PREV NEXT

[Page 193]

Chapter 5. Decisions

(This item omitted from WebBook edition)

5.1 Relational and Logical Operators 194

- ANSI Values
- Logical Operators
- Boolean Data Type

5.2 If Blocks 201

- If Block
- ElseIf Clauses

5.3 Select Case Blocks 218

5.4 A Case Study: Weekly Payroll 235

- Designing the Weekly Payroll Program
- Pseudocode for the Display Payroll Event
- Writing the Weekly Payroll Program
- The Program and the User Interface

Summary 243

Programming Projects 243

PREV PREV NEXT NEXT

[Page 194]

5.1. Relational and Logical Operators

In Chapter 2, we discussed the two logical programming constructs decision and loop. In order to make a decision or control a loop, you need to specify a condition that determines the course of action.

A condition is an expression involving relational operators (such as <and=) that is either true or false. Conditions also may incorporate logical operators (such as And, Or, and Not). ANSI values determine the order used to compare strings with the relational operators. Boolean variables and literals can assume

the values True or False.

ANSI Values

Each of the 47 different keys in the center typewriter portion of the keyboard can produce two characters, for a total of 94 characters. Adding 1 for the space character produced by the space bar makes 95 characters. These characters have numbers ranging from 32 to 126 associated with them. These values, called the ANSI (or ASCII) values of the characters, are given in Appendix A. Table 5.1 shows a few of the values.

Table 5.1. A few ANSI values.

32 (space)	48 0	66 B	122 z
33 !	49 1	90 Z	123 {
34 "	57 9	97 a	125 }
35 #	65 A	98 b	126 ~

Most of the best known fonts, such as Courier New, Microsoft San Serif, and Times New Roman, are essentially governed by the ANSI standard, which assigns characters to the numbers from 0 to 255. Table 5.2 shows a few of the higher ANSI values.

Table 5.2. A few higher ANSI values.

162 ¢	177 ±	181 µ	190 ¾
169 ©	178 ^{2}	188 ¼	247 ÷
176 °	179 ^{3}	189 ½	248 ϕ

If n is a number between 0 and 255, then

```
Chr(n)
```

is the string consisting of the character with ANSI value η. If str is any string, then

```
Asc(str)
```

is the ANSI value of the first character of str. For instance, the statement

```
txtBox.Text = Chr(65)
```

displays the letter A in the text box, and the statement

```
lstBox.Items.Add(Asc("Apple"))
```

displays the number 65 in the list box.

Concatenation can be used with Chr to obtain strings using the higher ANSI characters. For instance, with one of the fonts that conforms to the ANSI standard, the statement

---

[Page 195]

```
txtBox.Text = "32" & Chr(176) & " Fahrenheit"
```

displays 32° Fahrenheit in the text box.

The quotation-mark character (") can be placed into a string by using Chr(34). For example, after the statement

```
txtBox.Text = "George " & Chr(34) & "Babe" & Chr(34) & " Ruth"
```

is executed, the text box contains

```
George "Babe" Ruth
```

The relational operator less than(<) can be applied to both numbers and strings. The number a is said to be less than the number b if a lies to the left of b on the number line. For instance, 2 < 5, -5 < -2, and 0 < 3.5.

The string a is said to be less than the string b if a precedes b alphabetically when using the ANSI table to alphabetize their values. For instance, "cat" < "dog", "cart" < "cat", and "cat" < "catalog". Digits precede uppercase letters, which precede lowercase letters. Two strings are compared working from left to right, character by character, to determine which one should precede the other. Therefore, "9W" < "bat", "Dog" < "cat", and "Sales-99" < "Sales-retail".

Table 5.3 shows the different mathematical relational operators, their representations in Visual Basic, and their meanings.

Table 5.3. Relational operators.

Mathematical Notation	Visual Basic Notation	Numeric Meaning	String Meaning
=	=	equal to	identical to
	<>	not equal 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

Example 1.

(This item is displayed on pages 195 - 196 in the print version)

Determine whether each of the following conditions is true or false.

a. 1 <= 1

b. 1 < 1

c. "car" < "cat"

d. "Dog" < "dog"

Solution

a. True. The notation <= means "less than or equal to." That is, the condition is true provided either of the two circumstances holds. The second one (equal to) holds.

b. False. The notation < means "strictly less than" and no number can be strictly less than itself.

[Page 196]

c. True. The characters of the strings are compared one at a time working from left to right. Because the first two match, the third character decides the order.

d. True. Because uppercase letters precede lowercase letters in the ANSI table, the first character of "Dog" precedes the first character of "dog".

Conditions also can involve variables, numeric operators, and functions. To determine whether a condition is true or false, first compute the numeric or string values and then decide if the resulting assertion is true or false.

Example 2.

Suppose the numeric variables a and b have values 4 and 3, and the string variables c and d have values "hello" and "bye". Are the following conditions true or false?

a. (a + b) < 2 * a

b. (c.Length - b) = (a/2)

c. c < ("good" & d)

Solution

a. The value of a+b is 7 and the value of 2 * a is 8. Because 7 < 8, the condition is true.

b. True, because the value of c.Length - b is 2, the same as (a / 2).

c. The condition "hello" < "goodbye" is false, because "h" follows "g" in the ANSI table.

Logical Operators

Programming situations often require more complex conditions than those considered so far. For instance, suppose we would like to state that the value of a numeric variable, n, is strictly between 2 and 5. The proper Visual Basic condition is

(2 < n) And (n < 5)

The condition (2 < n) And (n < 5) is a combination of the two conditions 2 < n and n < 5 with the logical operator And.

The three main logical operators are And, Or, and Not. If cond1 and cond2 are conditions, then the condition

```
cond1 And cond2
```

is true if both cond1 and cond2 are true. Otherwise, it is false. The condition

```
cond1 Or cond2
```

is true if either cond1 or cond2 (or both) is true. Otherwise, it is false. The condition

```
Not cond1
```

is true if cond1 is false, and is false if cond1 is true.

[Page 197]

Example 3.

Suppose the numeric variable n has value 4 and the string variable answ has value "Y". Determine whether each of the following conditions is true or false.

a. (2 < n) And (n < 6)

b. (2 < n) Or (n = 6)

c. Not (n < 6)

d. (answ = "Y") Or (answ = "y")

e. (answ = "Y") And (answ = "y")

f. Not (answ = "y")

g. ((2 < n) And (n = 5 + 1)) Or (answ = "No")

h. ((n = 2) And (n = 7)) Or (answ = "Y")

i. (n = 2) And ((n = 7) Or (answ = "Y"))

Solution

a. True, because the conditions (2 < 4) and (4 < 6) are both true.

b. True, because the condition (2 < 4) is true. The fact that the condition (4 = 6) is false does not affect the conclusion. The only requirement is that at least one of the two conditions be true.

c. False, because (4 < 6) is true.

d. True, because the first condition becomes ("Y" = "Y") when the value of answ is substituted for answ.

e. False, because the second condition is false. Actually, this compound condition is false for every value of answ.

f. True, because ("Y" = "y") is false.

g. False. In this logical expression, the compound condition ((2 < n) And (n = 5 + 1)) and the simple condition (answ = "No") are joined by the logical operator Or. Because both of these conditions are false, the total condition is false.

h. True, because the second Or clause is true.

i. False. Comparing (h) and (i) shows the necessity of using parentheses to specify the intended grouping.

Boolean Data Type

A statement of the form

```
txtBox.Text = condition
```

will display either True or False in the text box, depending on the truth value of the condition. Any variable or expression that evaluates to either True or False is said to have a Boolean data type. The following lines of code display False in the text box.

[Page 198]

```
Dim x As Integer = 5
txtBox.Text = (3 + x) < 7
```

A variable is declared to be of type Boolean with a statement of the form

```
Dim varName As Boolean
```

The following lines of code will display True in the text box.

```
Dim boolVar as Boolean
Dim x As Integer = 2
Dim y As Integer = 3
boolVar = x < y
txtBox.Text = boolVar
```

The answer to part (i) of Example 3 can be confirmed to be "false" by executing the following lines of code.

```
Dim n as Integer = 4
Dim answ as String = "Y"
txtBox.Text = (n = 2) And ((n = 7) Or (answ = "Y"))
```

Comments

1. A condition involving numeric variables is different from an algebraic truth. The assertion (a + b) < 2*a, considered in Example 2, is not a valid algebraic truth because it isn't true for all values of a and b. When encountered in a Visual Basic program, however, it will be considered true if it is correct for the current values of the variables.

2. Conditions evaluate to either true or false. These two values often are called the possible truth values of the condition.

3. A condition such as 2 < n < 5 should never be used, because Visual Basic will not evaluate it as intended. The correct condition is (2 < n) And (n < 5).

4. A common error is to replace the condition Not (n < m) by the condition (n > m). The correct

replacement is (n >= m).

Practice Problems 5.1

1. Is the condition "Hello" = "Hello" true or false?
2. Complete Table 5.4.

Table 5.4. Truth values of logical operators.

cond1	cond2	cond1 And cond2	cond1 Or cond2	Not cond2
True	True	True		
True	False		True	
False	True			False
False	False			

[Page 199]

Exercises 5.1

In Exercises 1 through 6, determine the output displayed in the text box.

1. `txtBox.Text = Chr(104) & Chr(105)`
2. `txtBox.Text = "C" & Chr(35)`
3. `txtBox.Text = "The letter before G is" & Chr(Asc("G") - 1)`
4. `txtBox.Text = Chr(Asc("B")) 'The ANSI value of B is 66`
5.
```
Dim quote, person, qMark As String
quote = "We're all in this alone."
person = "Lily Tomlin"
qMark = Chr(34)
txtBox.Text = qMark & quote & qMark & "_" & person
```

6.
```
Dim letter As String
letter = "D"
txtBox.Text = letter & " is the " & (Asc(letter) - Asc("A") + 1) & _
              "th letter of the alphabet."
```

In Exercises 7 through 18, determine whether the condition is true or false. Assume a = 2 and b = 3.

7. 3 * a = 2 * b
8. (5 - a) * b < 7
9. b <= 3
10. a^b = b^a
11. a^(5 - 2) > 7
12. 3E-02 <.01 * a
13. (a < b) or (b < a)
14. (a * a < b) Or Not (a * a < a)
15. Not ((a < b) And (a < (b + a)))
16. Not (a < b) Or Not (a < (b + a))
17. ((a = b) And (a * a < b * b)) Or ((b < a) And (2 * a < b))
18. ((a = b) Or Not (b < a)) And ((a < b) Or (b = a + 1))

In Exercises 19 through 30, determine whether the condition is true or false.

19. "9W" < > "9w"
20. "Inspector" < "gadget"
21. "Car" < "Train"
22. "J" > = "J"
23. "99" > "ninety-nine"
24. "B" > "?"
25. ("Duck" < "pig") And ("pig" < "big")

---

26. "Duck" < "Duck" & "Duck"

27. Not (("B" = "b") Or ("Big" < "big"))

28. Not ("B" = "b") And Not ("Big" < "big")

29. (("Ant" < "hill") And ("mole" > "hill")) Or Not (Not ("Ant" < "hill") Or Not ("Mole" > "hill"))

30. (7 < 34) And ("7" > "34")

In Exercises 31 through 40, determine whether or not the two conditions are equivalentthat is, whether they will be true or false for exactly the same values of the variables appearing in them.

31. a <= b; (a < b) Or (a = b)

32. Not (a < b); a > b

33. (a = b) And (a < b); a <> b

34. Not ((a = b) Or (a = c)); (a <> b) And (a <> c)

35. (a < b) And ((a > d) Or (a > e)); ((a < b) And (a > d)) Or ((a < b) And (a > e))

36. Not ((a = b + c) Or (a = b)); (a <> b) Or (a <> b + c)

37. (a < b + c) Or (a = b + c); Not ((a > b) Or (a > c))

38. Not (a >= b); (a <= b) Or Not (a = b)

39. Not (a >= b); (a <= b) And Not (a = b)

40. (a = b) And ((b = c) Or (a = c)); (a = b) Or ((b = c) And (a = c))

In Exercises 41 through 45, write a condition equivalent to the negation of the given condition. (For example, a <> b is equivalent to the negation of a = b.)

41. a > b

42. (a = b) Or (a = d)

43. (a < b) And (c <> d)

44. Not ((a = b) Or (a > b))

45. (a <> "") And (a < b) And (a.Length < 5)

46. Rework Exercise 20 by evaluating a Boolean expression.

47. Rework Exercise 21 by evaluating a Boolean expression.

48. Rework Exercise 22 by evaluating a Boolean expression.

49. Rework Exercise 23 by evaluating a Boolean expression.

Solutions to Practice Problems 5.1

1. False. The first string has six characters, whereas the second has five. Two strings must be 100% identical to be called equal.

