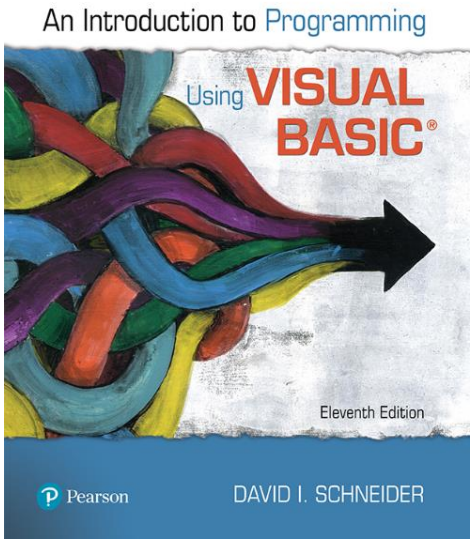
Introduction to Programming Using Visual Basic 11th edition Schneider Solutions Manual

SKU: 9780135862162-SOLUTIONS

INSTRUCTOR SOLUTIONS MANUAL

to accompany
An Introduction to Programming
Using VISUAL BASIC, 11th Edition

by David I. Schneider



Copyright © 2020 by Pearson Higher Education. All rights reserved.

"Visual Studio 2017 Community Edition" can be installed on PCs running Windows 7 (along with SP 1), Windows 8, or Windows10. It can be downloaded free of charge from the website

https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2017

CONTENTS

Chapter 2 Visual Basic Controls and Events
Exercises 2.2 Visual Basic Controls 5
Exercises 2.3 Visual Basic Events 11

Chapter 3 Variables, Input, and Output
Exercises 3.1 Numbers 16
Exercises 3.2 Strings 21
Exercises 3.3 Input and Output 29
Programming Projects 36

Chapter 4 Decisions
Exercises 4.1 Relational and Logical Operators 40
Exercises 4.2 If Blocks 41
Exercises 4.3 Select Case Blocks 56
Exercises 4.4 Input via User Selection 65
Programming Projects 79

Chapter 5 General Procedures
Exercises 5.1 Function Procedures 87
Exercises 5.2 Sub Procedures, Part I 95
Exercises 5.3 Sub Procedures, Part II 103
Programming Projects 109

Chapter 6 Repetition
Exercises 6.1 Do Loops 119
Exercises 6.2 For…Next Loops 127
Exercises 6.3 List Boxes and Loops 141
Programming Projects 155

Chapter 7 Arrays
Exercises 7.1 Creating and Using Arrays 165
Exercises 7.2 Using LINQ with Arrays 179
Exercises 7.3 Arrays of Structures 191
Exercises 7.4 Two-Dimensional Arrays 214
Programming Projects 226

Chapter 8 Text Files
Exercises 8.1 Managing Text Files 246
Exercises 8.2 StreamReaders, StreamWriters, Structured Exception Handling 255
Exercises 8.3 XML 266
Programming Projects 274

Introduction to Programming Using Visual Basic 11th edition Schneider Solutions Manual

Chapter 9 Additional Controls and Objects
Exercises 9.1 List Boxes and Combo Boxes 284
Exercises 9.2 Eight Additional Controls and Objects 290
Exercises 9.3 Multiple-Form Programs 300
Exercises 9.4 Graphics 317
Programming Projects 333

Chapter 10 Databases
Exercises 10.1 An Introduction to Databases 351
Exercises 10.2 Editing and Designing Databases 372
Programming Projects 375

Chapter 11 Object-Oriented Programming
Exercises 11.1 Classes and Objects 377
Exercises 11.2 Working with Objects 390
Exercises 11.3 Inheritance 415
Programming Projects 430

CHAPTER 2

EXERCISES 2.2 Visual Basic Controls

1. While the mouse cursor is still hovering over the button that was just clicked, the button has a pale blue color. After the cursor leaves the button, the button has a dark blue border.

2. Tabbing to another control, clicking on another control, or pressing Alt plus the access key for another control.

3. Click on the form to make it the selected object.
   Click on the Properties window or Press F4 to activate the Properties window.
   Select the Text property.
   Type "CHECKING ACCOUNT" and press the Enter key.

4. Double-click the TextBox icon in the Toolbox.
   Activate the Properties window, and select the ForeColor property.
   Click on the down-arrow button to the right of the Settings box.
   Click on the Custom tab.
   Click on the desired blue in the palette.
   Move around the Properties window with the up- and down-arrow keys until the Text property is selected.
   Click on the Settings box and then type "PLAY IT, SAM" (without the quotes).
   Click on the text box and then widen it slightly to see the words.

5. Double-click the TextBox icon in the Toolbox.
   Activate the Properties window.
   Select the BackColor property.
   Click on the down-arrow to the right of the Settings box.
   Click on the Custom tab, and then click on the desired yellow in the palette.
   Click on the form to see the yellow text box.

6. Double-click on the TextBox icon in the Toolbox.
   Activate the Properties window, and select the Name property.
   Type "txtGreeting". (The name will appear in the Settings box.)
   Select the Text property.
   Type the requested word, "HELLO".
   Select the Font property.
   Click on the ellipsis to the right of the Settings box.
   Click on the Size box.
   To increase the size of the word, either type the number for the font size (such as "14") or click on a number in the list below the current size.
   Click on Italic in the "Font style" list.
   Click OK.
   If necessary, widen the text box.

7. Double-click on the Label icon in the Toolbox.
Activate the Properties window, and select the AutoSize property.
Set the AutoSize property to False.
Select the Text property and type the requested sentence.
Select the TextAlign property.
Click on the down-arrow button to the right of the Settings box, and click on one of the center rectangles.
Resize the label so that the sentence occupies three lines.

8. Double-click on the TextBox icon in the Toolbox.
Activate the Properties window, and select the Text property.
Type "Visual Basic" and then press the Enter key.
If the text isn't all visible, drag the text box's right sizing handle to the right until all the text can be seen.
Select the ReadOnly property.
Double-click on the ReadOnly property to change it's setting to True. (Or, go to the Settings box, press the down-arrow button, and click on True.)
Select the Font property.
Click on the ellipsis to the right of the Settings box.
In the Font style box, click on Bold.
Click on the *OK* button.
Select the BackColor property and click on the down-arrow button.
Click on the Custom tab.
Click on the desired red in the palette.
Select the ForeColor property and click on the down-arrow button.
Click on the Custom tab.
Click on the white square in the upper-left corner of the palette.

9. Double-click on the TextBox icon in the Toolbox.
Activate the Properties window.
Set the Name property to txtLanguage.
Select the Text property and type "Visual Basic 2017".
Select the Font property and click on the ellipsis to the right of the Settings box.
Scroll up the Font box, and click on Courier New in the Font box.
Click on the *OK* button.
Widen the text box to accommodate its text.

10. Double-click on the Button icon in the Toolbox.
Activate the Properties window, and select the Text property.
Type "PUSH".
Select the BackColor property and click on the down-arrow button.
Click on the Custom tab.
Click on the yellow square (the third square in the fourth column) in the palette.
Double-click on the setting for the (Name) property, type in "btnPush", and press the Enter key.

Introduction to Programming Using Visual Basic 11th edition Schneider Solutions Manual

11. Double-click on the Button icon in the Toolbox.
Activate the Properties window, and select the BackColor property.
Click on the down-arrow button to the right of the Settings box.
Click on the Custom tab, and then click on the white square in upper-left corner of the palette.
Select the Text property and type "PUSH".
Select the Font property, and click on the ellipsis.
Click on *Oblique* in the "Font style" list.
Click on 24 in the Size list.
Click on the *OK* button.
Resize the button.

12. Double-click on the Button icon in the Toolbox.
Activate the Properties window, and select the Text property.
Type "&PUSH".
Select the Font property and click on the ellipsis.
Click on Bold in the "Font Style" box.
Click OK.
Click on the form to see the resulting button.

13. Double-click on the Button icon in the Toolbox.
Activate the Properties window.
Select the Text property and type "PUS&H".
Click on the form to see the resulting button.

14. Double-click on the Label icon in the Toolbox.
Activate the Properties window, and select the Text property.
Type "ALIAS".
Select the ForeColor property and click on the down-arrow button.
Click on the Custom tab.
Click on the desired white in the palette.
Select the BackColor property and click on the down-arrow button.
Click on the Custom tab.
Click on the desired shade of blue in the palette.
Click on the form to see the resulting label.

15. Double-click on the Label icon in the Toolbox.
Activate the Properties window.
Select the Name property and type "lblAKA".
Select the Text property and type "ALIAS".
Select the AutoSize property and set it to False.
Select the Font property and click on the ellipsis.
Click on *Oblique* in the "Font style" list.
Click on the *OK* button.
Select the TextAlign property, click on the down-arrow box to the right of the Settings box, and click on one of the center rectangles.

16. Click on the form to make it the selected object.
Activate the Properties window, and select the Text property.
Type "BALANCE SHEET".
Select the BackColor property and click on the down-arrow button.
Click on the Custom tab.
Click on the desired shade of yellow in the palette.

17. Double-click on the Label icon in the Toolbox.
Activate the Properties window, and select the TextAlign property.
Click on the down-arrow box to the right of the Settings box, and click on one of the rectangles on the right.
Select the AutoSize property and set it to False.
Select the Text property, type "VISUAL BASIC", and press Enter.
If the words " VISUAL BASIC " are on one line, resize the label until the words occupy two lines.

18. In the Solution Explorer, right-click on the file Form1.vb.
Select “Rename” from the drop-down list.
Change the filename from Form1.vb to frmHello.vb.
In the Properties window, change the setting of the Text property from Form1 to Hello World.

