STUDENT SOLUTIONS MANUAL

to accompany
An Introduction to Programming
Using Visual Basic 2010, 8th Edition

by David I. Schneider

CONTENTS

Chapter Comments

Chapter 2 Visual Basic Controls and Events

Chapter 3 Variables, Input, and Output

Chapter 4 Decisions

Chapter 5 General Procedures

Chapter 6 Repetition

Chapter 7 Arrays

Chapter 8 Text Files

Chapter 9 Additional Controls and Objects

Chapter 10 Databases

Chapter 11 Object-Oriented Programming

Chapter 12 Web Applications

Chapter Comments

Chapter 1

1. Your instructor might skip this chapter. If so, I recommend that you take a quick look at two items from the chapter. Look at the last question and answer on page 3 to see how the appearance of a program varies with the version of Windows being used. Also, look at the discussion of "Displaying File Extensions" on pages 5 and 6. I recommend that you configure Windows to show all file extensions.

Chapter 2

1. On page 44 you are asked to run a program that was downloaded from the Pearson website for the book. All of the programs appearing in Examples and Case Studies can be downloaded from that website. There is never any need for you to manually type in the code for a program in this textbook. The website also contains all the text files, databases, and pictures needed for the exercises.

   All of these files are contained in the folders “Ch02”, “Ch03”, “Ch04”, and so on. Each chapter file contains a subfolder named “Text_Files_for_Exercises” which contains the text files needed for that chapter’s exercises. The folder “Ch09” has a subfolder named “Pictures” that contains picture files. The folders “Ch10” and "Ch12" have a subfolder named “Databases” containing all the databases needed for the exercises.

   Each program is contained in a folder with a name in the form *chapter–section–example number*. For instance, the program in Chapter 3, Section 1, Example 2 is contained in the folder “3-1-2”. Many of the programs make use of a text file in the subfolder *Debug* of the program’s *bin* subfolder.

2. At the top of page 44, we explain our use of ellipses (...) as place holders for the phrase "**ByVal sender As System.Object, ByVal e As System.EventArgs".** A program will not run when the phrase is replaced with an ellipsis. However, a program *will* run if the phrase is simply deleted. Therefore, if you use Ctrl+C to copy code from this Student Solutions Manual and paste it into your own program, the code will execute after you delete the ellipses.

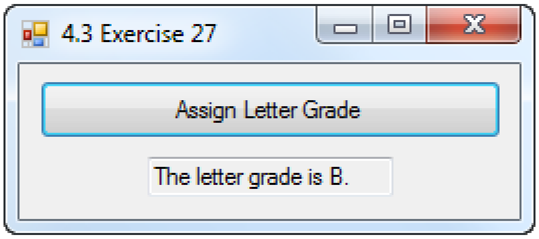
3. Every program you write requires use of the Properties window. By default, the Properties window is docked to the right side of the IDE. You might find the Properties window easier to use if you undock it while setting properties. If you double-click on the Properties window’s title bar, the window will become undocked and larger. After you have finished using the Properties window, right-click on its title bar and click on *Dock*. ***Note:*** This process also can be used to undock (and redock) the Toolbar and the Solution Explorer window.

Chapter 12

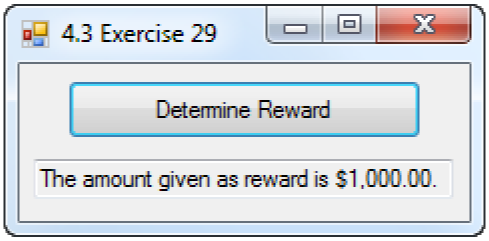
1. The programs in the chapter are not created with Visual Basic. Most people will use Visual Web Developer that is contained on the DVD packaged with this book. However, if you have a complete version of Visual Studio, you do not have to install Visual Web Developer. The Visual Studio *File* menu contains the items *New Web Site* and *Open Web Site* that you can use to create and access Web programs.
2. Be sure to read the solution to the practice problem in Section 12.3. It shows you how to save a lot of time when writing the programs for the exercise set.

27.

```
Private Sub btnAssign_Click(...) Handles btnAssign.Click
  Dim score As Integer, letterGrade As String
  score = CInt(InputBox("What is the score?"))
  Select Case score
    Case 90 To 100
      letterGrade = "A"
    Case 80 To 89
      letterGrade = "B"
    Case 70 To 79
      letterGrade = "C"
    Case 60 To 69
      letterGrade = "D"
    Case 0 To 59
      letterGrade = "F"
    Case Else
      letterGrade = "Invalid"
  End Select
  txtOutput.Text = "The letter grade is " & letterGrade & "."
End Sub
```



The letter grade is B.