2.

cond1	cond2	cond1 And cond2	cond1 Or cond2	Not cond2
True	True	True	True	False
True	False	False	True	True
False	True	False	True	False
False	False	False	False	True

PREV PREV NEXT NEXT

[Page 201]

5.2. If Blocks

An If block allows a program to decide on a course of action based on whether certain conditions are true or false.

If Block

A block of the form:

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

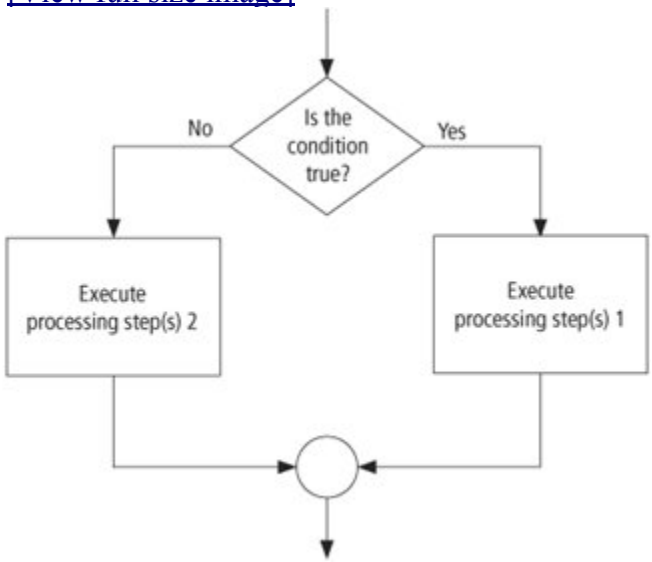
causes the program to take action1 if condition is true and action2 if condition is false. Each action

consists of one or more Visual Basic statements. After an action is taken, execution continues with the line after the If block. Figure 5.1 contains the pseudocode and flowchart for an If block.

Figure 5.1. Pseudocode and flowchart for an If block.

[View full size image]

```
If condition is true Then
   Processing step(s)1
Else
   Processing step(s)2
End If
```



Example 1.

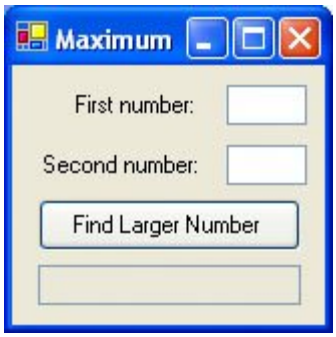
(This item is displayed on pages 201 - 202 in the print version)

The following program finds the larger of two numbers input by the user. The condition is

```
num1 > num2
```

and each action consists of a single assignment statement. With the inputs 3 and 7, the condition is false, and so the second action is taken.

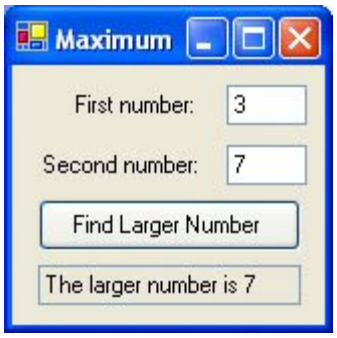
[Page 202]



Object	Property	Setting
frmMaximum	Text	Maximum
lblFirstNum	Text	First Number:
txtFirstNum		
lblSecondNum	Text	Second Number:
txtSecondNum		
btnFindLarger	Text	Find Larger Number
txtResult	ReadOnly	True

```
Private Sub btnFindLarger_Click(...) Handles btnFindLarger.Click
  Dim num1, num2, largerNum As Double
  num1 = CDbl (txtFirstNum.Text)
  num2 = CDbl (txtSecondNum.Text)
  If num1 > num2 Then
    largerNum = num1
  Else
    largerNum = num2
  EndIf
  txtResult.Text = "The larger number is " & largerNum
End Sub
```

[Run, type 3 and 7 into the text boxes, and press the button.]



Example 2.

(This item is displayed on pages 202 - 203 in the print version)

The following program requests the costs and revenue for a company and displays the message "Break even" if the costs and revenue are equal; otherwise, it displays the profit or loss. The action following Else is another If block.

Object	Property	Setting
frmStatus	Text	Profit/Loss
lblCosts	Text	Costs:
txtCosts		
lblRev	Text	Revenue:
txtRev		
btnShow	Text	Show Financial Status
txtResult	ReadOnly	True

[Page 203]

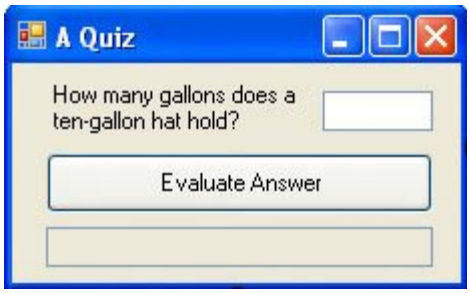
```
Private Sub btnShow_Click(...) Handles btnShow.Click
  Dim costs, revenue, profit, loss As Double
  costs = CDbl (txtCosts.Text)
  revenue = CDbl (txtRev.Text)
  If costs = revenue Then
    txtResult.Text = "Break even"
  Else
    If costs < revenue Then
      profit = revenue - costs
      txtResult.Text = "Profit is " & FormatCurrency(profit)
    Else
      loss = costs - revenue
      txtResult.Text = "Loss is " & FormatCurrency(loss)
    End If
  End If
End Sub
```

[Run, type 9500 and 8000 into the text boxes, and press the button.]

Example 3.

(This item is displayed on pages 203 - 204 in the print version)

The If block in the following program has a logical operator in its condition.



Object	Property	Setting
frmQuiz	Text	A Quiz
lblQuestion	Text	How many gallons does a ten-gallon hat hold?
txtAnswer		
btnEvaluate	Text	Evaluate Answer
txtSolution	ReadOnly	True

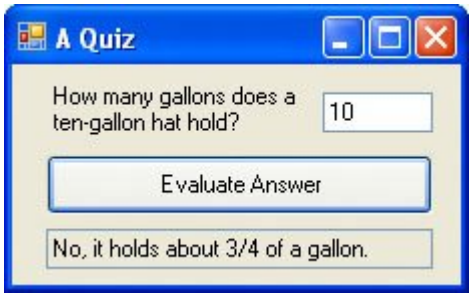
```
Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click
  'Evaluate answer
  Dim answer As Double
  answer = CDbl (txtAnswer.Text)
  If (answer >= 0.5) And (answer <= 1) Then
    txtSolution.Text = "Good, "
  Else
```

[Page 204]

```
    txtSolution.Text = "No, "
  End If
  txtSolution.Text &= "it holds about 3/4 of a gallon."
```

```
End Sub
```

[Run, type 10 into the text box, and press the button.]

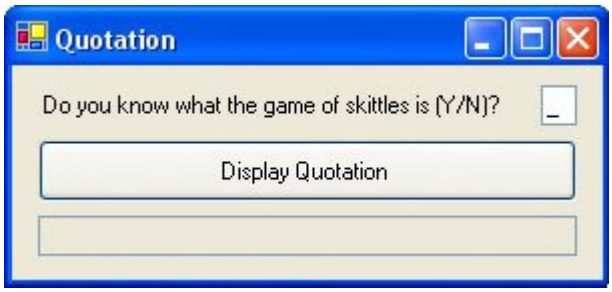


The Else part of an If block can be omitted. This important type of If block appears in the next example.

Example 4.

(This item is displayed on pages 204 - 205 in the print version)

The following program offers assistance to the user before presenting a quotation.



Object	Property	Setting
frmQuotation	Text	Quotation
lblQuestion	Text	Do you know what the game of skittles is (Y/N)?
mtxtAnswer	Mask	L
btnDisplay	Text	Display Quotation
txtQuote	ReadOnly	True

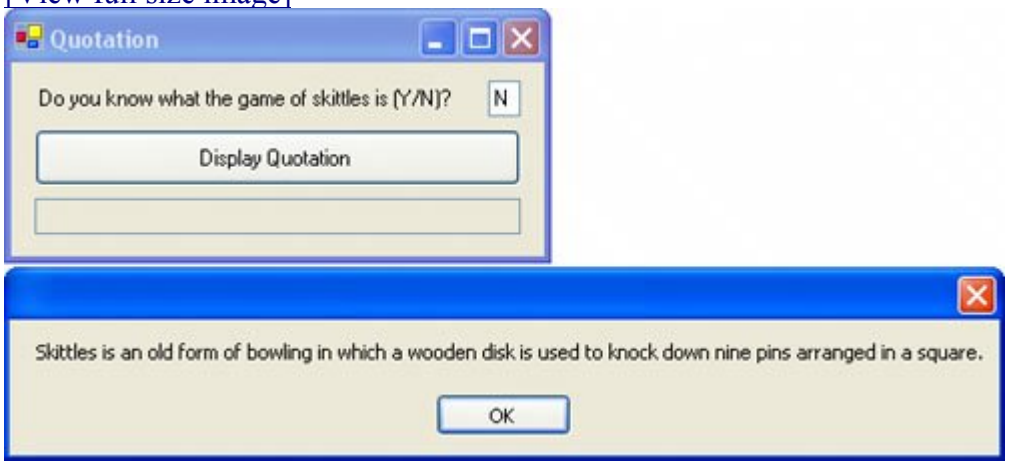
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim message As String
  message = "Skittles is an old form of bowling in which a wooden " & _
            "disk is used to knock down nine pins arranged in a square."
  If mtxtAnswer.Text.ToUpper = "N" Then
```

```
    MsgBox(message, 0, "")
  End If
  txtQuote.Text =  "Life ain't all beer and skittles." & _
                  " - Du Maurier (1894)"
End Sub
```

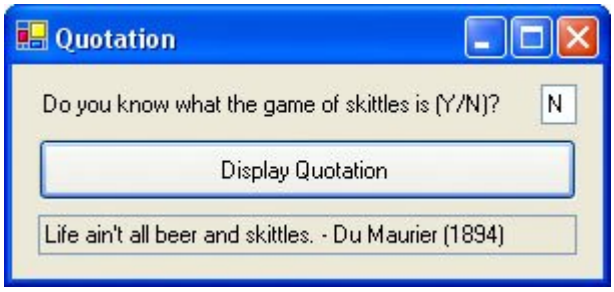
[Run, type "N" into the masked text box, and press the button.]

[Page 205]

[View full size image]



[Press OK.]



Note: Rerun the program, type "Y" into the masked text box, press the button, and observe that the description of the game is skipped.

ElseIf Clauses

An extension of the If block allows for more than two possible alternatives with the inclusion of ElseIf clauses. A typical block of this type is

```
If condition1 Then
  action1
ElseIf condition2 Then
  action2
ElseIf condition3 Then
  action3
```

```
Else
  action4
End If
```

This block searches for the first true condition, carries out its action, and then skips to the statement following End If. If none of the conditions is true, then Else's action is carried out. Execution then continues with the statement following the block. In general, an If block can contain any number of ElseIf clauses. As before, the Else clause is optional.

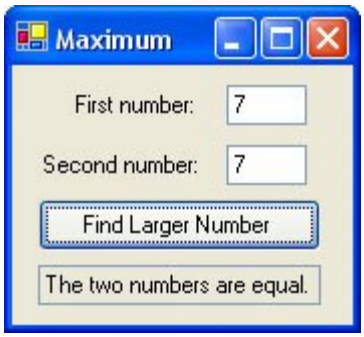
[Page 206]

Example 5.

The following program redoes Example 1 so that if the two numbers are equal, the program so reports:

```
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 = "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
```

[Run, type 7 into both text boxes, and press the button.]

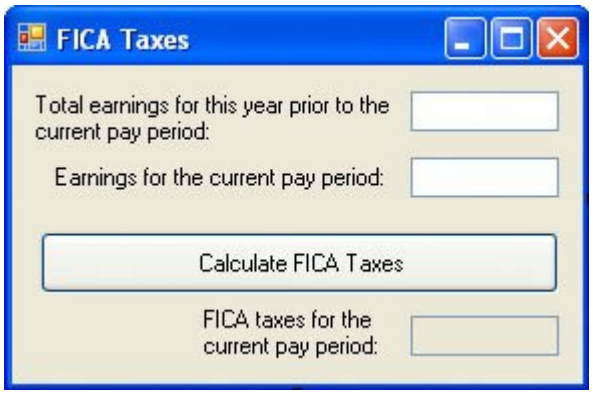


If blocks allow us to define functions whose values are not determined by a simple formula. The function in Example 6 uses an If block.

Example 6.

(This item is displayed on pages 206 - 207 in the print version)

The Social Security or FICA tax has two componentsthe Social Security benefits tax, which in 2005 is 6.2 percent on the first $90,000 of earnings for the year, and the Medicare tax, which is 1.45 percent of earnings. The following program calculates an employee's FICA tax for the current pay period.



Object	Property	Setting
frm FICA	Text	FICA Taxes
lblToDate	Text	Total earnings for this year prior to the current pay period:
txtToDate		
lblCurrent	Text	Earnings for the current pay period:
txtCurrent		
btnCalculate	Text	Calculate FICA Taxes
lblTax	Text	FICA taxes for the current pay period:
txtTax	ReadOnly	True

[Page 207]

```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim ficaTaxes As Double
  ficaTaxes = CalcFICA(CDbl (txtToDate.Text), CDbl (txtCurrent.Text))
  txtTax.Text = FormatCurrency(ficaTaxes)
End Sub

Function CalcFICA(ByVal ytdEarnings As Double, _
                  ByVal curEarnings As Double) As Double
  'Calculate Social Security benefits tax and Medicare tax
```

```
  'for a single pay period in 2005
  Dim socialSecurityBenTax, medicareTax As Double
  If (ytdEarnings + curEarnings) <= 90000 Then
    socialSecurityBenTax = 0.062 * curEarnings
  ElseIf ytdEarnings < 90000 Then
    socialSecurityBenTax = 0.062 * (90000 - ytdEarnings)
  End If
  medicareTax = 0.0145 * curEarnings
  Return socialSecurityBenTax + medicareTax
End Function
```

[Run, type 12345.67 and 543.21 into the top two text boxes, and press the button. The following is displayed in txtTax.]