19. Double-click on the Label icon in the Toolbox.
Activate the Properties window and set the Text property of the label to PROGRAM.
Select the Font property, and click on the ellipsis to the right of its Settings box.
Click on *Oblique* in the "Font style" list, and click on the *Underline* box.
Click on the *OK* button.

20. Double-click on the Label icon in the Toolbox.
Activate the Properties window, and select the Text property.
Type "ALIAS".
Select the Font property and click on the ellipsis.
Click on Bold in the Font style box.
Click on Courier New in the Font box and press OK.
Click on the form to see the resulting label.

21. Double-click on the ListBox icon in the Toolbox.
Activate the Properties window, and select the BackColor property.
Click on the down-arrow button to the right of the Settings box.
Click on the Custom tab and click on the desired yellow square in the palette.
Click on the form.

22. Double-click on the ListBox icon in the Toolbox.
Activate the Properties window, and select the Visible property.
Double-click on the Visible property to change it's setting to False. (Or, go to the Settings box, press the down-arrow button, and click on False.)

23. In the Solution Explorer window, right click on “Form1.vb” and select *Rename* from the context menu that appears.
Change Form1.vb to frmYellow.vb, and click the *No* button in the box that appears.
Right-click on the form in the Form Designer, and click on Properties in the context menu.
Click on BackColor property in the Properties window.
Click on the down-arrow button in the right part of the Settings box, click on the Custom tab, and click on a yellow square.

24. Double-click on the Button icon in the Toolbox.
Activate the Properties window and set the Text property of the button to BUTTON.
Select the Font property, and click on the ellipsis to the right of its Settings box.
Click on *Bold* in the "Font style" list.
Place a check mark in the small square to the left of the word "Underline" by clicking on the square.
Click on the *OK* button.

25. Begin a new program.
Change the text in the form's title bar to "Dynamic Duo".
Place two buttons on the form.
Position and resize the buttons as shown.
Enter “Batman” as the text of the first button, and enter “Robin” as the text of the second button.
Increase the font size for both buttons to 14.

26. Begin a new program.
Change the text in the form's title bar to "Enter Names".
Place on the form a label, a text box, and a button in the sizes and locations shown. Change the text on the label to "Name".
Change the text on the button to "Enter".
Increase the Font size for the label and the button to 12.

27. Begin a new program.
Change the text in the form's title bar to "Fill in the Blank".
Place a label, a text box, and another label on the form at appropriate locations.
Change the Text setting of the first label to "I’m the king of the" and the Text setting of the second label to "A Quote by Leonardo DiCaprio".

28. Begin a new program.
Change the text in the form's title bar to "Similarity".
Place a big label and a normal-sized label on the form.
Move the small label to the bottom of the form, and move and resize the big label so that it covers most of the top of the form.
Select the normal-sized label.
Change the setting of the Text property to "A Quote".
Select the big label.
Change the setting of the Text property to the specified sentence.
Increase the font size to 12.
Resize and position the labels as needed.

29. Begin a new program.
Change the text in the form's title bar to "Uncle's Advice".
Place five labels and three buttons on the form at appropriate locations.
Change the Text setting of each label as indicated.
Change the settings of the buttons' Text properties to "1", "2", and "3".
Resize and position the labels and buttons.

30. Begin a new program.
Change the text in the form's title bar "3 Rectangles".
Place a label on the form and increase its size to provide plenty of space.
Change the background color to red.
Place a smaller label inside the red label.
Change its background color to some shade of white.
Finally, place a yet smaller label inside the white box.
Change its background color to blue.
Resize and position the labels as needed.

33. 1 **34.** 0

35. Each arrow key moves the text box in the indicated direction.

36. Pressing the right- and left-arrow keys widen and narrow the text box. The up- and down-arrow keys have no effect.

37. Pressing the right- and left-arrow keys widens and narrows the text boxes, buttons, and list boxes in the group of selected controls. The up- and down-arrow keys shorten and lengthen the buttons and list boxes in the group. The arrow keys have no effect on the labels, and only the left- and right-arrow keys affect the text boxes.

38. Each arrow key moves every control in the group in the indicated direction.

39. Drag a label and a list box onto the form.
Click on the label.
Hold down the Ctrl key and click on the list box. (You have now selected a group of two controls.)
In the Properties window, click on the symbol to the left of the Font property.
Click on the Size property, change the setting to 12, and press the Enter key.

(***Alternative:*** Replace the last two lines with the following steps.)

In the Properties window, select the Font property.
Click on the ellipsis button to the right of the Settings box.
Click on 12 in the Size list and click on the *OK* button.

40. The button moves to a comfortable distance from one of the sides of the form.

41. The label is positioned just to the left of the text box, and the middles of the two controls are aligned.

42. The left sides of the buttons are aligned and the two buttons are a comfortable distance apart.

43. *Center* refers to the midpoint horizontally, whereas *middle* refers to the midpoint vertically.

44. Select the four buttons as a group. Click on the Format menu, click on Make Same Size, and then click on Both. Click on the Format menu again, click on Vertical Spacing, and then click on Make Equal.

45. First blue snap line: tops of the two controls are aligned
Purple snap line: texts of the two controls are aligned
Second blue snap line: bottoms of the two controls are aligned

46. The setting toggles between True and False.

47. The setting is cycling through the different available colors.

EXERCISES 2.3 Visual Basic Events

1. The word Hello

2. The word Hello in red letters

3. The word Hello on an orange-colored background

4. The word Hello

5. The text box vanishes.

6. The word Hello on a yellow background

7. The word Hello in green letters

8. The word Hello on a white background

9. The word Hello on a gold background.

10. Nothing, the label cannot be seen.

11. Form1.Text should be Me.Text.

12. The word Hello must be surrounded with quotation marks.

13. Red should be replaced with Color.Red.

14. Replace txtBox with txtBox.Text.

15. Font.Size is a read-only property. The statement `txtOutput.Text = txtBox.Font.Size` is valid since it is reading the value of txtBox.Font.Size. However, `txtBox.Font.Size = 20` is not valid since it is setting the value of txtBox.Font.Size.

16. Me.Color must be replaced by Me.ForeColor or Me.BackColor

```
17. lblTwo.Text = "E.T. phone home."

18. lblTwo.Text = "Play it, Sam."
    lblTwo.ForeColor = Color.Red

19. txtBox.ForeColor = Color.Red
    txtBox.Text = "The stuff that dreams are made of."

20. txtBox.ForeColor = Color.Blue
    txtBox.BackColor = Color.Gold
    txtBox.Text = "Life is like a box of chocolates."

21. txtBox.Enabled = False

22. Me.Text = "Hello World"

23. lblTwo.Visible = False

24. lblName.ForeColor = Color.Red

25. btnOutcome.Enabled = True

26. btnCompute.Focus()

27. txtBoxTwo.Focus()

28. Me.BackColor = Color.White
```

29. The Enter event occurs when a control gets the focus.

30. A control's Leave event occurs when the control loses the focus.

```
31. Private Sub Label1_Click(...) Handles Label1.Click
      lstOutput.Items.Add("Click")
    End Sub

    Private Sub Label1_DoubleClick(...) Handles Label1.DoubleClick
      lstOutput.Items.Add("Double Click")
    End Sub
```

Whenever the DoubleClick event is raised, the Click event is also raised.

```
32. Private Sub Button1_Click(...) Handles Button1.Click
      lstOutput.Items.Add("Click")
    End Sub

33. Private Sub btnLeft_Click(...) Handles btnLeft.Click
      txtBox.Text = "Left Justify"
      txtBox.TextAlign = HorizontalAlignment.Left
    End Sub

    Private Sub btnCenter_Click(...) Handles btnCenter.Click
      txtBox.Text = "Center"
      txtBox.TextAlign = HorizontalAlignment.Center
    End Sub
```

```
    Private Sub btnRight_Click(...) Handles btnRight.Click
      txtBox.Text = "Right Justify"
      txtBox.TextAlign = HorizontalAlignment.Right
    End Sub

34. Private Sub btnSmile_Click(...) Handles btnSmile.Click
      lblFace.Text = ":-)"
    End Sub

    Private Sub btnFrown_Click(...) Handles btnFrown.Click
      lblFace.Text = ":-("
    End Sub

35. Private Sub btnRed_Click(...) Handles btnRed.Click
      txtBox.BackColor = Color.Red
    End Sub

    Private Sub btnBlue_Click(...) Handles btnBlue.Click
      txtBox.BackColor = Color.Blue
    End Sub

    Private Sub btnWhite_Click(...) Handles btnWhite.Click
      txtBox.ForeColor = Color.White
    End Sub

    Private Sub btnYellow_Click(...) Handles btnYellow.Click
      txtBox.ForeColor = Color.Yellow
    End Sub

36. Private Sub txtOne_Enter(...) Handles txtOne.Enter
      txtOne.ForeColor = Color.Red
      txtTwo.ForeColor = Color.Black
      txtThree.ForeColor = Color.Black
    End Sub

    Private Sub txtTwo_Enter(...) Handles txtTwo.Enter
      txtOne.ForeColor = Color.Black
      txtTwo.ForeColor = Color.Red
      txtThree.ForeColor = Color.Black
    End Sub

    Private Sub txtThree_Enter(...) Handles txtThree.Enter
      txtOne.ForeColor = Color.Black
      txtTwo.ForeColor = Color.Black
      txtThree.ForeColor = Color.Red
    End Sub

    Private Sub btnLeft_Click(...) Handles btnLeft.Click
      txtOne.TextAlign = HorizontalAlignment.Left
      txtTwo.TextAlign = HorizontalAlignment.Left
      txtThree.TextAlign = HorizontalAlignment.Left
    End Sub

    Private Sub btnRight_Click(...) Handles btnRight.Click
      txtOne.TextAlign = HorizontalAlignment.Right
      txtTwo.TextAlign = HorizontalAlignment.Right
      txtThree.TextAlign = HorizontalAlignment.Right
    End Sub
```