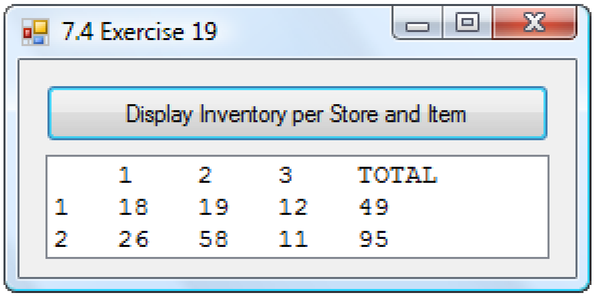


The amount given as reward is $1,000.00.

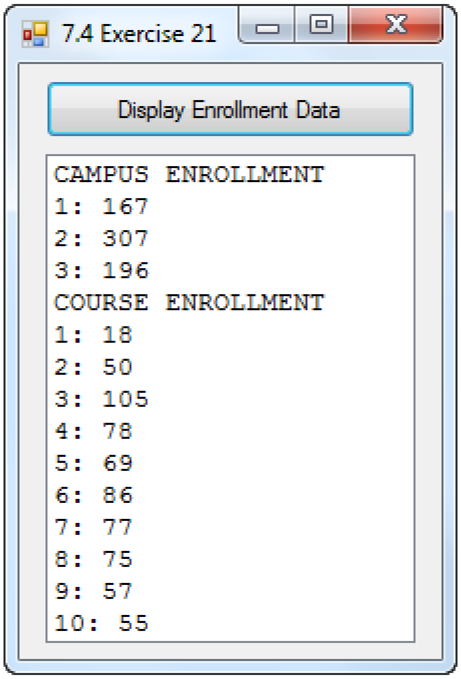
29.

```
Private Sub btnDescribe_Click(...) Handles btnDetermine.Click
  Dim amountRecovered, payment As Double
  amountRecovered = CDbl(InputBox("How much was recovered?"))
  Select Case amountRecovered
    Case Is <= 75000
      payment = 0.1 * amountRecovered
    Case Is <= 100000
      payment = 7500 + 0.05 * (amountRecovered - 75000)
    Case Is > 100000
      payment = 8750 + 0.01 * (amountRecovered - 100000)
      If payment > 50000 Then
        payment = 50000
      End If
  End Select
  txtOutput.Text = "The amount given as reward is " &
                   FormatCurrency(payment) & "."
End Sub
```

```
19. Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
      'Display a company's inventory from its two stores
      Dim inventory(,) As Integer = {{25, 64, 23}, {30, 82, 19}}
      Dim sales(,) As Integer = {{7, 45, 11}, {4, 24, 8}}
      Dim total(2) As Integer
      'Adjust the inventory values to reflect today's sales
      For store As Integer = 1 To 2
        For item As Integer = 1 To 3
          inventory(store - 1, item - 1) =
              inventory(store - 1, item - 1) - sales(store - 1, item - 1)
          'Accumulate the total inventory per store
          total(store) += inventory(store - 1, item - 1)
        Next
      Next
      'Display the store's inventory and totals
      lstOutput.Items.Add("    1    2    3    TOTAL")
      For store As Integer = 1 To 2
        lstOutput.Items.Add(store & "   " & inventory(store - 1, 0) &
                          "   " & inventory(store - 1, 1) & "   " &
                          inventory(store - 1, 2) & "   " & total(store))
      Next
    End Sub
```



```
     1    2    3    TOTAL
1    18   19   12   49
2    26   58   11   95
```



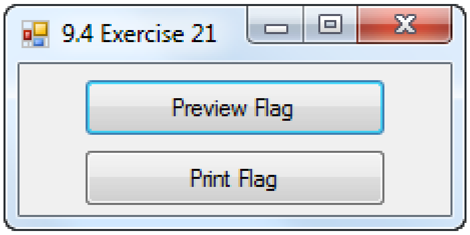
```
CAMPUS ENROLLMENT
1: 167
2: 307
3: 196
COURSE ENROLLMENT
1: 18
2: 50
3: 105
4: 78
5: 69
6: 86
7: 77
8: 75
9: 57
10: 55
```