```
$41.56
```

Comments

1. Constructs in which an If block is contained inside another If block are referred to as nested If blocks.

2. Care should be taken to make If blocks easy to understand. For instance, in Figure 5.2, the block on the left is difficult to follow and should be replaced by the clearer block on the right.

Figure 5.2. A confusing If block and an improvement.

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

3. In Appendix C, the section "Stepping through Programs Containing Decision Structures: Chapter 5" uses the Visual Basic debugging tools to trace the flow through an If block.

4. Some programs call for selecting among many possibilities. Although such tasks can be accomplished with complicated If blocks, the Select Case block (discussed in the next section) is often a better alternative.

[Page 208]

Practice Problems 5.2

1. Suppose the user is asked to input a number into txtNumber for which the square root is to be taken. Fill in the If block so that the lines of code that follow will display either the

message "Number can't be negative." or the square root of the number.

```
Private Sub btnSqrt_Click(...) Handles btnSqrt.Click
  'Check reasonableness of data
  Dim num As Double
  num = CDbl(txtNumber.Text)
  If
  End If
End Sub
```

2. Improve the block

```
If a < b Then
  If c < 5 Then
    txtBox.Text = "hello"
  End If
End If
```

Exercises 5.2

In Exercises 1 through 12, determine the output displayed in the text box when the button is clicked.

1.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num As Double = 4
  If num <= 9 Then
    txtOutput.Text = "Less than ten."
  Else
    If num = 4 Then
      txtOutput.Text = "Equal to four."
    End If
  End If
End Sub
```

2.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim gpa As Double = 3.49
  txtOutput.Clear()
  If gpa >= 3.5 Then
    txtOutput.Text = "Honors "
  End If
 txtOutput.Text &= "Student"
End Sub
```

3.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim a As Double = 5
  txtOutput.Clear()
  If (3 * a - 4) < 9 Then
    txtOutput.Text = "Remember, "
  End If
  txtOutput.Text &= "tomorrow is another day."
End Sub
```

4.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim change As Double = 356    'Amount of change in cents
  If change >= 100 Then
    txtOutput.Text = "Your change contains "  & _
                     Int(change / 100) & " dollars."
  Else
    txtOutput.Text = "Your change contains no dollars."
  End If
End Sub
```

5.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim a as Double = 2
  Dim b As Double = 3
  Dim c As Double = 5
  If a * b < c Then
    b = 7
  Else
    b = c * a
  End If
  txtOutput.Text = CStr(b)
End Sub
```

6.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim a, b As Double
  a = CDbl(InputBox("Enter a number."))
  b = CDbl(InputBox("Enter another number."))
  If a > b Then
    a += 1
  Else
    b += 1
  End If
  txtOutput.Text = a & "  "& b
End Sub
```

(Assume the responses are 7 and 11.)

7.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Cost of phone call from New York to London
  Dim length As Double
  InputLength(length)
```

```
  DisplayCost(length)
End Sub
```

[Page 210]

```
Function Cost(ByVal length As Double) As Double
  If length < 1 Then
    Return .46
  Else
    Return .46 + (length - 1) * .36
  End If
End Function
Sub DisplayCost(ByVal length As Double)
  'Display the cost of a call
  txtBox.Text = "Cost of call: "& FormatCurrency(Cost(length))
End Sub

Sub InputLength(ByRef length As Double)
  'Request the length of a phone call
  length = CDbl(InputBox("Duration of the call in minutes?"))
End Sub
```

(Assume the response is 31.)

8.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim letter As String
  letter = InputBox("Enter A, B, or C.")
  If letter = "A" Then
    DisplayAmessage()
  ElseIf letter = "B" Then
    DisplayBmessage()
  ElseIf letter = "C" Then
    DisplayCmessage()
  Else
    txtOutput.Text = "Not a valid letter."
  End If
End Sub

Sub DisplayAmessage()
  txtOutput.Text = "A, my name is Alice."
End Sub

Sub DisplayBmessage()
  txtOutput.Text = "To be, or not to be."
End Sub

Sub DisplayCmessage()
  txtOutput.Text = "Oh, say, can you see."
End Sub
```

(Assume the response is B.)

9.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim vowels As Integer    'Number of vowels
```

```
  ExamineLetter(vowels)
```

```
  ExamineLetter(vowels)
  ExamineLetter(vowels)
  txtOutput.Text = "The number of vowels is "& vowels
End Sub

Sub ExamineLetter(ByRef vowels As Integer)
  Dim ltr As String
  ltr = InputBox("Enter a letter.")
  ltr = ltr.ToUpper
  If (ltr = "A") Or (ltr = "E") Or (ltr = "I") Or _
    (ltr = "O") Or (ltr = "U") Then
    vowels += 1
  End If
End Sub
```

(Assume the three responses are U, b, and a.)

10.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim a As Double = 5
  If (a > 2) And ((a = 3) Or (a < 7)) Then
    txtOutput.Text = "Hi"
  End If
End Sub
```

11.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num As Double = 5
  If num < 0 Then
    txtOutput.Text = "neg"
  Else
    If num = 0 Then
      txtOutput.Text = "zero"
    Else
      txtOutput.Text = "positive"
    End If
  End If
End Sub
```

12.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim msg As String, age As Integer
  msg = "You are eligible to vote"
  age = CInt(InputBox("Enter your age."))
  If age >= 18 Then
    txtOutput.Text = msg
  Else
    txtOutput.Text = msg & " in "& (18 - age) &  " years."
  End If
End Sub
```

(Assume the response is 16.)

[Page 212]

In Exercises 13 through 20, identify the errors.

13.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num As Double = 0.5
  If (1 < num < 3) Then
    txtOutput.Text = "Number is between 1 and 3."
  End If
End Sub
```

14.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num As Double = 6
  If num > 5 And < 9 Then
    txtOutput.Text = "Yes"
  Else
    txtOutput.Text = "No"
  End If
End Sub
```

15.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  If (2 <> 3)
    txtOutput.Text = "Numbers are not equal"
  End If
End Sub
```

16.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim major As String
  major = "Computer Science"
  If major = "Business" Or "Computer Science" Then
    txtOutput.Text = "Yes"
  End If
End Sub
```

17.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim numName As String, num As Double
  numName = "Seven"
  num = CDbl(InputBox("Enter a number."))
  If num < numName Then
    txtOutput.Text = "Less than"
  Else
    txtOutput.Text = "Greater than"
  End If
```

```
End Sub
```

18.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Change switch from "on" to "off", or from "off" to "on"
  Dim switch As String
  switch = InputBox("Enter On or Off.")
  If switch = "Off" Then
    switch = "On"
  End If
```

[Page 213]

```
  If switch = "On" Then
    switch = "Off"
  End If
  txtOutput.Text = switch
End Sub
```

19.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Display "OK" if either j or k equals 4
  Dim j As Double = 2
  Dim k As Double = 3
  If j Or k = 4 Then
    txtOutput.Text = "OK"
  End If
End Sub
```

20.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Is your program correct?
  Dim query, answer1, answer2 As String
  query = "Are you sure everything in your program is correct?"
  answer1 = InputBox(query)
  answer1 = answer1.ToUpper.Substring(0, 1)
  If answer1 = "N" Then
    txtOutput.Text = "Don't patch bad code, rewrite it."
  Else
    query = "Does your program run correctly?"
    answer2 = InputBox(query)
    answer2 = answer2.ToUpper.Substring(0, 1)
    If answer2 = "Y" Then
      txtOutput.Text = "Congratulations."
    Else
      txtOutput.Text = "Something you are sure about is wrong."
    End If
End Sub
```

In Exercises 21 through 26, simplify the code.

21.

```
If (a = 2) Then
  a = 3 + a
```

```
Else
  a = 5
End If
```

22.
```
If Not (answer <> "y") Then
  txtOutput.Text = "YES"
Else
  If (answer = "y") Or (answer = "Y") Then
  txtOutput.Text = "YES"
  End If
End If
```

[Page 214]

23.
```
If (j = 7) Then
  b = 1
Else
  If (j <> 7) Then
    b = 2
  End If
End If
```

24.
```
If (a < b) Then
  If (b < c) Then
    txtOutput.Text = b & " is between "& a & " and "& c
  End If
End If
```

25.
```
message = "Is Alaska bigger than Texas and California combined?"
answer = InputBox(message)
If (answer.Substring(0, 1) = "Y") Then
  answer = "YES"
End If
If (answer.Substring(0, 1) = "y") Then
  answer = "YES"
End If
If (answer = "YES") Then
  txtOutput.Text = "Correct"
Else
  txtOutput.Text = "Wrong"
End If
```

26.
```
message = "How tall (in feet) is the Statue of Liberty?"
feet = CDbl(InputBox(message))
```