37.
```
Private Sub txtLife_Enter(...) Handles txtLife.Enter
  txtQuote.Text = "I like life, it's something to do."
End Sub

Private Sub txtFuture_Enter(...) Handles txtFuture.Enter
  txtQuote.Text = "The future isn't what it used to be."
End Sub

Private Sub txtTruth_Enter(...) Handles txtTruth.Enter
  txtQuote.Text = "Tell the truth and run."
End Sub
```

38.
```
Private Sub btnDisable_Click(...) Handles btnDisable.Click
  txtBox.Enabled = False
End Sub

Private Sub btnEnable_Click(...) Handles btnEnable.Click
  txtBox.Enabled = True
  txtBox.Focus()
End Sub
```

39.
```
Private Sub btnOne_Click(...) Handles btnOne.Click
  btnOne.Visible = False
  btnTwo.Visible = True
  btnThree.Visible = True
  btnFour.Visible = True
End Sub

Private Sub btnTwo_Click(...) Handles btnTwo.Click
  btnOne.Visible = True
  btnTwo.Visible = False
  btnThree.Visible = True
  btnFour.Visible = True
End Sub

Private Sub btnThree_Click(...) Handles btnThree.Click
  btnOne.Visible = True
  btnTwo.Visible = True
  btnThree.Visible = False
  btnFour.Visible = True
End Sub

Private Sub btnFour_Click(...) Handles btnFour.Click
  btnOne.Visible = True
  btnTwo.Visible = True
  btnThree.Visible = True
  btnFour.Visible = False
End Sub
```

Introduction to Programming Using Visual Basic 11th edition Schneider Solutions Manual

40.
```
Private Sub txtGreen_Enter(...) Handles txtGreen.Enter
  txtGreen.BackColor = Color.Green
  txtYellow.BackColor = Color.DarkGray
  txtRed.BackColor = Color.DarkGray
End Sub

Private Sub txtYellow_Enter(...) Handles txtYellow.Enter
  txtGreen.BackColor = Color.DarkGray
  txtYellow.BackColor = Color.Yellow
  txtRed.BackColor = Color.DarkGray
End Sub

Private Sub txtRed_Enter(...) Handles txtRed.Enter
  txtGreen.BackColor = Color.DarkGray
  txtYellow.BackColor = Color.DarkGray
  txtRed.BackColor = Color.Red
End Sub
```

41.
```
Private Sub btnVanish_Click(...) Handles btnVanish.Click
  lblFace.Visible = False
End Sub

Private Sub btnReappear_Click(...) Handles btnReappear.Click
  lblFace.Visible = True
End Sub
```

42.
```
Private Sub txtName_Enter(...) Handles txtName.Enter
  lblInstructions.Text = "Enter your full name."
End Sub

Private Sub txtPhone_Enter(...) Handles txtPhone.Enter
  lblInstructions.Text = "Enter your phone number, including area code."
End Sub
```

43.
```
Private Sub btnAny_Click(...) Handles btnOne.Click, btnTwo.Click
  txtOutput.Text = "You just clicked on a button."
End Sub
```

44.
```
Private Sub txtBox1_Click(...) Handles txtBox1.Click
  txtBox2.Text = txtBox1.Text
  txtBox1.Clear()
End Sub

Private Sub txtBox2_Click(...) Handles txtBox2.Click
  txtBox1.Text = txtBox2.Text
  txtBox2.Clear()
End Sub
```

CHAPTER 3

EXERCISES 3.1 Numbers

1. 12 **2.** 49 **3.** .125 **4.** 23 **5.** 8 **6.** –96

7. 2 **8.** 2 **9.** 1 **10.** 1 **11.** 3 **12.** 0

13. Not valid **14.** Not valid **15.** Valid **16.** Not valid **17.** Not valid **18.** Not valid

19. 10 **20.** 14 **21.** 16 **22.** 16 **23.** 9 **24.** 8

25.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  lstOutput.Items.Add((7 * 8) + 5)
End Sub
```

26.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  lstOutput.Items.Add((1 + (2 * 9)) ^ 3)
End Sub
```

27.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  lstOutput.Items.Add(0.055 * 20)
End Sub
```

28.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  lstOutput.Items.Add(15 - 3 * (2 + (3 ^ 4)))
End Sub
```

29.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  lstOutput.Items.Add(17 * (3 + 162))
End Sub
```

30.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  lstOutput.Items.Add((4 + (1 / 2)) - (3 + (5 / 8)))
End Sub
```

31.

	x	y
`Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click`		
`  Dim x, y As Double`	0	0
`  x = 2`	2	0
`  y = 3 * x`	2	6
`  x = y + 5`	11	6
`  lstResults.Items.Clear()`	11	6
`  lstResults.Items.Add(x + 4)`	11	6
`  y = y + 1`	11	7
`End Sub`		

32.

	bal	inter	withDr
`Private Sub btnEvaluate_Click(...) Handles btnEvaluate.Click`			
`  Dim bal, inter, withDr As Decimal`	0	0	0
`  bal = 100`	100	0	0
`  inter = 0.05D`	100	0.05	0
`  withDr = 25`	100	0.05	25
`  bal += inter * bal`	105	0.05	25
`  bal = bal - withDr`	80	0.05	25
`End Sub`			

33. 6 **34.** 24 **35.** 1 8 9 **36.** 10 **37.** 1 64 **38.** 225 **39.** 2 15 **40.** 5 10

41. The third line should read `c = a + b`

42. The third assignment statement does not work. In the last statement, there needs to be an asterisk after the 5. Also, the last statement needs two extra closing parenthesis at the end.

43. The first assignment statement should not contain a comma. The second assignment statement should not contain a dollar sign.

44. The first assignment statement should read `interest = 0.05D`

45. 9W is not a valid variable name.

46. `Round` should be changed to `Math.Round`

47.
```
Dim quantity As Integer = 12
```

48.
```
Dim m As Integer = 2, n As Double = 3
```

49. 10 **50.** 7 **51.** 6 **52.** 8 **53.** 3.128 **54.** -3

55. -3 **56.** 2 **57.** 0 **58.** 3 **59.** 6 **60.** 1

61.
```
cost += 5
```

62.
```
sum *= 2
```

63.
```
cost /= 6
```

64.
```
sum -= 7
```

65.
```
sum ^= 2
```

66.
```
sum += 3
```

67.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim revenue, costs, profit As Decimal
  revenue = 98456
  costs = 45000
  profit = revenue - costs
  lstOutput.Items.Add(profit)
End Sub
```

68.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim costPerShare, numberOfShares, amount As Decimal
  costPerShare = 25.625D
  numberOfShares = 400
  amount = costPerShare * numberOfShares
  lstOutput.Items.Add(amount)
End Sub
```

69.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim price, discountPercent, markdown As Decimal
  price = 19.95D
  discountPercent = 30
  markdown = (discountPercent / 100) * price
  price = price - markdown
  lstOutput.Items.Add(Math.Round(price, 2))
End Sub
```

70.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim fixedCosts, pricePerUnit, costPerUnit, breakEvenPoint As Decimal
  fixedCosts = 5000
  pricePerUnit = 8
  costPerUnit = 6
  breakEvenPoint = fixedCosts / (pricePerUnit - costPerUnit)
  lstOutput.Items.Add(breakEvenPoint)
End Sub
```

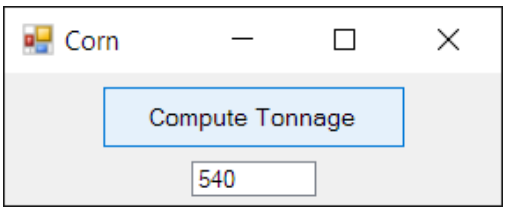
71.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim balance As Decimal
  balance = 100
  balance += 0.05D * balance
  balance += 0.05D * balance
  balance += 0.05D * balance
  lstOutput.Items.Add(Math.Round(balance, 2))
End Sub
```

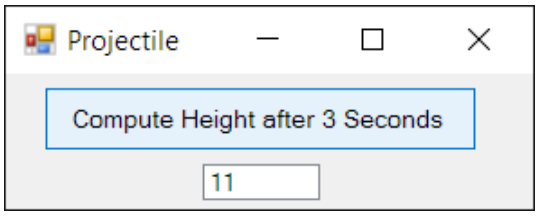
72.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim balance As Decimal
  balance = 100
  balance += 0.05D * balance + 100
  balance += 0.05D * balance + 100
  balance += 0.05D * balance
  lstOutput.Items.Add(Math.Round(balance, 2))
End Sub
```

73.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim purchasePrice As Decimal = 215.50D
  Dim sellingPrice As Decimal = 644.99D
  Dim markup As Decimal = sellingPrice - purchasePrice
  Dim percentageMarkup As Decimal = 100 * (markup / purchasePrice)
  lstOutput.Items.Add(percentageMarkup)
End Sub
```

74.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim purchasePrice As Decimal = 215.50D
  Dim sellingPrice As Decimal = 429.99D
  Dim markup As Decimal = sellingPrice - purchasePrice
  Dim profitMargin As Decimal = 100 * (markup / sellingPrice)
  lstOutput.Items.Add(profitMargin)
End Sub
```