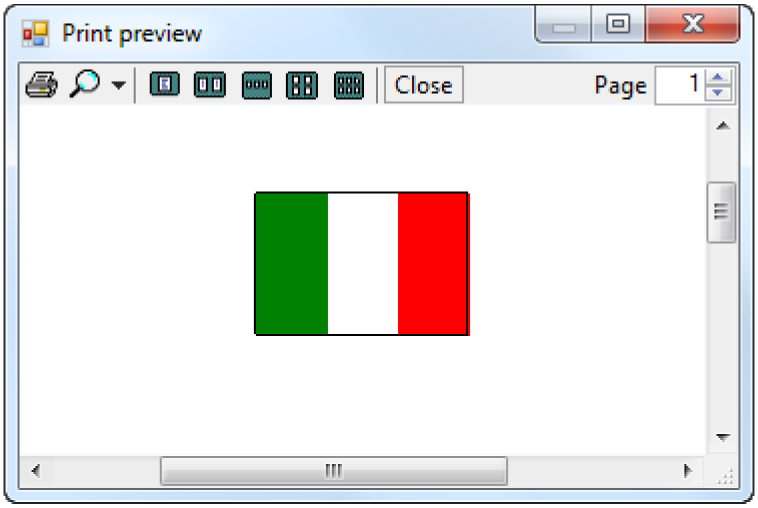
21.

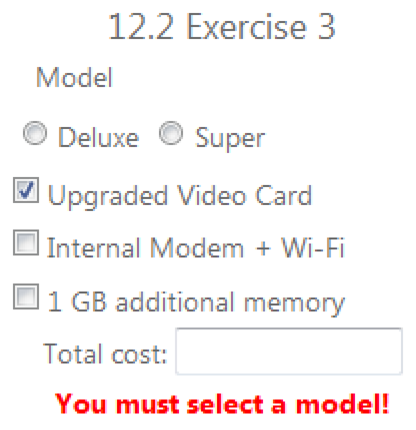
```
Private Sub btnPrint_Click(...)  Handles btnPrint.Click
  PrintDocument1.Print()
End Sub

Private Sub PrintDocument1_PrintPage...) Handles _
                PrintDocument1.PrintPage
  Dim gr As Graphics = e.Graphics
  Dim br() As Brush = {Brushes.Green, Brushes.White, Brushes.Red}
  For i As Integer = 0 To 2
    gr.FillRectangle(br(i), 300 + i * 50, 200, 50, 99)
  Next
  gr.DrawLine(Pens.Black, 300, 200, 448, 200)  'top border
  gr.DrawLine(Pens.Black, 300, 200, 300, 298)  'left border
  gr.DrawLine(Pens.Black, 300, 298, 448, 298)  'bottom border
  gr.DrawLine(Pens.Black, 448, 200, 448, 298)  'right border
End Sub

Private Sub btnPreview_Click(...) Handles btnPreview.Click
  PrintPreviewDialog1.Document = PrintDocument1
  PrintPreviewDialog1.ShowDialog()
End Sub
```







12.2 Exercise 3

Model

○ Deluxe ○ Super

☑ Upgraded Video Card

☐ Internal Modem + Wi-Fi

☐ 1 GB additional memory

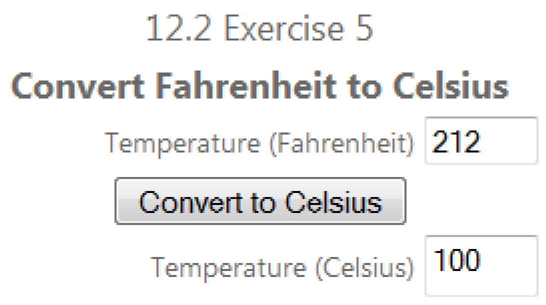
Total cost:

You must select a model!

5.

```
Protected Sub btnConvert_Click(...) Handles btnConvert.Click
  Dim fahrenheitTemp, celsiusTemp As Double
  fahrenheitTemp = CDbl(txtTempF.Text)
  celsiusTemp = FtoC(fahrenheitTemp)
  txtTempC.Text = CStr(celsiusTemp)
End Sub

Function FtoC(ByVal t As Double) As Double
  Return (5 / 9) * (t - 32)
End Function
```



12.2 Exercise 5

Convert Fahrenheit to Celsius

Temperature (Fahrenheit) 212

Temperature (Celsius) 100