```
If (feet <= 141) Then
  lstOutput.Items.Add("Nope")
End If
If (feet > 141) Then
  If (feet < 161) Then
    lstOutput.Items.Add("Close")
  Else
    lstOutput.Items.Add("Nope")
  End If
End If
sltOutput.Items.Add("The statue is 151 feet from base to torch.")
```

27. Write a program to determine how much to tip the server in a restaurant. The tip should be 15 percent of the check, with a minimum of $1.

28. A bagel shop charges 75 cents per bagel for orders of less than a half-dozen bagels and charges 60 cents per bagel for orders of a half-dozen or more bagels. Write a program that requests the number of bagels ordered and displays the total cost. (Test the program for orders of four bagels and a dozen bagels.)

[Page 215]

29. A computer store sells diskettes at 25 cents each for small orders or at 20 cents each for orders of 100 diskettes or more. Write a program that requests the number of diskettes ordered and displays the total cost. (Test the program for purchases of 5 and 200 diskettes.)

30. A copy center charges 5 cents per copy for the first 100 copies and 3 cents per copy for each additional copy. Write a program that requests the number of copies as input and displays the total cost. (Test the program with the quantities 25 and 125.)

31. Write a quiz program to ask "Who was the first Ronald McDonald?" The program should display "Correct." if the answer is "Willard Scott" and "Nice try" for any other answer.

32. Suppose a program has a button with the caption "Quit." Suppose also that the Name property of this button is btnQuit. Write a btnQuit_Click event procedure that gives the user a second chance before ending the program. The procedure should use an input box to request that the user confirm that the program should be terminated, and then end the program only if the user responds in the affirmative.

33. Write a program to handle a savings-account withdrawal. The program should request the current balance and the amount of the withdrawal as input and then display the new balance. If the withdrawal is greater than the original balance, the program should display "Withdrawal denied." If the new balance is less than $150, the message "Balance below $150" should be displayed.

34. Write a program that requests three scores as input and displays the average of the two highest scores. The input and output should be handled by Sub procedures, and the average should be determined by a user-defined function.

35. A grocery store sells apples for 79 cents per pound. Write a cashier's program that requests the number of pounds and the amount of cash tendered as input and displays the change from the transaction. If the cash is not enough, the message "I need $x.xx more." should be displayed, where $x.xx is the difference between the cash and the total cost. (Test the program with six pounds and $5, and four pounds and $3.)

36. Federal law requires hourly employees be paid "time-and-a-half" for work in excess of 40 hours in a week. For example, if a person's hourly wage is $8 and he works 60 hours in a week, his gross pay should be

(40 x 8) + (1.5 x 8 x (60 - 40)) = $560

Write a program that requests as input the number of hours a person works in a given week and his hourly wage, and then displays his gross pay.

37. Write a program that requests a word (with lowercase letters) as input and translates the word into pig latin. The rules for translating a word into pig latin are as follows:

- **a.** If the word begins with a consonant, move the first letter to the end of the word and add ay. For instance, chip becomes hipcay.
- **b.** If the word begins with a vowel, add way to the end of the word. For instance, else becomes elseway.

38. The current calendar, called the Gregorian calendar, was introduced in 1582. Every year divisible by four was declared to be a leap year, with the exception of the years ending in 00 (that is, those divisible by 100) and not divisible by 400. For instance, the years 1600 and 2000 are leap years, but 1700, 1800, and 1900 are not. Write a program that requests a year as input and states whether it is a leap year. (Test the program on the years 1994, 1995, 1900, and 2000.)

[Page 216]

39. Create a form with a text box and two buttons captioned Bogart and Raines. When Bogart is first pressed, the sentence "I came to Casablanca for the waters." is displayed in the text box. The next time Bogart is pressed, the sentence "I was misinformed." is displayed. When Raines is pressed, the sentence "But we're in the middle of the desert." is displayed. Run the program and then press Bogart, Raines, and Bogart to obtain a dialogue.

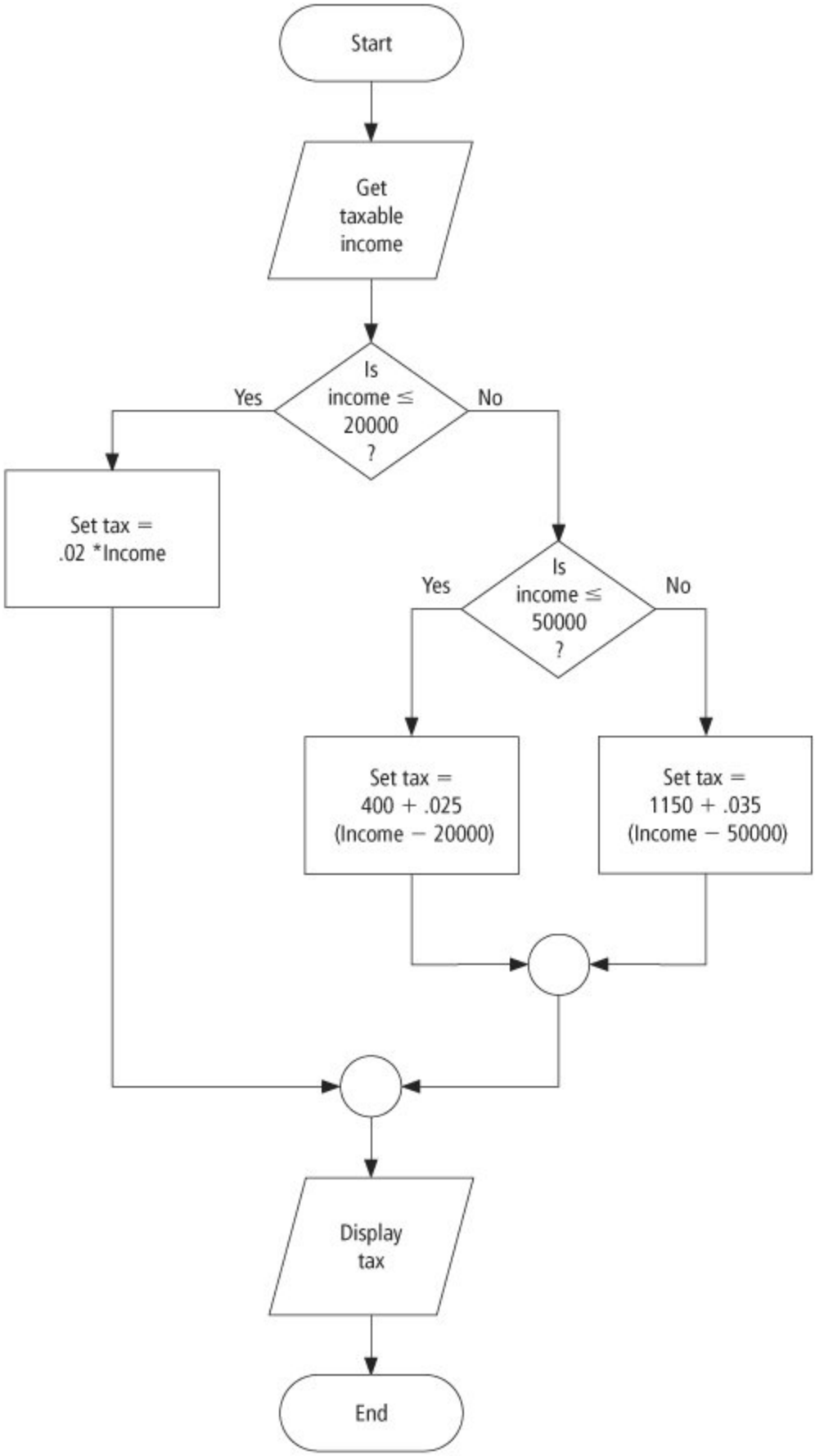
40. Write a program that allows the user to use a button to toggle the color of the text in a text box between black and red.

41. Write a program that allows the user ten tries to answer the question, "Which U.S. President was born on July 4?" After three incorrect guesses, the program should display the hint, "He once said, 'If you don't say anything, you won't be called upon to repeat it.'" in a message box. After seven incorrect guesses, the program should give the hint, "His nickname was 'Silent Cal.'" The number of guesses should be displayed in a text box. Note: Calvin Coolidge was born on July 4, 1872.

42. Write a program that reads a test score from a text box each time a button is clicked and then displays the two highest scores whenever a second button is clicked. Use two class-level variables to track the two highest scores.

43. The flowchart in Figure 5.3 (on the next page) calculates New Jersey state income tax. Write a program corresponding to the flowchart. (Test the program with taxable incomes of $15,000, $30,000, and $60,000.)

Figure 5.3. Flowchart for New Jersey state income tax program
(This item is displayed on page 217 in the print version)


Start
Get taxable income
Is income ≤ 20000 ?
Yes
No
Set tax = .02 *Income
Is income ≤ 50000 ?
Yes
No
Set tax = 400 + .025 (Income − 20000)
Set tax = 1150 + .035 (Income − 50000)
Display tax
End

44. Write a program to play "Hide and Seek" with the name of our programming language. When the button is pressed, the name should disappear and the caption on the button should change to "Show Name of Language." The next time the button is pressed, the name should reappear and the caption should revert to "Hide Name of Language," and so on.

Object	Property	Setting
frmHideSeek	Text	Hide and Seek
lblLanguage	Text	VB 2005
	Font.Size	26
btnDisplay	Text	Hide Name of Language

Solutions to Practice Problems 5.2

1.
```
If (num < 0) Then
   MsgBox("Number can't be negative.", 0, "Input Error")
   txtNumber.Clear()
   txtNumber.Focus()
Else
   txtSquareRoot.Text = CStr(Math.Sqrt(num))
End If
```

[Page 218]

2. The word "hello" will be displayed when (a < b)is true and (c < 5)is also true. That is, it will be displayed when both of these two conditions are true. The clearest way to write the block is

```
If (a < b) And (c < 5) Then
```

```
    txtBox.Text = "hello"
End If
```

PREV PREV NEXT NEXT

[Page 218 (continued)]

5.3. Select Case Blocks

A Select Case block is an efficient decision-making structure that simplifies choosing among several actions. It avoids complex 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. 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.

(This item is displayed on pages 218 - 219 in the print version)

The following program converts the finishing position in a horse race into a descriptive phrase. After the variable position is assigned a value from txtPosition, Visual Basic searches for the first Case clause whose value list contains that value and executes the succeeding statement. If the value of position is greater than 5, then the statement following Case Else is executed.

Object	Property	Setting
frmRace	Text	Horse Race

lblPosition	AutoSize	False
	Text	Finishing position (1, 2, 3,...):
txtPosition		
btnEvaluate	Text	Evaluate Position
txtOutcome	ReadOnly	True

```
Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click
  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"
```

[Page 219]

```
    Case 4, 5
      txtOutcome.Text = "You almost placed in the money."
    Case Else
      txtOutcome.Text = "Out of the money."
  End Select
End Sub
```

[Run, type 2 into the text box, and press the button.]

Example 2.

In the following variation of Example 1, the value lists specify ranges of values. The first value list provides another way to stipulate the numbers 1, 2, and 3. The second value list covers all numbers from 4 on.

```
Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click
  '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
```

[Run, type 2 into the text box, and press the button.]

[Page 220]

A typical form of the Select Case block is

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

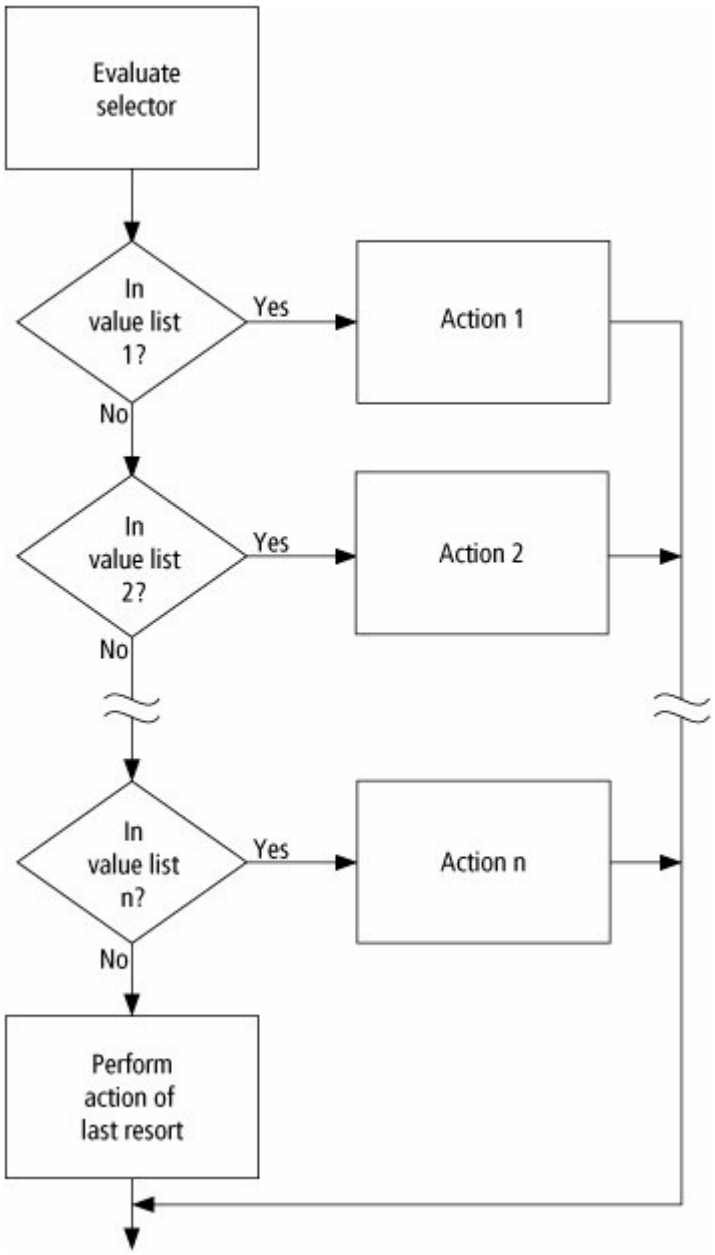
where Case Else (and its action) is optional, and each value list contains one or more of the following types of items:

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 expressed in the form `a To b`, where a and b are literals, variables, or expressions.

Different items appearing in the same list must be separated by commas. Each action consists of one or more statements. After the selector is evaluated, Visual Basic looks for the first value-list item including the value of the selector and carries out its associated action. (If the value of the selector appears in two different value lists, only the action associated with the first value list will be carried out.) If the value of the selector does not appear in any of the value lists and there is no Case Else clause, execution of the program will continue with the statement following the Select Case block.

Figure 5.4 (on the next page) contains the flowchart for a Select Case block. The pseudocode for a Select Case block is the same as for the equivalent If block.

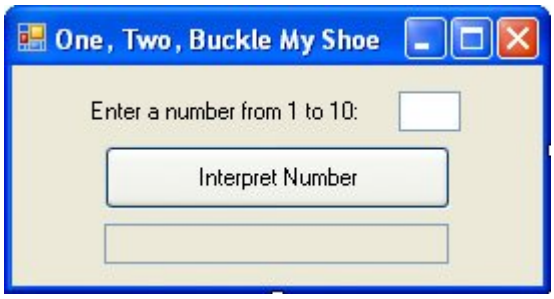
Figure 5.4. Flowchart for a Select Case block.
(This item is displayed on page 221 in the print version)



Example 3.

(This item is displayed on pages 220 - 222 in the print version)

The following program uses several different types of value lists. With the response shown, the second action was selected.



Object	Property	Setting
frmRhyme	Text	One, Two, Buckle My Shoe
lblEnterNum	Text	Enter a number from 1 to 10:
txtNumber		
btnInterpret	Text	Interpret Number
txtPhrase	ReadOnly	True

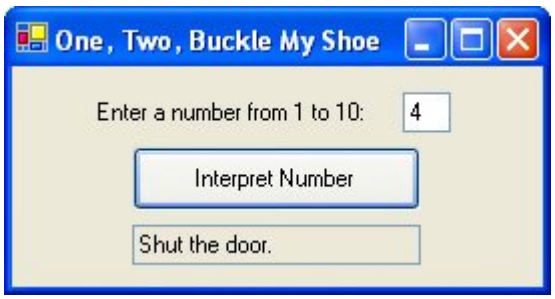
```
Private Sub btnInterpret_Click(...) Handles btnInterpret.Click
  'One, Two, Buckle My Shoe
  Dim x As Integer = 2, y As Integer = 3
  Dim num As Integer
  num = CInt(txtNumber.Text)
```

[Page 221]