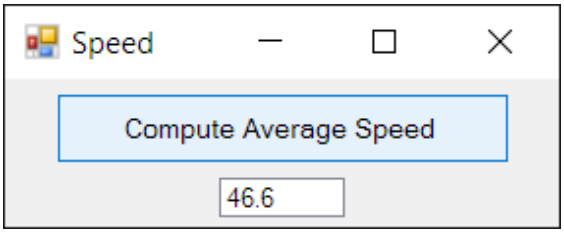
75.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim acres, yieldPerAcre, corn As Double
  acres = 30
  yieldPerAcre = 18
  corn = yieldPerAcre * acres
  lstOutput.Items.Add(corn)
End Sub
```

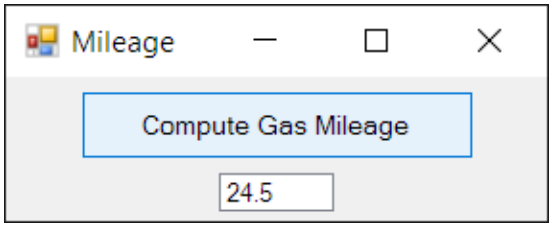




76.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim initialVelocity, initialHeight, height, t As Double
  initialVelocity = 50
  initialHeight = 5
  t = 3
  height = (-16 * t ^ 2) + (initialVelocity * t) + initialHeight
  lstOutput.Items.Add(height)
End Sub
```

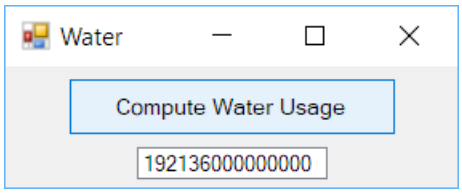
77.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim distance, elapsedTime, averageSpeed As Double
  distance = 233
  elapsedTime = 7 - 2
  averageSpeed = distance / elapsedTime
  lstOutput.Items.Add(averageSpeed)
End Sub
```

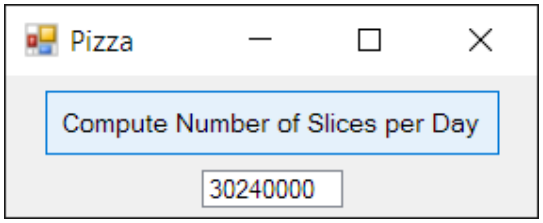




78.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim miles, gallonsUsed, milesPerGallon As Double
  miles = 23695 - 23352
  gallonsUsed = 14
  milesPerGallon = miles / gallonsUsed
  lstOutput.Items.Add(milesPerGallon)
End Sub
```

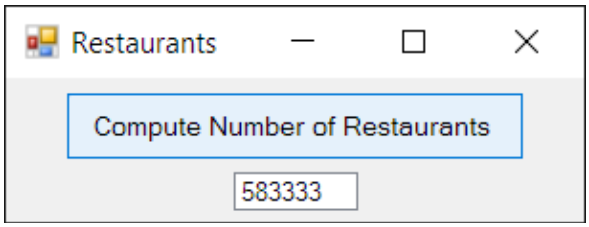
79.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim waterPerPersonPerDay, people, days, waterUsed As Double
  waterPerPersonPerDay = 1600
  people = 329000000
  days = 365
  waterUsed = waterPerPersonPerDay * people * days
  lstOutput.Items.Add(waterUsed)
End Sub
```

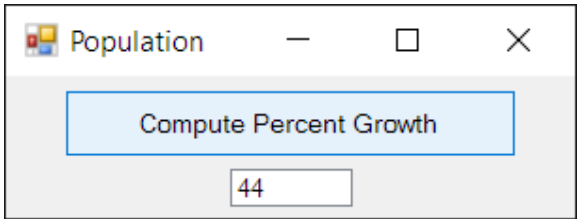




80.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim numPerDay As Integer
  numPerDay = 350 * 60 * 60 * 24
  lstOutput.Items.Add(numPerDay)
End Sub
```

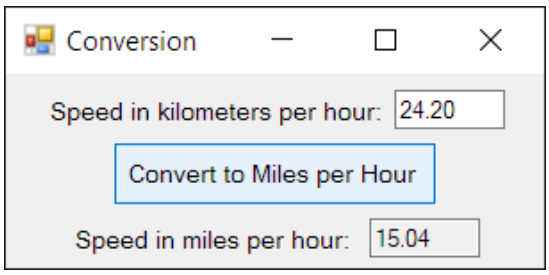
81.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim pizzarias, percent, restaurants As Double
  pizzarias = 70000
  percent = 0.12
  restaurants = pizzarias / percent
  lstOutput.Items.Add(Math.Round(restaurants))
End Sub
```

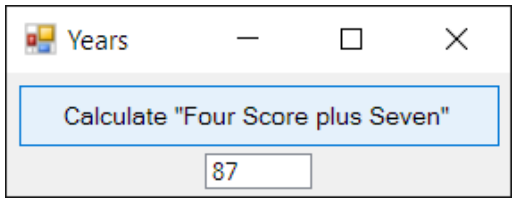




82.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim pop2000, pop2050, percentGrowth As Double
  pop2000 = 281
  pop2050 = 404
  percentGrowth = Math.Round(100 * ((pop2050 - pop2000) / pop2000))
  lstOutput.Items.Add((percentGrowth))
End Sub
```

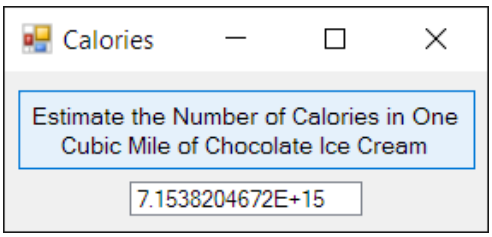
83.
```
Private Sub btnConvert_Click(...) Handles btnConvert.Click
  Dim conversionFactor As Double = 0.6214
  Dim kilometersPerHr, milesPerHr As Double
  kilometersPerHr = CDbl(txtSpeedKil.Text)
  milesPerHr = conversionFactor * kilometersPerHr
  txtSpeedMph.Text = milesPerHr.ToString("N")
End Sub
```

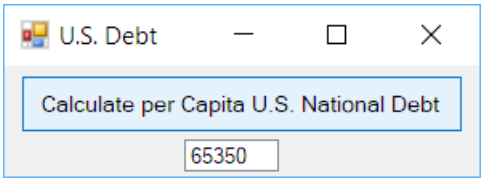





```
84. Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
      lstValue.Items.Add(4 * 20 + 7)
    End Sub
```

```
85. Private Sub btnEstimate_Click(...) Handles btnEstimate.Click
      Dim cal As Double
      cal = (5280 ^ 3) * 48600
      lstCalories.Items.Add(cal)
    End Sub
```






```
86. Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
      Dim debt As Double = 2.15 * 10 ^ 13
      Dim pop As Double = 3.29 * 10 ^ 8
      lstDebt.Items.Add(CInt(debt / pop))
    End Sub
```

EXERCISES 3.2 Strings

```
1. Visual Basic      2. Hello        3. Ernie        4. Bert

5. flute             6. acute        7. 123          8. 8

9. Your age is 21.                  10. Fred has 2 children.

11. A ROSE IS A ROSE IS A ROSE      12. 76 trombones

13. 5.5          14. 3              15. goodbye     16. eighth

17. WALLAWALLA   18. murmur

19. ABC          20. 8 ball         21. 12          22. 9
    2                -1                 MUNICIPALITY    microsoft
    4                5                  city            os
    55 mph           evolutionary       6               5
    STU              10
```

23. 8 (0 through 7) **24.** 7 **25.** True **26.** True

27. The variable *phoneNumber* should be declared as type String, not Double.

28. The sentence in the second line should be surrounded by quotation marks.

29. *End* is a keyword and cannot be used as a variable name.

30. txtBox should be changed to txtBox.Text.

31. The IndexOf method cannot be applied to a number, only a string.

32. A number does not have a Length property, only a String has a Length property.

33. `2 ^ 3` is a Double value and therefore cannot be assigned to the Decimal variable *m.* Also, 4 / 2 is a Double value and cannot be assigned to an Integer variable.

34. `3.45` is a Double value and therefore cannot be assigned to a Decimal variable. The second line should be replaced with either `m = 3.45D` or `m = CDec(3.45).`

35.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim firstName, middleName, lastName As String, yearOfBirth As Integer
  firstName = "Thomas"
  middleName = "Alva"
  lastName = "Edison"
  yearOfBirth = 1847
  txtOutput.Text = firstName & " " & middleName & " " & lastName &
                   ", " & yearOfBirth
End Sub
```

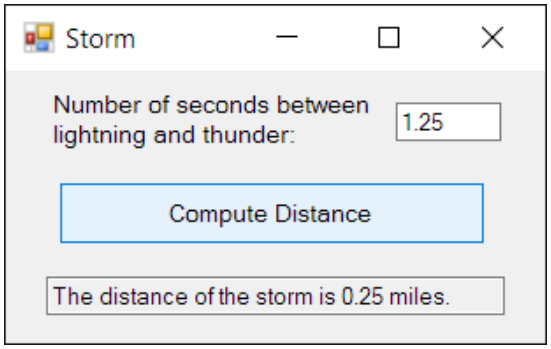
36.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim item As String, regularPrice, discount As Decimal
  item = "ketchup"
  regularPrice = 1.8D
  discount = 0.27D
  txtOutput.Text = (regularPrice - discount) & " is the sale price of " &
                    item & "."
End Sub
```

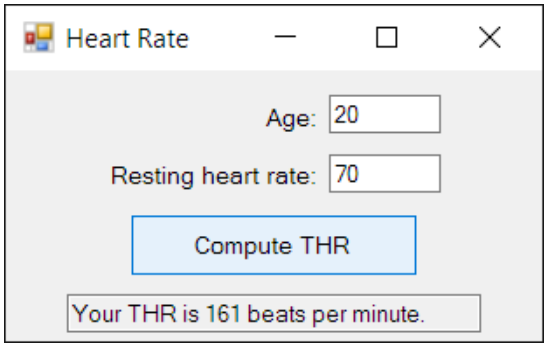
37.
```
Dim str As String  'Place in the Declarations section of the program.
```

