtTotal.Text = sum.ToString("C")
End Sub
Example 5: Output

![Benefits Menu]

- Prescription Drug Plan ($39.15)
- Vision Plan ($2.25)
- Medical Plan ($55.52)

Determine Total Monthly Cost

Total monthly cost: $96.92