```
  Select Case num
    Case y - x, x
      txtPhrase.Text = "Buckle 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
```

[Page 222]

[Run, type 4 into the text box, and press the button.]

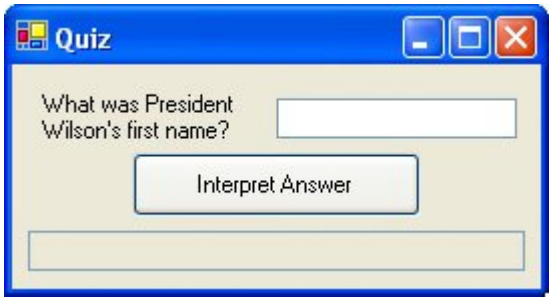


In each of the three preceding examples, the selector was a numeric variable; however, the selector also can be a string variable or an expression.

Example 4.

(This item is displayed on pages 222 - 223 in the print version)

The following program has the string variable firstName as a selector.



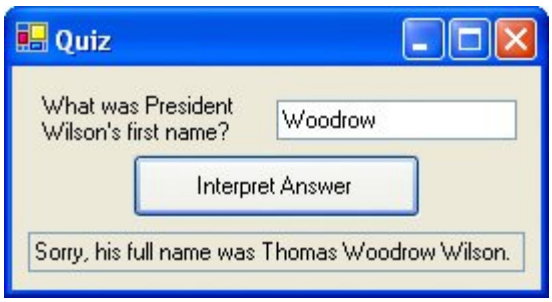
Object	Property	Setting
frmQuiz	Text	Quiz
lblQuestion	AutoSize	False
	Text	What was President Wilson's first name?
txtName		
btnInterpret	Text	Interpret Answer
txtReply	ReadOnly	True

```
Private Sub btnInterpret_Click(...) Handles btnInterpret.Click
  'Quiz
  Dim firstName As String
  firstName = txtName.Text.ToUpper
  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
```

[Page 223]

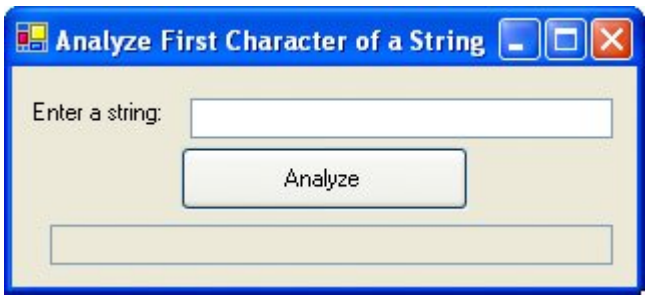
[Run, type "Woodrow" into the text box, and press the button.]



Example 5.

(This item is displayed on pages 223 - 224 in the print version)

The following program has the string selector anyString.Substring (0, 1). In the sample run, only the first action was carried out, even though the value of the selector was in both of the first two value lists. Visual Basic stops looking as soon as it finds the value of the selector.



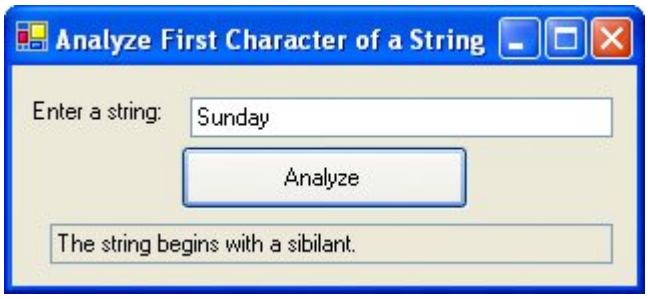
Object	Property	Setting
frmAnalyze	Text	Analyze First Character of a String

lblEnter	Text	Enter a string:
txtString		
btnAnalyze	Text	Analyze
txtResult	ReadOnly	True

```
Private Sub btnAnalyze_Click(...) Handles btnAnalyze.Click
  'Analyze the first character of a string
  Dim anyString As String
  anyString = txtString.Text.ToUpper
  Select Case anyString.Substring(0, 1)
    Case "S", "Z"
      txtResult.Text = "The string begins with a sibilant."
    Case "A" To "Z"
      txtResult.Text = "The string begins with a nonsibilant."
    Case "0" To "9"
      txtResult.Text = "The string begins with a digit."
    Case Is < "0"
      txtResult.Text = "The string begins with a character of " & _
                       "ANSI value less than 48."
    Case Else
      txtResult.Text = "The string begins with :  ;  <  =  > " & _
                       " ?  @  [  \  ]  ^  _  or '  .  "
  End Select
End Sub
```

[Page 224]

[Run, type "Sunday" into the text box, and press the button.]



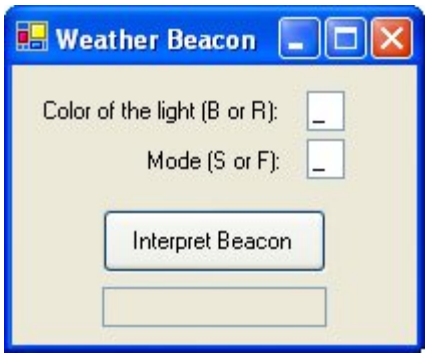
Example 6.

(This item is displayed on pages 224 - 225 in the print version)

The color of the beacon light atop Boston's John Hancock Building forecasts the weather according to the following rhyme:

Steady blue, clear view.
Flashing blue, clouds due.
Steady red, rain ahead.
Flashing red, snow instead.

The following program requests a color (Blue or Red) and a mode (Steady or Flashing) as input and displays the weather forecast. The program contains a Select Case block with a string expression as selector.



Object	Property	Setting
frmWeather	Text	Weather Beacon
lblColor	Text	Color of the light (B or R):
mtxtColor	Mask	L
lblMode	Text	Mode (S or F):
mtxtMode	Mask	L
btnInterpret	Text	Interpret Beacon
txtForecast	ReadOnly	True

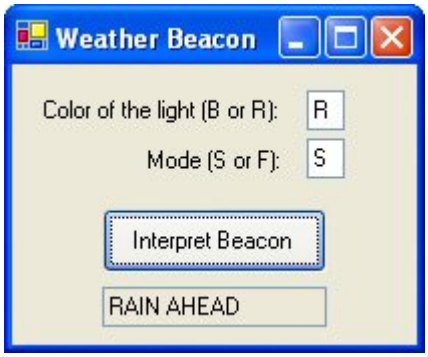
```
Private Sub btnInterpret_Click(...) Handles btnInterpret.Click
  'Interpret a weather beacon
  Dim color, mode As String
  color = mtxtColor.Text
  mode = mtxtMode.Text
  Select Case mode.ToUpper & color.ToUpper
    Case "SB"
      txtForecast.Text = "CLEAR VIEW"
    Case "FB"
      txtForecast.Text = "CLOUDS DUE"
    Case "SR"
      txtForecast.Text = "RAIN AHEAD"
```

[Page 225]

```
    Case "FR"
      txtForecast.Text = "SNOW AHEAD"
  End Select
```

```
End Sub
```

[Run, type "R" and "S" into the masked text boxes, and press the button.]



Example 7.

(This item is displayed on pages 225 - 226 in the print version)

Select Case is useful in defining functions that are not determined by a formula. The following program assumes that the current year is not a leap year:

Object	Property	Setting
frmSeasons	Text	Seasons
lblSeason	Text	Season:
txtSeason		
btnNumber	Text	Number of Days
txtNumDays	ReadOnly	True

```
Private Sub btnNumber_Click(...) Handles btnNumber.Click
  'Determine the number of days in a season
  Dim season As String
  season = txtSeason.Text
  txtNumDays.Text = season & " has " & NumDays(season) & " days."
```

```
End Sub

Function NumDays(ByVal season As String) As Integer
  'Look up the number of days in a given season
  Select Case season.ToUpper
    Case "WINTER"
      Return 87
    Case "SPRING"
      Return 92
    Case "SUMMER", "AUTUMN", "FALL"
      Return 93
  End Select
End Function
```

[Page 226]

[Run, type "Summer" into the text box, and press the button.]

Comments

1. 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. Otherwise, the clause is meaningless.

2. If the word `Is`, which should precede an inequality sign in a value list, is accidentally omitted, the editor will automatically insert it when checking the line.

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

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

   then the clause

   ```
   Case firstName.Length
   ```

   would be meaningless.

4. Variables are rarely declared inside an If ... Then block or a Select Case block. If so, such a variable has block-level scope; that is, the variable cannot be referred to by code outside of the

block.

5. In Appendix D, the section "Stepping through Programs Containing Selection Structures: Chapter 5" uses the Visual Basic debugging tools to trace the flow through a Select Case block.

Practice Problems 5.3

1. Suppose the selector of a Select Case block is the numeric variable num. Determine whether each of the following Case clauses is valid.

   a. `Case 1, 4, Is < 10`

   b. `Case Is < 5, Is >= 5`

   c. `Case num = 2`

2. Do the following two programs always produce the same output for a whole-number grade from 0 to 100?

[Page 227]

```
grade = CDbl(txtBox.Text)
Select Case grade
  Case Is >= 90
     txtOutput.Text = "A"
  Case Is >= 60
     txtOutput.Text = "Pass"
  Case Else
     txtOutput.Text = "Fail"
End Select
```

```
grade = CDbl(txtBox.Text)
Select Case grade
  Case Is >= 90
    txtOutput.Text = "A"
  Case 60 To 89
    txtOutput.Text = "Pass"
  Case 0 To 59
    txtOutput.Text = "Fail"
End Select
```

Exercises 5.3

In Exercises 1 through 8, for each of the responses shown in the parentheses, determine the output displayed in the text box when the button is clicked.

1.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim age, price As Double
  age = CDbl(InputBox("What is your age?"))
  Select Case age
    Case Is < 6
      price = 0
    Case 6 To 17
      price = 3.75
```

```
    Case Is >= 17
      price = 5
  End Select
  txtOutput.Text = "The price is "& FormatCurrency(price)
End Sub
```

(8.5, 17)

2.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim n As Double
  n = CDbl(InputBox("Enter a number from 5 to 12"))
  Select Case n
    Case 5
      txtOutput.Text = "case 1"
    Case 5 To 7
      txtOutput.Text = "case 2"
    Case 7 To 12
      txtOutput.Text = "case 3"
  End Select
End Sub
```

(7, 5, 11.2)

3.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim age As Integer
  age = CDbl(InputBox("Enter age (in millions of years)"))
```

[Page 228]

```
  Select Case age
    Case Is < 70
      txtOutput.Text = "Cenozoic Era"
    Case Is < 225
      txtOutput.Text = "Mesozoic Era"
    Case Is <= 600
      txtOutput.Text = "Paleozoic Era"
    Case Else
      txtOutput.Text = "?"
  End Select
End Sub
```

(100, 600, 700)

4.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim yearENIAC As Integer
  AskQuestion(yearENIAC)
  ProcessAnswer(yearENIAC)
End Sub
Sub AskQuestion(ByRef yearENIAC As Integer)
  'Ask question and obtain answer
  Dim message As String
  message = "In what year was the ENIAC computer completed?"
  yearENIAC = CInt(InputBox(message))
End Sub
```

```
Sub ProcessAnswer(ByVal yearENIAC As Integer)
  'Respond to answer
  Select Case yearENIAC
    Case 1945
      txtOutput.Text = "Correct."
    Case 1943 To 1947
      txtOutput.Text = "Close, 1945."
    Case Is < 1943
      txtOutput.Text = "Sorry, 1945. Work on the ENIAC "& _
                       "began in June 1943."
    Case Is > 1947
      txtOutput.Text = "No, 1945. By then IBM had built "& _
                       "a stored-program computer."
  End Select
End Sub
```

(1940, 1945, 1950)

5.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim name As String
  name = InputBox("Who developed the stored-program concept?")
  Select Case name.ToUpper
    Case "JOHN VON NEUMANN", "VON NEUMANN"
      txtOutput.Text = "Correct."
```