38.
```
Dim str As String  'Place near the top of the event procedure.
```

39.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim distance As Double
  distance = CDbl(txtNumSec.Text) / 5
  distance = Math.Round(distance, 2)
  txtOutput.Text = "The distance of the storm is " & distance & " miles."
End Sub
```

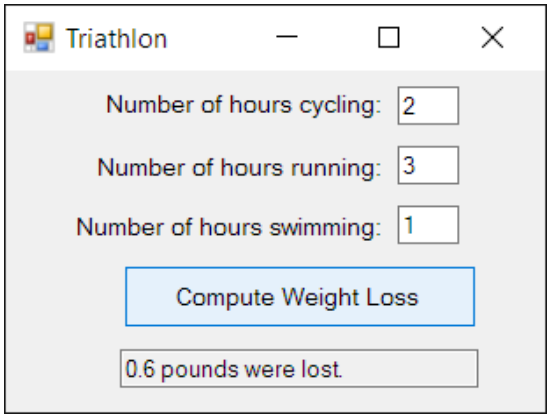
Introduction to Programming Using Visual Basic 11th edition Schneider Solutions Manual

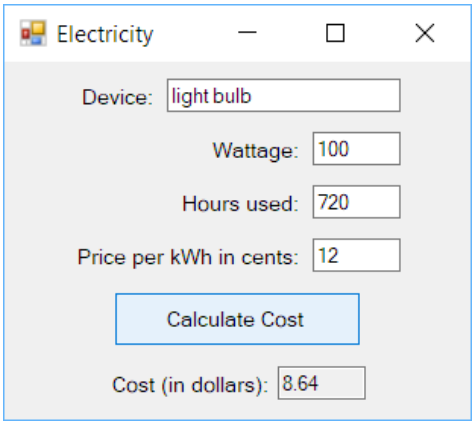




40.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim thr, age, rhr As Double
  age = CDbl(txtAge.Text)
  rhr = CDbl(txtRestHR.Text)
  thr = 0.7 * (220 - age) + 0.3 * rhr
  thr = Math.Round(thr)
  txtTrainHR.Text = "Your THR is " & thr & " beats per minute."
End Sub
```

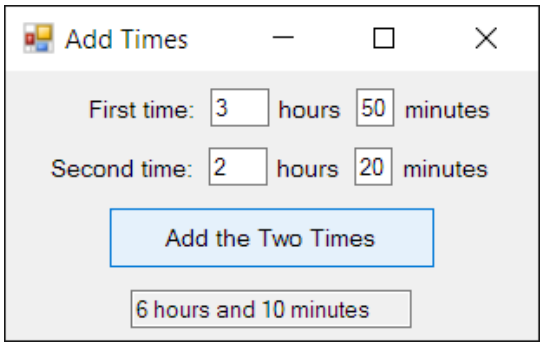
41.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim cycling, running, swimming, pounds As Double
  cycling = CDbl(txtCycle.Text)
  running = CDbl(txtRun.Text)
  swimming = CDbl(txtSwim.Text)
  pounds = ((200 * cycling) + (475 * running) + (275 * swimming)) / 3500
  pounds = Math.Round(pounds, 1)
  txtWtLoss.Text = pounds & " pounds were lost."
End Sub
```

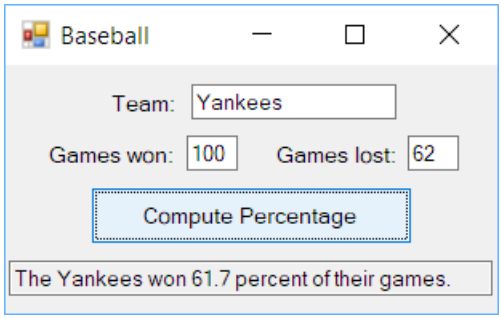




42.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim wattage, hours, price, costInCents, costInDollars As Double
  wattage = CDbl(txtWattage.Text)
  hours = CDbl(txtHoursUsed.Text)
  price = CDbl(txtPricePerKWh.Text)
  costInCents = (wattage * hours * price) / 1000
  costInDollars = Math.Round(costInCents) / 100
  txtCost.Text = CStr(costInDollars)
End Sub
```

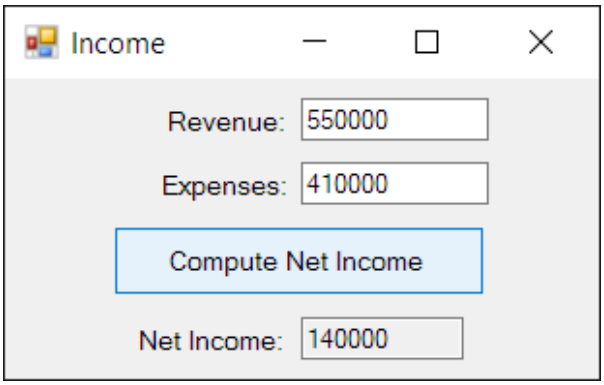
43.
```
Private Sub btnAdd_Click(...) Handles btnAdd.Click
  Dim hr1, hr2, min1, min2, totalMinutes, totalHours As Integer
  hr1 = CInt(txtHours1.Text)
  hr2 = CInt(txtHours2.Text)
  min1 = CInt(txtMin1.Text)
  min2 = CInt(txtMin2.Text)
  totalMinutes = (min1 + min2) Mod 60
  totalHours = hr1 + hr2 + ((min1 + min2) \ 60)
  txtSum.Text = totalHours & " hours and " & totalMinutes & " minutes"
End Sub
```

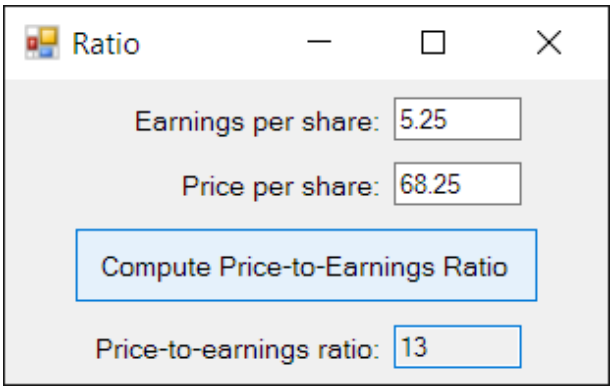




44.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim numberOfGames As Integer
  Dim percent, percentage As Double
  numberOfGames = CInt(txtWon.Text) + CInt(txtLost.Text)
  percent = CDbl(txtWon.Text) / numberOfGames
  percentage = 100 * Math.Round(percent, 5)
  txtPercent.Text = "The " & txtTeam.Text & " won " & percentage &
                    " percent of their games."
End Sub
```

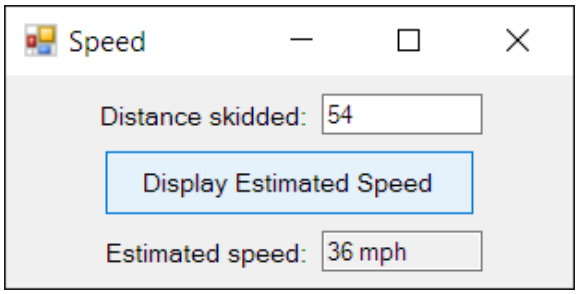
45.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim revenue, expenses, income As Decimal
  revenue = CDec(txtRevenue.Text)
  expenses = CDec(txtExpenses.Text)
  income = revenue - expenses
  txtNetIncome.Text = CStr(income)
End Sub
```

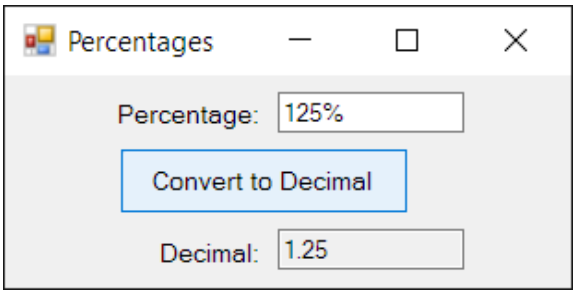




46.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim per As Decimal
  per = CDec(txtPrice.Text) / CDec(txtEarnings.Text)
  txtPER.Text = CStr(per)
End Sub
```

47.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim speed, distance As Double
  distance = CDbl(txtDistanceSkidded.Text)
  speed = Math.Sqrt(24 * distance)
  speed = Math.Round(speed, 2)
  txtEstimatedSpeed.Text = speed & " mph"
End Sub
```

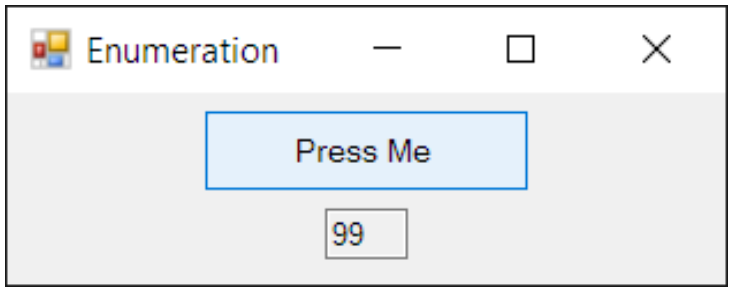


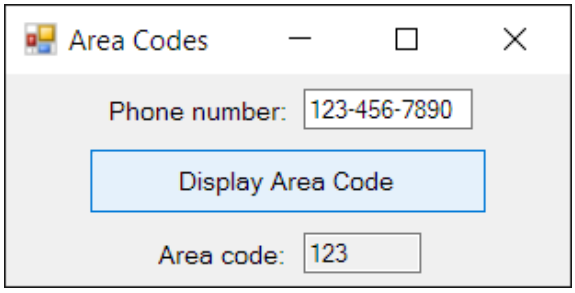


48.
```
Private Sub btnConvert_Click(...) Handles btnConvert.Click
  Dim per As String, num As Double
  per = txtPercent.Text
  per = per.Substring(0, per.Length - 1)
  num = CDbl(per) / 100
  txtNumber.Text = CStr(num)
End Sub
```

49.
```
Dim number As Integer = 100   'in Declarations section

Private Sub btnPressMe_Click(...) Handles btnPressMe.Click
  number = number - 1  'decrease number by 1
  txtOutput.Text = CStr(number)
End Sub
```



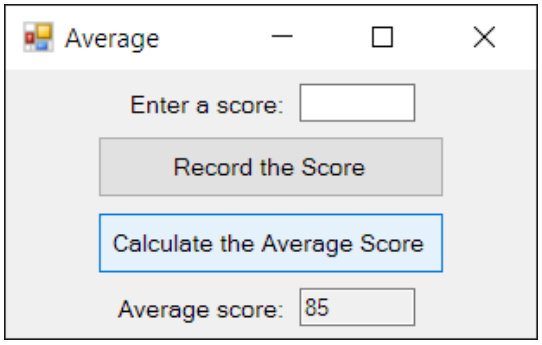


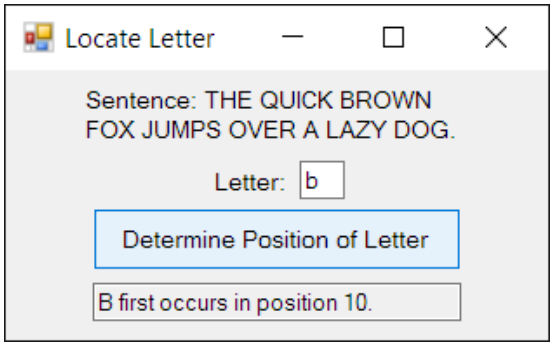
50.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim phoneNumber As String
  phoneNumber = txtPhoneNumber.Text
  txtAreaCode.Text = phoneNumber.Substring(0, 3)
End Sub
```

51.
```
Dim sum As Double   'sum of the scores entered
Dim num As Integer  'number of scores entered

Private Sub btnRecord_Click(...) Handles btnRecord.Click
  num += 1
  sum += CDbl(txtScore.Text)
  txtScore.Clear()
  txtScore.Focus()
End Sub

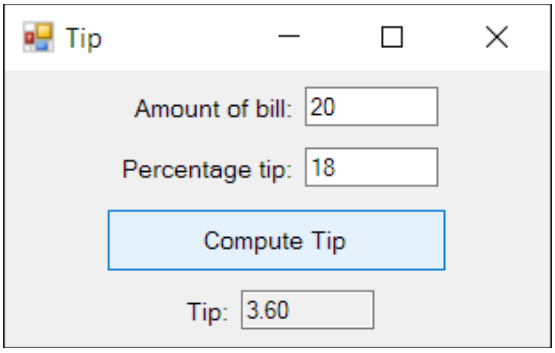
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  txtAverage.Text = CStr(sum / num)
End Sub
```

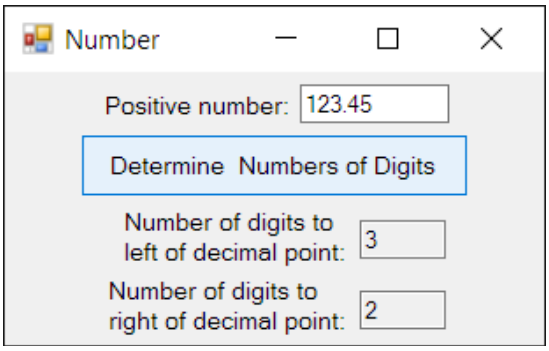




52.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim sentence = "THE QUICK BROWN FOX JUMPS OVER A LAZY DOG"
  Dim letter As String, position As Integer
  letter = txtLetter.Text.ToUpper
  position = sentence.IndexOf(letter)
  txtOutput.Text = letter & " first occurs in position " & position & "."
End Sub
```

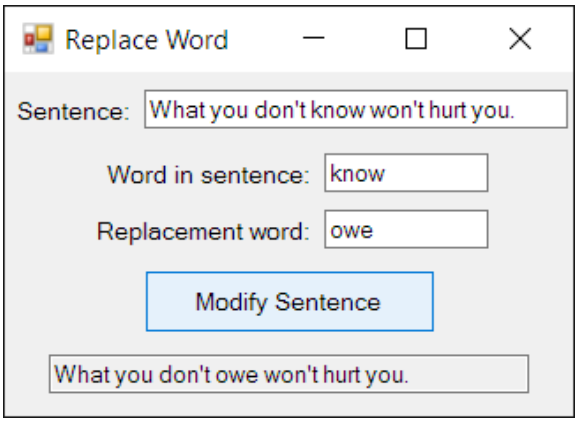
53.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim amount, percentage, tip As Decimal
  amount = CDec(txtAmount.Text)
  percentage = CDec(txtPercentage.Text)
  tip = amount * (percentage / 100)
  txtTip.Text = CStr(Math.Round(tip, 2))
End Sub
```

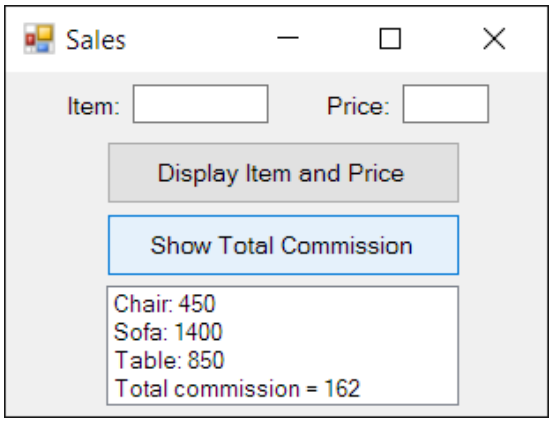




54.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim number As String, position As Integer
  number = txtNumber.Text
  position = number.IndexOf(".")
  txtLeft.Text = CStr(position)
  txtRight.Text = CStr(number.Length - position - 1)
End Sub
```

55.
```
Private Sub btnModifySentence_Click(...) Handles btnModifySentence.Click
  Dim sentence, oldWord, newWord As String
  Dim position As Integer
  sentence = txtSentence.Text
  oldWord = txtOriginalWord.Text
  newWord = txtReplacementWord.Text
  position = sentence.IndexOf(oldWord)
  txtOutput.Text = sentence.Substring(0, position) & newWord &
                   sentence.Substring(position + oldWord.Length)
End Sub
```





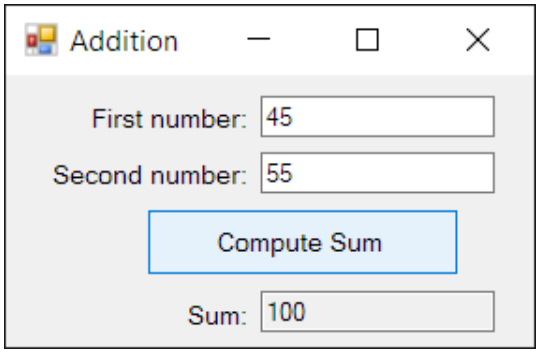
56.
```
Dim totalSales As Decimal

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  lstOutput.Items.Add(txtItem.Text & ": " & txtPrice.Text)
  totalSales += CDec(txtPrice.Text)
  txtItem.Clear()
  txtPrice.Clear()
  txtItem.Focus()
End Sub

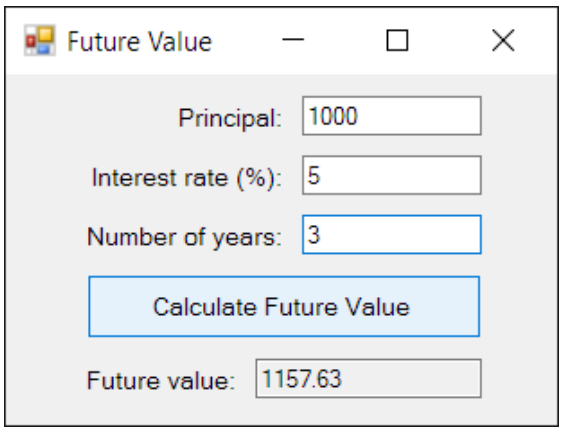
Private Sub btnShow_Click(...) Handles btnShow.Click
  lstOutput.Items.Add("Total commission = " & (0.06 * totalSales))
End Sub
```

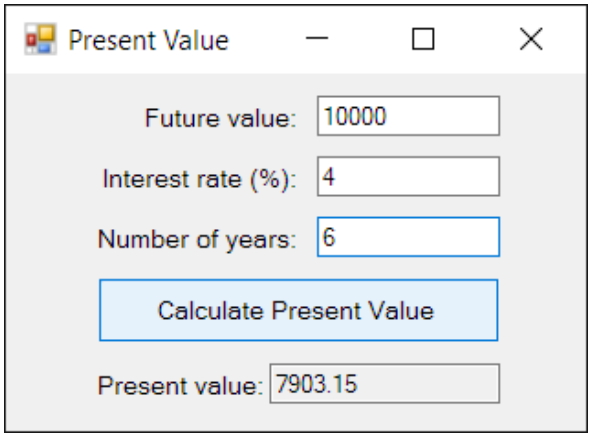
57.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim num1, num2, sum As Double
  num1 = CDbl(txtFirstNum.Text)
  num2 = CDbl(txtSecondNum.Text)
  sum = num1 + num2
  txtSum.Text = CStr(sum)
End Sub