[Page 229]

```
    Case "JOHN MAUCHLY", "MAUCHLY", "J. PRESPER ECKERT", "ECKERT"
      txtOutput.Text = "He worked with the developer, "& _
                       "von Neumann, on the ENIAC."
    Case Else
      txtOutput.Text = "Nope."
  End Select
End Sub
```

(Grace Hopper, Eckert, John von Neumann)

6.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim message As String, a, b, c As Double
  message = "Analyzing solutions to the quadratic equation "
  message &= "AX^2 + BX + C = 0.  Enter the value for "
  a = CDbl(InputBox(message & "A"))
  b = CDbl(InputBox(message & "B"))
  c = CDbl(InputBox(message & "C"))
  Select Case (b ^ 2) - (4 * a * c)
    Case Is < 0
      txtOutput.Text = "The equation has no real solutions."
    Case 0
      txtOutput.Text = "The equation has exactly one solution."
    Case Is > 0
      txtOutput.Text = "The equation has two solutions."
  End Select
End Sub
```

(1,2,3; 1,5,1; 1,2,1)

7.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'State a quotation
  Dim num1, num2 As Double
  Dim word As String = "hello"
  num1 = 3
  num2 = CDbl(InputBox("Enter a number"))
  Select Case (2 * num2 - 1)
    Case num1 * num1
      txtBox.Text = "Less is more."
    Case Is &gt; word.Length
      txtBox.Text = "Time keeps everything from happening at once."
    Case Else
      txtBox.Text = "The more things change, the less "&amp; _
                    "they remain the same."
  End Select
End Sub
```

(2, 5, 6)

8.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim whatever As Double
  whatever = CDbl(InputBox("Enter a number:"))
```

[Page 230]

```
  Select Case whatever
    Case Else
      txtOutput.Text = "Hi"
  End Select
End Sub
```

(7,-1)

In Exercises 9 through 16, identify the errors.

9.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num As Double = 2
  Select Case num
    txtOutput.Text = "Two"
  End Select
End Sub
```

10.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num1 As Double = 5
  Dim num2 As Double = 2
  Select Case num1
    Case 3 <= num1 <= 10
      txtOutput.Text = "between 3 and 10."
    Case num2 To 5; 4
      txtOutput.Text = "near 5."
```

```
  End Select
End Sub
```

11.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim a As String
  a = InputBox("What is your name?")
  Select Case a
    Case a = "Bob"
      txtOutput.Text = "Hi, Bob."
    Case Else
  End Select
End Sub
```

12.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim word As String = "hello"
  Select Case word.Substring(0,1)
    Case h
      txtOutput.Text = "begins with h."
  End Select
End Sub
```

13.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim word As String
  word = InputBox("Enter a word from the United States motto.")
  Select Case word.ToUpper
    Case "E"
      txtOutput.Text = "This is the first word of the motto."
```

[Page 231]

```
    Case word.Substring(0,1) = "P"
      txtOutput.Text = "The second word is PLURIBUS."
    Case Else
      txtOutput.Text = "The third word is UNUM."
  End Select
End Sub
```

14.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim num As Double = 5
  Select Case num
    Case 5, Is <> 5
      txtOutput.Text = "five"
    Case Is > 5
      txtOutput.Text = "greater than five"
End Sub
```

15.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim fruit As String = "Peach"
  Select Case fruit.ToUpper
    Case Is >= "Peach"
      txtOutput.Text = "Georgia"
    Case "ORANGE To PEACH"
      txtOutput.Text = "Ok"
  End Select
End Sub
```

16.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim purchase As Double
  purchase = CDbl(InputBox("Quantity purchased?"))
  Select Case purchase
    Case purchase < 10000
      txtOutput.Text = "Five dollars per item."
    Case Is 10000 To 30000
      txtOutput.Text = "Four dollars per item."
    Case Is > 30000
      txtOutput.Text = "Three dollars per item."
  End Select
End Sub
```

In Exercises 17 through 22, suppose the selector of a Select Case block, word, evaluates to a String value. Determine whether the Case clause is valid.

17. `Case "un" & "til"`

18. `Case "hello", Is < "goodbye"`

19. `Case 0 To 9`

20. `Case word <> "No"`

21. `Case "abc".Substring(0, 1)`

22. `Case Is <> "No"`

[Page 232]

In Exercises 23 through 26, rewrite the code using a Select Case block.

23.
```
If a = 1 Then
  txtOutput.Text = "one"
```

```
Else
  If a > 5 Then
    txtOutput.Text = "two"
  End If
End If
```

24.

```
If a = 1 Then
  lstOutput.Items.Add("lambs")
End If
If ((a <= 3) And (a < 4)) Then
  lstOutput.Items.Add("eat")
End If
If ((a = 5) Or (a > 7)) Then
  lstOutput.Items.Add("ivy")
End If
```

25.

```
If a < 5 Then
  If a = 2 Then
    txtOutput.Text = "yes"
  Else
    txtOutput.Text = "no"
  End If
Else
  If a = 2 Then
   txtOutput.Text = "maybe"
  End If
End If
```

26.

```
If a = 3 Then
  a = 1
End If
If a = 2 Then
  a = 3
End If
If a = 1 Then
  a = 2
End If
```

27. Table 5.5 gives the terms used by the National Weather Service to describe the degree of cloudiness. Write a program that requests the percentage of cloud cover as input and then displays the appropriate descriptor.

Table 5.5. Cloudiness descriptors.
(This item is displayed on page 233 in the print version)

Percentage of Cloud Cover	Descriptor

030	clear
3170	partly cloudy
7199	cloudy
100	overcast

28. Table 5.6 shows the location of books in the library stacks according to their call numbers. Write a program that requests the call number of a book as input and displays the location of the book.

[Page 233]

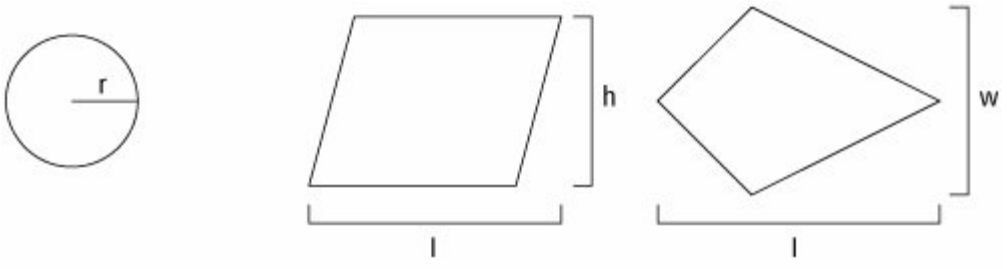
Table 5.6. Location of library books.

Call Numbers	Location
100 to 199	basement
200 to 500 and over 900	main floor
501 to 900 except 700 to 750	upper floor
700 to 750	archives

29. Write a program that requests a month of the year and then gives the number of days in the month. If the month is February, the user should be asked whether the current year is a leap year.

30. Figure 5.5 shows some geometric shapes and formulas for their areas. Write a program that requests the user to select one of the shapes, requests the appropriate lengths, and then gives the area of the figure. The areas should be computed by Function procedures.

Figure 5.5. Areas of geometric shapes.



31. Write a program that requests an exam score and assigns a letter grade with the scale 90100 (A), 8089 (B), 7079 (C), 6069 (D), 059 (F). The computation should be carried out in a Function procedure. (Test the program with the grades 84, 100, and 57.)

32. Computerized quiz show. Write a program that asks the contestant to select one of the numbers 1, 2, or 3 and then calls a Sub procedure that asks the question and requests the answer. The program should then tell the contestant if the answer is correct. Use the following three questions:

[Page 234]

1. Who was the only living artist to have his work displayed in the Grand Gallery of the Louvre?
2. Who said, "Computers are useless. They can only give you answers."?
3. By what name is Pablo Blasio better known?

Note: These questions have the same answer, Pablo Picasso.

33. IRS informants are paid cash awards based on the value of the money recovered. If the information was specific enough to lead to a recovery, the informant receives 10 percent of the first $75,000, 5 percent of the next $25,000, and 1 percent of the remainder, up to a maximum award of $50,000. Write a program that requests the amount of the recovery as input and displays the award. (Test the program on the amounts $10,000, $125,000, and $10,000,000.)(Note:The source of this formula is The Book of Inside Information, Boardroom Books, 1993.)

34. Table 5.7 contains information on several states. Write a program that requests a state and category (flower, motto, and nickname) as input and displays the requested information. If the state or category requested is not in the table, the program should so inform the user.

Table 5.7. State flowers, nicknames, and mottoes.

State	Flower	Nickname	Motto
California	Golden Poppy	Golden State	Eureka
Indiana	Peony	Hoosier State	Crossroads of America
Mississippi	Magnolia	Magnolia State	By valor and arms
New York	Rose	Empire State	Ever upward

35. Write a program that, given the last name of one of the five most recent presidents, displays his state and a colorful fact about him. (Hint: The program might need to request

further information.) Note: Carter: Georgia; The only soft drink served in the Carter White House was Coca-Cola. Reagan: California; His Secret Service code name was Rawhide. George H. W. Bush: Texas; He was the third left-handed president. Clinton: Arkansas; In college he did a good imitation of Elvis Presley. George W. Bush: Texas; He once owned the Texas Rangers baseball team.

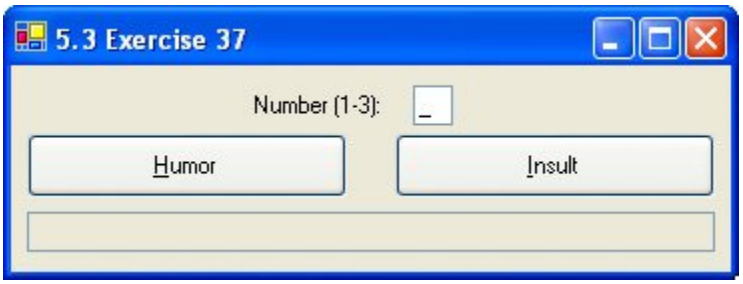
36. Table 5.8 contains the meanings of some abbreviations doctors often use for prescriptions. Write a program that requests an abbreviation and gives its meaning. The user should be informed if the meaning is not in the table.

Table 5.8. Physicians' abbreviations.

Abbreviation	Meaning
ac	before meals
ad lib	freely as needed
bid	twice daily
gtt	a drop
hs	at bedtime
qid	four times a day

[Page 235]

37. The user enters a number into a masked text box and then clicks on the appropriate button to have either one of three pieces of humor or one of three insults displayed in a text box below the buttons. Place the humor and insults in Function procedures, with Select Case statements in each to return the appropriate phrase. Also, if the number entered is not between 1 and 3, the masked text box should be cleared. Note: Some possible bits of humor are "I can resist everything except temptation," "I just heard from Bill Bailey. He's not coming home," and "I have enough money to last the rest of my life, unless I buy something." Some possible insults are "How much would you charge to haunt a house?" "I bet you have no more friends than an alarm clock," and "When your IQ rises to 30, sell."



Object	Property	Setting
frmExercise 37		
lblNumber	Text	Number(13):
mtxtNumber	Mask	0
btnHumor	Text	&Humor
btnInsult	Text	&Insult
txtSentence	ReadOnly	True

Solutions to Practice Problems 5.3

1. **a.** Valid. These items are redundant because 1 and 4 are just special cases of `Is < 10`. However, this makes no difference in Visual Basic.

   b. Valid. These items are contradictory. However, Visual Basic looks at them one at a time until it finds an item containing the value of the selector. The action following this Case clause will always be carried out.

   c. Not valid. It should be `Case 2`.

2. Yes. However, the program on the right is clearer and therefore preferable.

PREV PREV NEXT NEXT

[Page 235 (continued)]

5.4. A Case Study: Weekly Payroll

This case study processes a weekly payroll using the 2005 Employer's Tax Guide. Table 5.9 shows typical data used by a company's payroll office. These data are processed to produce the information in Table 5.10 that is supplied to each employee along with his or her paycheck. The program should request the data from Table 5.9 for an individual as input and produce output similar to that in Table 5.10.

[Page 236]

Table 5.9. Employee data.
(This item is displayed on page 235 in the print version)

Name	Hourly Wage	Hours Worked	Withholding Exemptions	Marital Status	Previous Year-to-Date Earnings
Al Clark	$45.50	38	4	Married	$88,600.00
Ann Miller	$44.00	35	3	Married	$68,200.00
John Smith	$17.95	50	1	Single	$30,604.75
Sue Taylor	$25.50	43	2	Single	$36,295.50

Table 5.10. Payroll information.

Name	Current Earnings	Yr. to Date Earnings	FICA Taxes	Income Tax Wh.	Check Amount
Al Clark	$1,729.00	$90,329.00	$111.87	$206.26	$1,410.87

The items in Table 5.10 should be calculated as follows:

Current Earnings: hourly wage times hours worked (with time-and-a-half after 40 hours)

Year-to-Date Earnings: previous year-to-date earnings plus current earnings

FICA Taxes: sum of 6.2 percent of first $90,000 of earnings (Social Security benefits tax) and 1.45 percent of total wages (Medicare tax)

Federal Income Tax Withheld: subtract $61.54 from the current earnings for each withholding exemption and use Table 5.11 or Table 5.12, depending on marital status

Check Amount:[current earnings] - [FICA taxes] - [income tax withheld]

Table 5.11. 2005 Federal income tax withheld for a single person paid weekly.

Adjusted Weekly Income	Income Tax Withheld
$0 to $51	$0
Over $51 to $188	10% of amount over $51
Over $188 to $606	$13.70 + 15% of amount over $188
Over $606 to $1,341	$76.40 + 25% of amount over $606
Over $1,341 to $2,922	$260.15 + 28% of amount over $1,341
Over $2,922 to $6,313	$702.83 + 33% of amount over $2,922
Over $6,313	$1,821.86 + 35% of amount over $6,313

Table 5.12. 2005 Federal income tax withheld for a married person paid weekly.

Adjusted Weekly Income	Income Tax Withheld
$0 to $154	$0
Over $154 to $435	10% of amount over $154
Over $435 to $1,273	$28.10 + 15% of amount over $435
Over $1,273 to $2,322	$153.80 + 25% of amount over $1,273
Over $2,322 to $3,646	$416.05 + 28% of amount over $2,322
Over $3,646 to $6,409	$786.77 + 33% of amount over $3,646
Over $6,409	$1,698.56 + 35% of amount over $6,409

[Page 237]

Designing the Weekly Payroll Program

After the data for an employee have been gathered from the text boxes, the program must compute the five items appearing in Table 5.10 and then display the payroll information. The five computations form the basic tasks of the program:

1. Compute current earnings.
2. Compute year-to-date earnings.
3. Compute FICA tax.
4. Compute federal income tax withheld.
5. Compute paycheck amount (that is, take-home pay).

Tasks 1, 2, 3, and 5 are fairly simple. Each involves applying a formula to given data. (For instance, if hours worked are at most 40, then [Current Earnings] = [Hourly Wage] times [Hours Worked].) Thus, we won't break down these tasks any further. Task 4 is more complicated, so we continue to divide it into smaller subtasks.

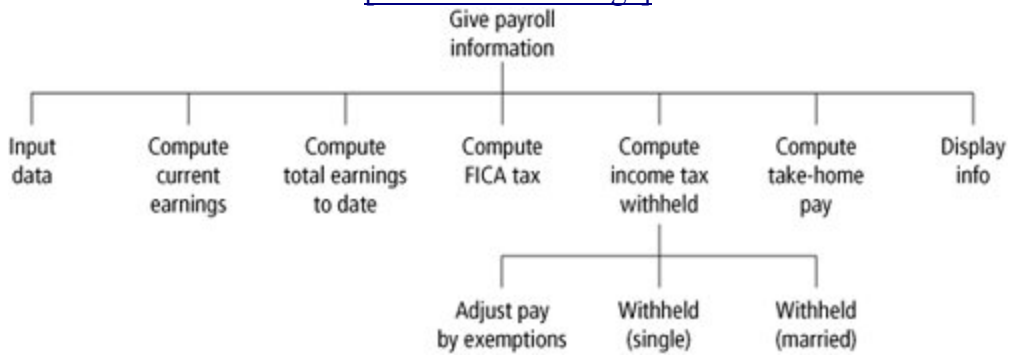
> 4. Compute Federal Income Tax Withheld. First, the employee's pay is adjusted for exemptions, and then the amount of income tax to be withheld is computed. The computation of the income tax withheld differs for married and single individuals. Task 4 is, therefore, divided into the following subtasks:
>
> 4.1. Compute pay adjusted by exemptions.
>
> 4.2. Compute income tax withheld for single employee.

4.3. Compute income tax withheld for married employee.

The hierarchy chart in Figure 5.6 shows the stepwise refinement of the problem.

Figure 5.6. Hierarchy chart for the weekly payroll program.

[View full size image]



Pseudocode for the Display Payroll Event