Private Sub txtEitherNum_TextChanged(...) Handles _
            txtFirstNum.TextChanged, txtSecondNum.TextChanged
  txtSum.Clear()
End Sub
```



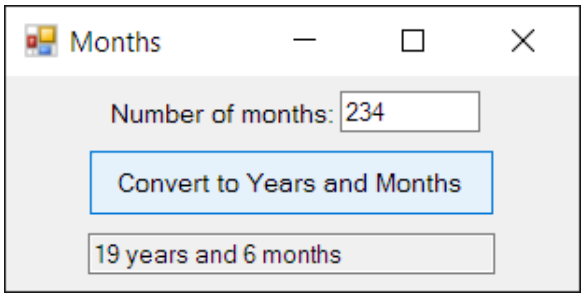
58.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim principal, interestRate, balance As Decimal
  Dim numYears As Integer
  principal = CDec(txtPrincipal.Text)
  interestRate = CDec(txtInterest.Text)
  numYears = CInt(txtYears.Text)
  balance = CDec(principal * (1 + interestRate / 100) ^ numYears))
  txtBalance.Text = balance.ToString("C")
End Sub
```

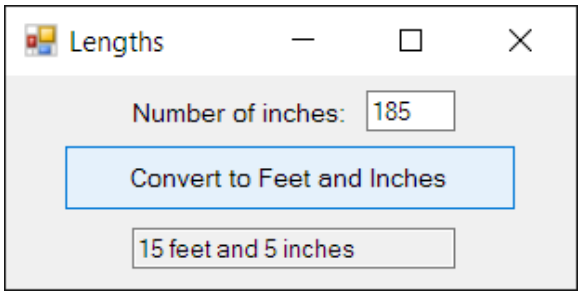




59.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim futureValue, interestRate, presentValue As Decimal
  Dim numYears As Integer
  futureValue = CDec(txtFutureValue.Text)
  interestRate = CDec(txtInterest.Text)
  numYears = CInt(txtYears.Text)
  presentValue = CDec(futureValue / (1 + interestRate / 100) ^ numYears)
  txtPresentValue.Text = CStr(Math.Round(presentValue, 2))
End Sub
```

```
60. Private Sub btnConvert_Click(...) Handles btnConvert.Click
      Dim numMonths As Integer = CInt(txtMonths.Text)
      Dim years As Integer = numMonths \ 12
      Dim remainingMonths = numMonths Mod 12
      txtAnswer.Text = years.ToString("N0") & " years and " &
                       remainingMonths & " months"
    End Sub
```






```
61. Private Sub btnConvert_Click(...) Handles btnConvert.Click
      Dim feet, inches As Integer
      inches = CInt(txtInches.Text)
      feet = inches \ 12
      inches = inches Mod 12
      txtFeetAndInches.Text = feet & " feet and " & inches & " inches"
    End Sub

62. Private Sub btnFind_Click(...) Handles btnFind.Click
      Dim taxBracket As Decimal = CDec(txtTaxBracket.Text)
      Dim bondRate As Decimal = CDec(txtMunIntRate.Text)
      Dim equivCDrate As Decimal = bondRate / (1 - taxBracket)
      txtCDIntRate.Text = CStr(Math.Round(equivCDrate, 3))
    End Sub
```

EXERCISES 3.3 Input and Output

```
1. 1,235      2. -12.346      3. 1,234.0      4. 12,345.00

5. 0.0      6. 0.1      7. -0.67      8. 1.20      9. 12,346.000      10. 12

11. 12          12. $1,234.50      13. $12,346      14. ($1,234,567.00)

15. ($0.23)      16. $3,200.00      17. $0.80      18. 4 %

19. $0.08      20. -5.000 %      21. 100.00 %      22. 1.00 %

23. 66.67 %      24. 75.0 %      25. Pay to France $27,267,622.00

26. Manhattan was purchased for $24.00

27. 25.6 % of the U.S. population 25+ years old are college graduates.

28. 1,711,500 degrees were conferred.

29. The likelihood of Heads is 50 %  30. Pi = 3.1416

31. 10/23/2021                      32. Sunday, June 21, 2020

33. Thursday, November 26, 2020      34. 1/1/2015
```

35. 10/2/2021 36. 2/9/2020 37. 4/5/2019 38. 11/2/2019

39. 29 40. 366 41. 5 42. 2

43. 1 44. 2020 45. 2022 46. 1, 2, 3

47. You might win 360 dollars. 48. Untied 49. Hello John Jones

50. At the current interest rate, money will double in 18 years.

51. $106.00 52. You owe $900,000.00 in estate taxes.

53.
```
12345678901234567890123456789012345678901234567890
Mountain      Place      Ht (ft)
K2            Kashmir    28,250
```

54.
```
1234567890123456789012345678901234567890
    College    Mascot
Univ. of MD    Terrapins
       Duke    Blue Devils
```

55.
```
12345678901234567890123456789012345678901234567890
Element Weight  Percent
Oxygen  97.5    65.0 %
Carbon  27.0    18.0 %
```

56.
```
12345678901234567890123456789012345678901234567890
                Tuition
   College      & Fees
  Stanford      $46,300
   Harvard      $45,278
```

57. The statement `n += 1` is not valid since the value of a named constant cannot be changed.

58. The statement `n = n.ToUpper` is not valid since the value of a constant cannot be changed.

59. The second line should use CDbl to convert the right-hand side to type Double.

60. There needs to be a string inside the InputBox() call (i.e., the "Prompt" argument is required).

61. (123456).ToString("N") is a string and therefore cannot be assigned to a numeric variable.

62. Must remove the dollar sign.

63. You must insert `.Show` after the word `MessageBox`.

64. 1776 should be surrounded with quote marks

65. 000 66. LLLLLLLLLL 67. LLL000

68. 0LLL000 69. 0-00-000000-& 70. LL

71.
```
MessageBox.Show("First solve the problem. Then write the code.",
                "Good Advice")
```

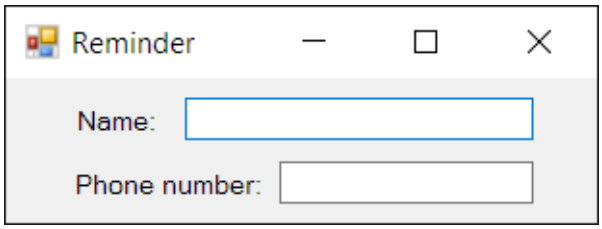
72.
```
MessageBox.Show("You can't steal second base and keep one foot on first.",
                "Taking Risks Proverb")
```

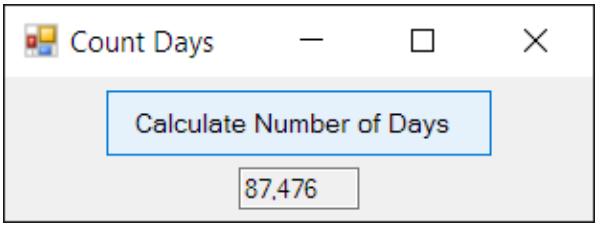
Introduction to Programming Using Visual Basic 11th edition Schneider Solutions Manual

73.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Inflation
  Dim begOfYearCost, endOfYearCost As Decimal
  Dim percentIncrease As Decimal
  begOfYearCost = 200
  endOfYearCost = CDec(InputBox("Enter cost at the end of the year:"))
  percentIncrease = (endOfYearCost - begOfYearCost) / begOfYearCost
  txtOutput.Text = "The increase in cost for the year is " &
                   percentIncrease.ToString("P") & "."
End Sub
```

74.
```
Private Sub btnDisplayReport_Click(..) Handles btnDisplayReport.Click
  'Report the amount of money raised in a walk-a-thon
  Dim pledge, miles As Decimal
  pledge = CDec(InputBox("Enter amount pledged per mile:", "Walk-A-Thon"))
  miles = CDec (InputBox("Enter number of miles walked:", "Walk-A-Thon"))
  txtReport.Text = "Collect " & (pledge * miles).ToString("C") &
                   " from this sponsor."
End Sub
```

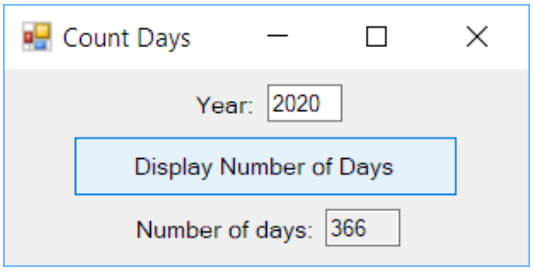
75.
```
Private Sub txtPhoneNumber_Enter(...) Handles txtPhoneNumber.Enter
  'Reminder
  MessageBox.Show("Be sure to include the area code!", "Reminder")
End Sub
```

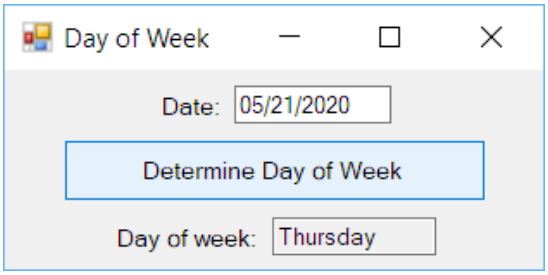




76.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  'Days since Declaration of Independence ratified
  Dim dateDOC As Date = #7/4/1776#
  Dim numDays As Double = DateDiff(DateInterval.Day, dateDOC, Today)
  txtNumDays.Text = numDays.ToString("N0")
End Sub
```

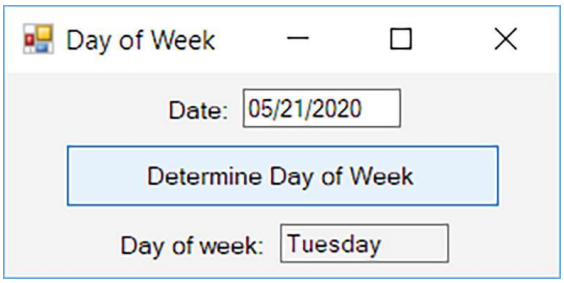
77.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Length of year
  Dim firstDayOfYr, firstDayOfNextYr As Date
  Dim numDays As Double
  firstDayOfYr = CDate("1/1/" & mtbYear.Text)
  firstDayOfNextYr = firstDayOfYr.AddYears(1)
  numDays = DateDiff(DateInterval.Day, firstDayOfYr, firstDayOfNextYr)
  txtNumDays.Text = CStr(numDays)
End Sub
```

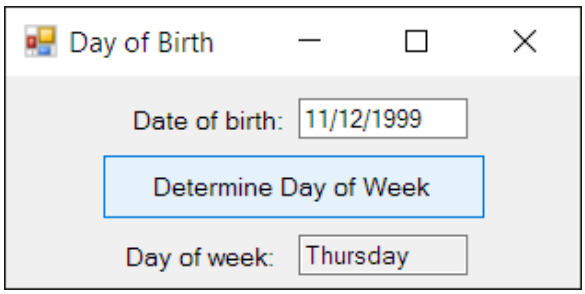




78.
```
Private Sub Determine_Click(...) Handles btnDetermine.Click
  'Day of week
  Dim dt As Date = CDate(mtbDate.Text)
  Dim fullDate As String = dt.ToString("D")
  Dim position As Integer = fullDate.IndexOf(",")
  Dim dayOfWeek As String = fullDate.Substring(0, position)
  txtDayOfWeek.Text = dayOfWeek
End Sub
```

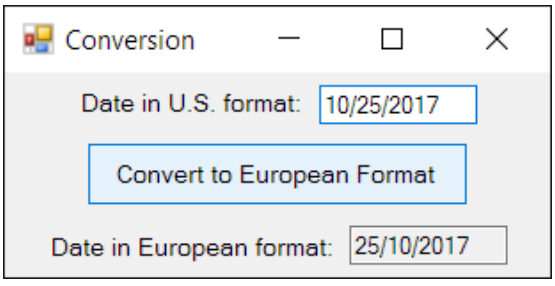
79.
```
Private Sub Determine_Click(...) Handles btnDetermine.Click
  'Day of week
  Dim dt As Date = CDate(mtbDate.Text)
  Dim laterDate = dt.AddYears(10)
  Dim fullDate As String = laterDate.ToString("D")
  Dim position As Integer = fullDate.IndexOf(",")
  Dim dayOfWeek As String = fullDate.Substring(0, position)
  txtDayOfWeek.Text = dayOfWeek
End Sub
```

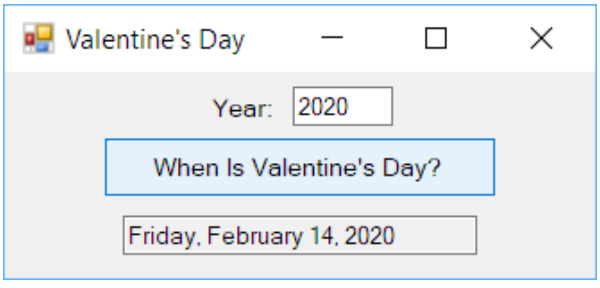




80.
```
Private Sub Determine_Click(...) Handles btnDetermine.Click
  'Day of birth
  Dim dt As Date = CDate(mtbDate.Text)
  Dim turn21Date = dt.AddYears(21)
  Dim fullDate As String = turn21Date.ToString("D")
  Dim position As Integer = fullDate.IndexOf(",")
  Dim dayOfWeek As String = fullDate.Substring(0, position)
  txtDayOfWeek.Text = dayOfWeek
End Sub
```

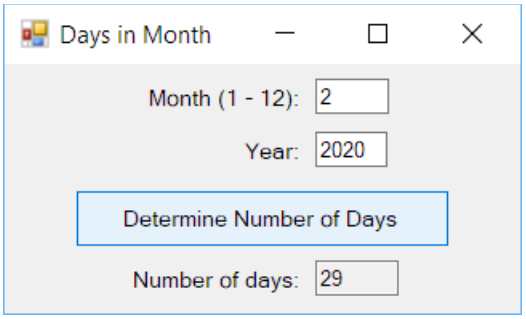
81.
```
Private Sub btnConvert_Click(...) Handles btnConvert.Click
  'Convert date formats
  Dim dt As Date
  dt = CDate(txtUS.Text)
  txtEurope.Text = dt.Day & "/" & dt.Month & "/" & dt.Year
End Sub
```



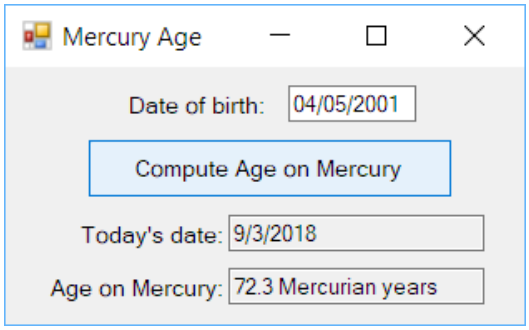


82.
```
Private Sub btnWhen_Click(...) Handles btnWhen.Click
  'Valentine's day
  Dim yr As String
  Dim dt As Date
  yr = txtYear.Text
  dt = CDate("2/14/" & yr)
  txtDate.Text = dt.ToString("D")
End Sub
```

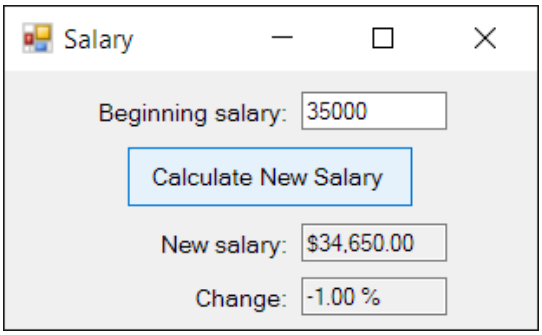
83.
```
Private Sub Determine_Click(...) Handles btnDetermine.Click
  'Length of month
  Dim month, yr As Integer   'month given as 1 through 12
  Dim dt, dt2 As Date
  Dim numDays As Double
  month = CInt(txtMonth.Text)
  yr = CInt(mtbYear.Text)
  dt = CDate(month & "/1/" & yr)
  dt2 = dt.AddMonths(1)
  numDays = DateDiff(DateInterval.Day, dt, dt2)
  txtNumDays.Text = CStr(numDays)
End Sub
```

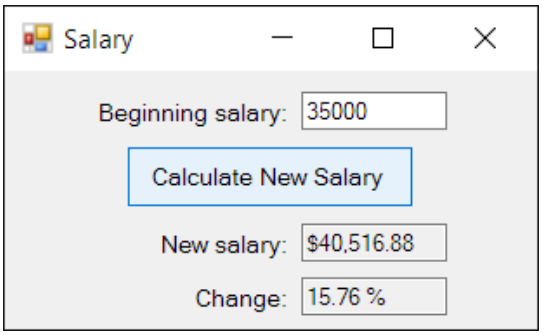


84.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  'Age on Mercury
  Dim dob As Date = CDate(mtbDayOfBirth.Text)
  Dim ageOnEarthInDays, ageOnMercuryInYears As Double
  txtToday.Text = Today.ToString("d")
  ageOnEarthInDays = DateDiff(DateInterval.Day, dob, Today)
  ageOnMercuryInYears = ageOnEarthInDays / 88
  txtAgeOnMercury.Text = ageOnMercuryInYears.ToString("N1") & " Mercurian years"
End Sub
```



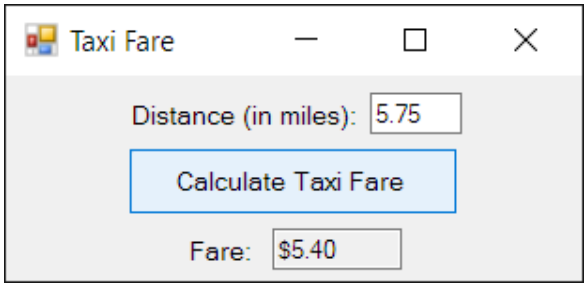
85.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  'Change in salary
  Dim begSalary, salary As Decimal
  begSalary = CDec(txtBeginningSalary.Text)
  salary = begSalary + (0.1D * begSalary)
  salary = salary - (0.1D * salary)
  txtNewSalary.Text = salary.ToString("C")
  txtChange.Text = ((salary - begSalary) / begSalary).ToString("P")
End Sub
```





86.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  'Change in salary
  Dim begSalary, salary As Decimal
  begSalary = CDec(txtBeginningSalary.Text)
  salary = begSalary + 0.05D * begSalary
  salary = salary + 0.05D * salary
  salary = salary + 0.05D * salary
  txtNewSalary.Text = salary.ToString("C")
  txtChange.Text = ((salary - begSalary) / begSalary).ToString("P")
End Sub
```

87.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  'Taxi fare
  Dim distance As Double
  distance = CDbl(txtDistance.Text)
  txtFare.Text = (0.8 + 0.2 * (Int(4 * distance))).ToString("C")
End Sub
```



```
88. Private Sub txtCalculate_Click(...) Handles txtCalculate.Click
      'Marketing terms
      Dim purchasePrice As Decimal = CDec(txtPurchasePrice.Text)
      Dim sellingPrice As Decimal = CDec(txtSellingPrice.Text)
      Dim markup, percentageMarkup, profitMargin As Decimal
      markup = sellingPrice - purchasePrice
      percentageMarkup = markup / purchasePrice
      profitMargin = markup / sellingPrice
      txtMarkup.Text = markup.ToString("C")
      txtPercentageMarkup.Text = percentageMarkup.ToString("P")
      txtProfitMargin.Text = profitMargin.ToString("P")
    End Sub
```