```
INPUT employee data (Sub procedure InputData)
COMPUTE CURRENT GROSS PAY (Function Gross_Pay)
COMPUTE TOTAL EARNINGS TO DATE (Function Total_Pay)
COMPUTE FICA TAX (Function FICA_Tax)
COMPUTE FEDERAL TAX (Function Fed_Tax)
```

[Page 238]

```
  Adjust pay for exemptions
  If employee is single Then
    COMPUTE INCOME TAX WITHHELD (Function TaxSingle)
  Else
    COMPUTE INCOME TAX WITHHELD (Function TaxMarried)
  End If
    COMPUTE PAYCHECK AMOUNT (Function Net_Check)
DISPLAY PAYROLL INFORMATION (Sub procedure ShowPayroll)
```

Writing the Weekly Payroll Program

The btnDisplay_Click event procedure calls a sequence of seven procedures. Table 5.13 shows the tasks and the procedures that perform the tasks.

Table 5.13. Tasks and their procedures.

Task	Procedure

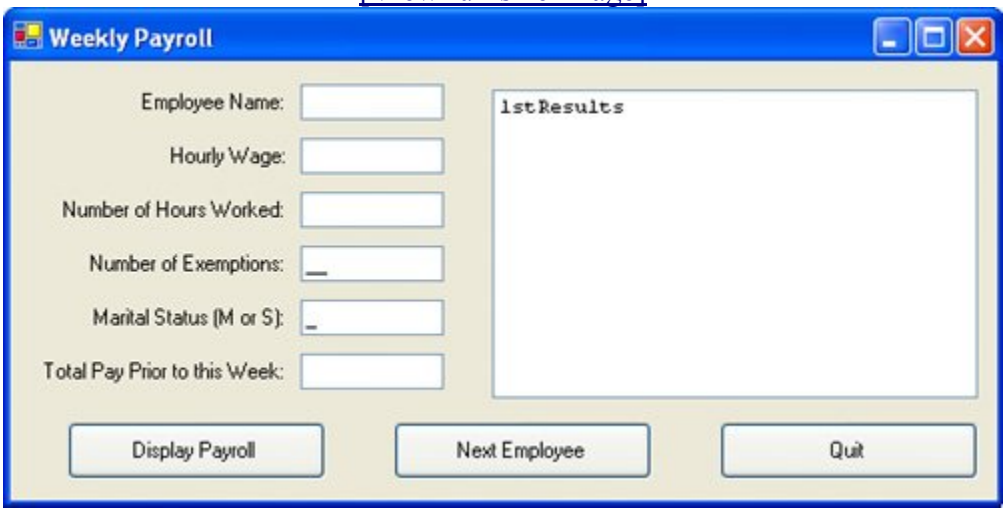
0. Input employee data.	InputData
1. Compute current earnings.	Gross_Pay
2. Compute year-to-date earnings.	Total_Pay
3. Compute FICA tax.	FICA_Tax
4. Compute federal income tax withheld.	Fed_Tax
4.1 Compute adjusted pay.	Fed_Tax
4.2 Compute amount withheld for single employee.	TaxSingle
4.3 Compute amount withheld for married employee.	TaxMarried
5. Compute paycheck amount.	Net_Check
6. Display payroll information.	ShowPayroll

The Program and the User Interface

Figure 5.7 and Table 5.14 define the user interface for the Weekly Payroll Program.

Figure 5.7. Template for entering payroll data.

[View full size image]



[Page 239]

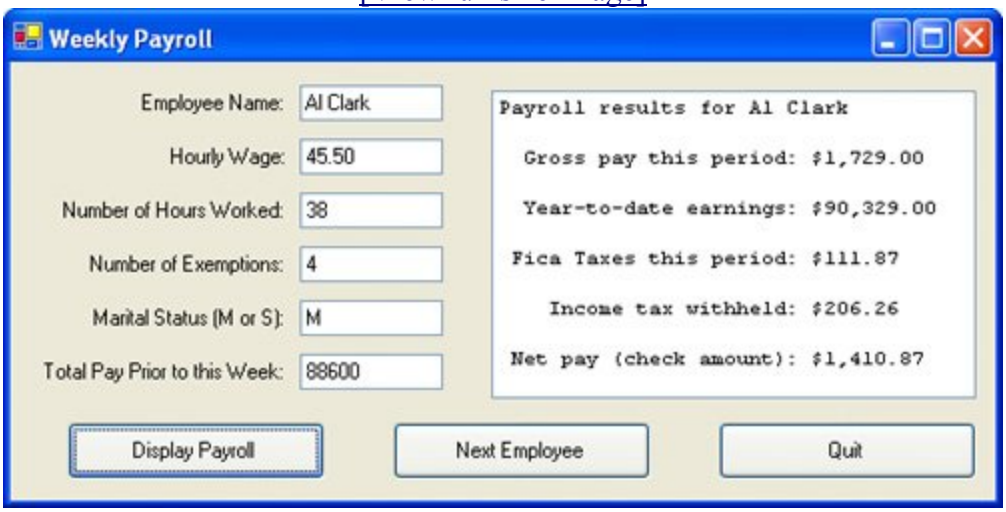
Table 5.14. Objects and initial properties for the weekly payroll program.

Object	Property	Setting
frmPayroll	Text	Weekly Payroll

lblName	Text	Employee Name:
txtName		
lblWage	Text	Hourly Wage:
txtWage		
lblHours	Text	Number of Hours Worked:
txtHours		
lblExempts	Text	Number of Exemptions:
txtExempts		
lblMarital	Text	Marital Status (M or S):
mtxtMarital	Mask	L
lblPriorPay	Text	Total Pay Prior to this Week:
txtPriorPay		
btnDisplay	Text	Display Payroll
btnNext	Text	Next Employee
btnQuit	Text	Quit
lstResults		

Figure 5.8. Sample run of weekly payroll program.

[View full size image]



[Page 240]

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim empName As String = "" 'Name of employee
```

```
  Dim hrWage As Double        'Hourly wage
  Dim hrsWorked As Double     'Hours worked this week
  Dim exemptions As Integer   'Number of exemptions for employee
  Dim mStatus As String = "" 'Marital status: S - Single; M - Married
  Dim prevPay As Double       'Total pay for year excluding this week
  Dim pay As Double           'This week's pay before taxes
  Dim totalPay As Double      'Total pay for year including this week
  Dim ficaTax As Double       'FICA taxes for this week
  Dim fedTax As Double        'Federal income tax withheld this week
  Dim check As Double         'Paycheck this week (take-home pay)
  'Obtain data, compute payroll, display results
  InputData(empName, hrWage, hrsWorked, exemptions, _
                                        mStatus, prevPay)      'Task 0
  pay = Gross_Pay(hrWage, hrsWorked)                           'Task 1
  totalPay = Total_Pay(prevPay, pay)                           'Task 2
  ficaTax = FICA_Tax(pay, prevPay, totalPay)                   'Task 3
  fedTax = Fed_Tax(pay, exemptions, mStatus)                   'Task 4
  check = Net_Check(pay, ficaTax, fedTax)                      'Task 5
  ShowPayroll(empName, pay, totalPay, ficaTax, fedTax, check)  'Task 6
End Sub

Private Sub btnNext_Click(...) Handles btnNext.Click
  'Clear all masked text boxes for next employee's data
  txtName.Clear()
  txtWage.Clear()
  txtHours.Clear()
  txtExempts.Clear()
  mtxtMarital.Clear()
  txtPriorPay.Clear()
  lstResults.Items.Clear()
End Sub

Private Sub btnQuit_Click(...) Handles btnQuit.Click
  End
End Sub

Sub InputData(ByRef empName As String, ByRef hrWage As Double, _
              ByRef hrsWorked As Double, ByRef exemptions As Integer, _
              ByRef mStatus As String, ByRef prevPay As Double)
  'Task 0: Get payroll data for employee
  empName = txtName.Textd
  hrWage = CDbl(txtWage.Text)
  hrsWorked = CDbl(txtHours.Text)
  exemptions = CInt(txtExempts.Text)
  mStatus = mtxtMarital.Text.ToUpper.Substring(0, 1)  'M or S
  prevPay = CDbl(txtPriorPay.Text)
End Sub

Function Gross_Pay(ByVal hrWage As Double, ByVal hrsWorked As Double) _
         As Double
  'Task 1: Compute weekly pay before taxes
  If hrsWorked <= 40 Then
    Return hrsWorked * hrWage
```

[Page 241]

```
  Else
    Return 40 * hrWage + (hrsWorked - 40) * 1.5 * hrWage
  End If
End Function
```

```
Function Total_Pay(ByVal prevPay As Double, _
                   ByVal pay As Double) As Double
  'Task 2: Compute total pay before taxes
  Return prevPay + pay
End Function

Function FICA_Tax(ByVal pay As Double, ByVal prevPay As Double, _
                  ByVal totalPay As Double) As Double
  'Task 3: Compute social security and medicare tax
  Dim socialSecurity As Double      'Social Security tax for this week
  Dim medicare As Double            'Medicare tax for this week
  Dim sum As Double                 'Sum of above two taxes
  If totalPay <= 90000 Then
    socialSecurity = 0.062 * pay
  ElseIf prevPay < 90000 Then
    socialSecurity = 0.062 * (90000 - prevPay)
  End If
  medicare = 0.0145 * pay
  sum = socialSecurity + medicare
  Return Math.Round(sum, 2)         'Round to nearest cent
End Function

Function Fed_Tax(ByVal pay As Double, ByVal exemptions As Integer, _
                 ByVal mStatus As String) As Double
  'Task 4.1: Compute federal income tax rounded to two decimal places
  Dim adjPay As Double
  Dim tax As Double                 'Unrounded federal tax
  adjPay = pay - (61.54 * exemptions)
  If adjPay < 0 Then
    adjPay = 0
  End If
  If mStatus = "S" Then
    tax = TaxSingle(adjPay)          'Task 4.2
  Else
    tax = TaxMarried(adjPay)         'Task 4.3
  End If
  Return Math.Round(tax, 2)          'Round to nearest cent
End Function

Function TaxSingle(ByVal adjPay As Double) As Double
  'Task 4.2: Compute federal tax withheld for single person
  Select Case adjPay
    Case 0 To 51
      Return 0
    Case 51 To 188
      Return 0.1 * (adjPay - 51)
    Case 188 To 606
      Return 13.7 + 0.15 * (adjPay - 188)
    Case 606 To 1341
      Return 76.4 + 0.25 * (adjPay - 606)
```

[Page 242]

```
    Case 1341 To 2922
      Return 260.15 + 0.28 * (adjPay - 1341)
    Case 2922 To 6313
      Return 702.83 + 0.33 * (adjPay - 2922)
    Case Is > 6313
      Return 1821.86 + 0.35 * (adjPay - 6313)
  End Select
End Function
```

```
Function TaxMarried(ByVal adjPay As Double) As Double
  'Task 4.3: Compute federal tax withheld for married person
  Select Case adjPay
    Case 0 To 154
      Return 0
    Case 154 To 435
      Return 0.1 * (adjPay - 154)
    Case 435 To 1273
      Return 28.1 + 0.15 * (adjPay - 435)
    Case 1273 To 2322
      Return 153.8 + 0.25 * (adjPay - 1273)
    Case 2322 To 3646
      Return 416.05 + 0.28 * (adjPay - 2322)
    Case 3646 To 6409
      Return 786.77 + 0.33 * (adjPay - 3646)
    Case Is > 6409
      Return 1698.56 + 0.35 * (adjPay - 6409)
  End Select
End Function

Function Net_Check(ByVal pay As Double, ByVal ficaTax As Double, _
                   ByVal fedTax As Double) As Double
  'Task 5: Compute amount of money given to employee
  Return pay - ficaTax - fedTax
End Function

Sub ShowPayroll(ByVal empName As String, ByVal pay As Double, _
                ByVal totalPay As Double, ByVal ficaTax As Double, _
                ByVal fedTax As Double, ByVal check As Double)
  'Task 6: Display results of payroll computations
  Dim fmtStr As String = "{0,24} {1,-10:C}"
  With lstResults.Items
    .Clear()
    .Add("Payroll results for " & empName)
    .Add("")
    .Add(String.Format(fmtStr, "Gross pay this period:", pay))
    .Add("")
    .Add(String.Format(fmtStr, "Year-to-date earnings:", totalPay))
    .Add("")
    .Add(String.Format(fmtStr, "Fica Taxes this period:", ficaTax))
    .Add("")
    .Add(String.Format(fmtStr, "Income tax withheld:", fedTax))
    .Add("")
    .Add(String.Format(fmtStr, "Net pay (check amount):", check))
  End With
End Sub
```

[Page 243]

Comments

1. In the function FICA_Tax, care has been taken to avoid computing Social Security benefits tax on income in excess of $90,000 per year. The logic of the program makes sure an employee whose income crosses the $90,000 threshold during a given week is taxed only on the difference between $90,000 and his previous year-to-date income.

2. The two functions TaxMarried and TaxSingle use Select Case blocks to incorporate the tax brackets given in Tables 5.11 and 5.12 for the amount of federal income tax withheld. The upper limit of each Case clause is the same as the lower limit of the next Case clause. This ensures that fractional values for adjPay, such as 61.54 in the TaxSingle function, will be properly treated as part of the higher salary range.

PREV PREV NEXT NEXT

[Page 243 (continued)]

Chapter 5 Summary

1. The function Chr associates a character with each number from 0 through 255 as determined by the ANSI table. The function Asc is the inverse of the Chr function.

2. The relational operators are <,>,=,<>,<=,and >=.

3. The principal logical operators are And, Or, and Not.

4. A condition is an expression involving literals, variables, functions, and operators (arithmetic, relational, or logical) that can be evaluated as either True or False.

5. The value of a variable or expression of Boolean data type is either True or False.

6. An If block decides what action to take depending on the truth values of one or more conditions. To allow several courses of action, the If, ElseIf, and Else parts of an If statement can contain other If statements.

7. A Select Case block selects from one of several actions depending on the value of an expression, called the selector. The entries in the value lists should have the same type as the selector.

PREV PREV NEXT NEXT

[Page 243 (continued)]

Chapter 5 Programming Projects

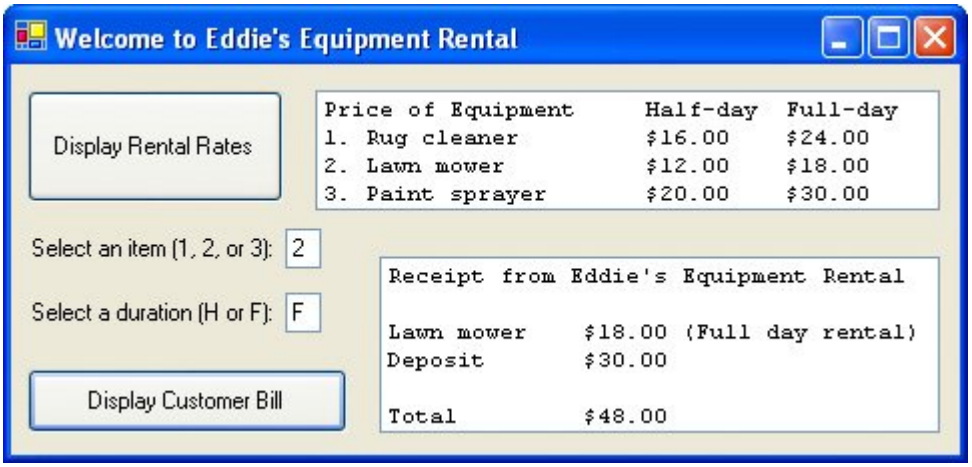
1. Table 5.15 gives the price schedule for Eddie's Equipment Rental. Full-day rentals cost one-and-a-half times half-day rentals. Write a program that displays Table 5.15 in a list box when an appropriate button is clicked and displays a bill in another list box based on the item number and time period chosen by a customer. The bill should include a $30.00 deposit. A possible form layout and sample run are shown in Figure 5.9.

[Page 244]

Table 5.15. Price schedule for Eddie's Equipment Rental.
(This item is displayed on page 243 in the print version)

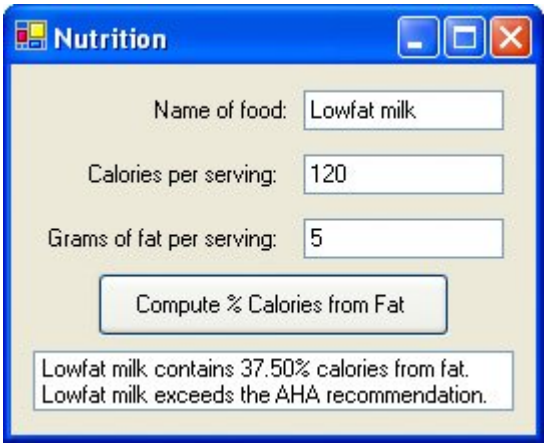
Piece of Equipment	Half-Day	Full-Day
1.Rug cleaner	$16.00	$24.00
2.Lawn mower	$12.00	$18.00
3.Paint sprayer	$20.00	$30.00

Figure 5.9. Form layout and sample run for Programming Project 1.



2. The American Heart Association suggests that at most 30 percent of the calories in our diet come from fat. Although food labels give the number of calories and amount of fat per serving, they often do not give the percentage of calories from fat. This percentage can be calculated by multiplying the number of grams of fat in one serving by 9, dividing that number by the total number of calories per serving, and multiplying the result by 100. Write a program that requests the name, number of calories per serving, and the grams of fat per serving as input, and tells us whether the food meets the American Heart Association recommendation. A sample run is as in Figure 5.10.

Figure 5.10. Sample run of Programming Project 2.



3. Table 5.16 gives the 2004 federal income tax rate schedule for single taxpayers. Write a program that requests taxable income and calculates federal income tax. Use a Sub procedure for the input and a Function procedure to calculate the tax.

Table 5.16. 2004 federal income tax rates for single tax payers.

Taxable Income Over	But Not Over	Your Tax Is	Of Amount Over
$0	$7,150	10%	$0
$7,150	$29,050	$715.00 + 15%	$7,150
$29,050	$70,350	$4,000.00 + 25%	$29,050
$70,350	$146,750	$14,325.00 + 28%	$70,350
$146,750	$319,100	$35,717.00 + 33%	$146,750
$319,100		$92,592.50 + 35%	$319,100

4. Write a program to determine the real roots of the quadratic equation $ax^2 + bx + c = 0$ (where a  0) after requesting the values of a, b, and c. Use a Sub procedure to ensure that a is nonzero. Note: The equation has 2, 1, or 0 solutions depending on whether the value of b^2 - 4*a*c is positive, zero, or negative. In the first two cases, the solutions are given by the quadratic formula (-b ± Sqrt(b^2 - 4*a*c))/2*a). Test the program with the following sets of coefficients:

a = 1	b = -11	c = 28	Solutions are 4 and 7
a = 1	b = -6	c = 9	Solution is 3
a = 1	b = 4	c = 5	No solution

5. Table 5.17 contains seven proverbs and their truth values. Write a program that presents these proverbs one at a time and asks the user to evaluate them as true or false. The program should then tell the user how many questions were answered correctly and display one of the following evaluations: Perfect (all correct), Excellent (5 or 6 correct), You might consider taking Psychology 101 (less than 5 correct).

Table 5.17. Seven proverbs.

Proverb	Truth Value
The squeaky wheel gets the grease.	True
Cry and you cry alone.	True
Opposites attract.	False
Spare the rod and spoil the child.	False
Actions speak louder than words.	True

Familiarity breeds contempt.	False
Marry in haste, repent at leisure.	True

Source: "You Know What They Say..., " by Alfie Kohn, Psychology Today, April 1988.

[Page 246]

6. Write a program to analyze a mortgage. The user should enter the amount of the loan, the annual rate of interest, and the duration of the loan in months. When the user clicks on the button, the information that was entered should be checked to make sure it is reasonable. If bad data have been supplied, the user should be so advised. Otherwise, the monthly payment and the total amount of interest paid should be displayed. The formula for the monthly payment is

payment = p * r / (1 - (1 + r)^(-n)),

where p is the amount of the loan, r is the monthly interest rate (annual rate divided by 12) given as a number between 0 (for 0 percent) and 1 (for 100 percent), and n is the duration of the loan in months. The formula for the total interest paid is

total interest = n * payment - p.

(Test the program for a mortgage of $240,000 at 6% annual rate of interest, and duration 360 months. Such a mortgage will have a monthly payment of $1,438.92 and total interest of $278,011.65.)

7. Five, Six, Pick up Sticks.

Write a program that allows the user to challenge the computer to a game of Pick-up-Sticks. Here is how the game works. The user chooses the number of matchsticks (from 5 to 50) to place in a pile. Then, the computer chooses who will go first. At each turn, the contestant can remove one, two, or three matchsticks from the pile. The contestant who chooses the last matchstick loses.

The computer should make the user always select from a pile where the number of matchsticks has a remainder of 1 when divided by 4. For instance, if the user initially chooses a number of matchsticks that has a remainder of 1 when divided by 4, then the computer should have the user go first. Otherwise, the computer should go first and remove the proper number of matchsticks. [Note: The remainder when n is divided by 4 is (n Mod 4).] After writing the program, play a few games with the computer and observe that the computer always wins.

Figure 5.11. A possible outcome of Programming Project 7.

[View full size image]


Five Six, Pick up Sticks
Give the Rules of the Game
Begin a New Game
How many matchsticks would you like to pick up?
One
Two
Three
Current status of the pile:
Hooray
I select 2 matchsticks. I win.
OK

PREV NEXT