

```

Sub ResetInput()
    'reset all text entry fields except date
    txtToFrom.Clear()
    txtAmount.Clear()
    txtMemo.Clear()
    If isChecked Then
        'make txtNum text box reflect next check number
        txtNum.Text = CStr(lastCkNum + 1)
    Else
        'make txtNum text box reflect next deposit slip number
        txtNum.Text = CStr(lastDpNum + 1)
    End If
    'set focus on To/From text box for the next entry
    txtToFrom.Focus()
End Sub

Private Sub btnReport_Click(...) Handles btnReport.Click
    If IO.File.Exists(TRANS_FILE) Then
        Dim transFileContents() As String = IO.File.ReadAllLines(TRANS_FILE)
        Dim query = From trans In transFileContents
            Let data = trans.Split(",")
            Let transDate = CDate(data(7))
            Let number = FormNumber(data(0), data(3), data(4))
            Let toFrom = data(1)
            Let Memo = data(6)
            Let Amount = FormatCurrency(data(5))
            Let Balance = FormatCurrency(data(2))
            Select transDate, number, toFrom, Memo, Amount, Balance
        dgvTransactions.DataSource = query.ToList
        dgvTransactions.CurrentCell = Nothing
        dgvTransactions.Columns("transDate").HeaderText = "Transaction Date"
        dgvTransactions.Columns("number").HeaderText = "Description"
        dgvTransactions.Columns("toFrom").HeaderText = "Recipient or Source"
    Else
        MessageBox.Show("There are no transactions to report.")
    End If
End Sub

Function FormNumber(ByVal type As String, ByVal checkNumber As String,
    ByVal depositNumber As String) As String
    If type = "Check" Then
        Return "Check #" & checkNumber
    Else
        Return "Deposit #" & depositNumber
    End If
End Function

Private Sub btnQuit_Click(...) Handles btnQuit.Click
    Me.Close() 'exit the program
End Sub

```

## CHAPTER 8 SUMMARY

1. The *IO.File.WriteAllLines* method copies an array to a text file.
2. When data are stored in text files with the fields of each record separated by commas, LINQ can be used to sort, search, and reorganize the data with a little help from the *Split* method.

3. Arrays and queries can be combined with the set methods *Concat*, *Union*, *Intersect*, and *Except*.
4. When text files are opened, the program must specify whether they will be read using a *StreamReader* or written using a *StreamWriter*. Files used for input are specified with the *IO.File.ReadText* method. Output files can be created (*IO.File.CreateText*) or just added to (*IO.File.AppendText*). A line of data is written to a file with the *WriteLine* method and read from a file with the *ReadLine* method.
5. *Structured exception handling* can reduce the likelihood that a program will crash. If an exception occurs while the code in the Try block is executing, execution branches to the code in a Catch block that alerts the user of an error and provides a workaround. The Finally block contains code that executes regardless of whether an exception occurs.
6. *XML files* are text files of a special format that is popular for data transmitted over the Internet.

## CHAPTER 8 PROGRAMMING PROJECTS

1. The file *ALE.txt* contains the information shown in Table 8.5. Write a program to use the file to produce a text file containing the information in Table 8.6 in which the baseball teams are in descending order by the percentage of games won.

**TABLE 8.5** American League East games won and lost in 2009.

Team	Won	Lost
Baltimore	64	98
Boston	95	67
New York	103	59
Tampa Bay	84	78
Toronto	75	87

**TABLE 8.6** Final 2009 American League East standings.

Team	W	L	Pct
New York	103	59	0.636
Boston	95	67	0.586
Tampa Bay	84	78	0.519
Toronto	75	87	0.463
Baltimore	64	98	0.385

2. The file *Senate110.txt* contains the members of the 110th U.S. Senate—that is, the Senate prior to the 2008 election. Each record of the file consists of three fields—name, state, and party affiliation.<sup>2</sup> Some records in the file are as follows:

```
Richard Selby,Alabama,R
Joseph Lieberman,Connecticut,I
Barack Obama,Illinois,D
```

<sup>2</sup>We refer to anyone who is neither a Republican nor a Democrat as Independent.

The file `RetiredSen.txt` contains the records from the file `Senate110.txt` for senators who left the Senate after the 2008 election due to retirement, defeat, death, or resignation. Some records in the file are as follows:

```
Barack Obama,Illinois,D
Hillary Clinton,New York,D
Elizabeth Dole,North Carolina,R
```

The file `NewSen.txt` contains records for the senators who were newly elected in 2008 or who were appointed to fill the seats of senators who left after the 2008 election. Some records in the file are as follows:

```
Mike Johanns,Nebraska,R
Kirsten Gillibrand,New York,D
Mark Warner,Virginia,D
```

- (a) Write a program that uses the three files above to create the file `Senate111.txt` that contains records (each consisting of three fields) for the members of the 111th Senate. Use this file in parts (b), (c), and (d).
- (b) Write a program that determines the number of senators of each party affiliation.
- (c) Write a program that determines the number of states whose two senators have the same party affiliation. **Hint:** Use a procedure with the heading `Function SameParty(ByVal state As String) As Boolean`.
- (d) Write a program that asks the user to select a state from a list box, and then displays the two senators from that state. See Fig. 8.17.

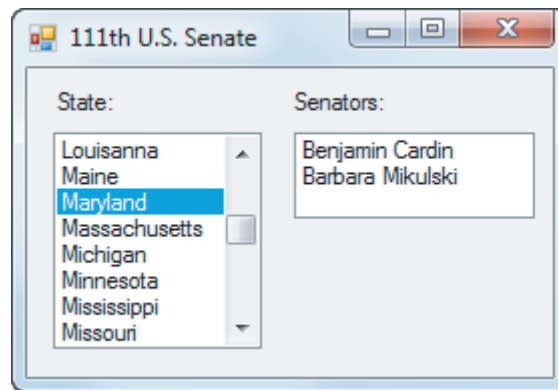


FIGURE 8.17 Possible outcome of Programming Project 2.

3. The file `Names.txt` contains a list of names in alphabetical order. Write two programs that request a name from the user and insert the name into the list in its proper location. If it is already in the list, the name should not be inserted.
  - (a) Write the first program without using any arrays or LINQ; that is, use only a `StreamReader` and a `StreamWriter`.
  - (b) Write the second program using arrays and LINQ.
4. *Create and Maintain Telephone Directories.* Write a program to create and maintain telephone directories. Each telephone directory should be contained in a separate text file. In addition, a file named `Directories.txt` should hold the names of the telephone directories. At any time, names of all the telephone directories should be displayed in a list box. After a telephone directory is selected, it becomes the *current phone directory*. The following buttons should be available.
  - (a) Create a new telephone directory. (The filename should be provided by an input dialog box.)

- (b) Add a listing (as given in text boxes) to the end of the current phone directory.
- (c) Delete a name (as given in a text box) from the current phone directory.
- (d) Display the names and phone numbers in the current phone directory. See Fig. 8.18

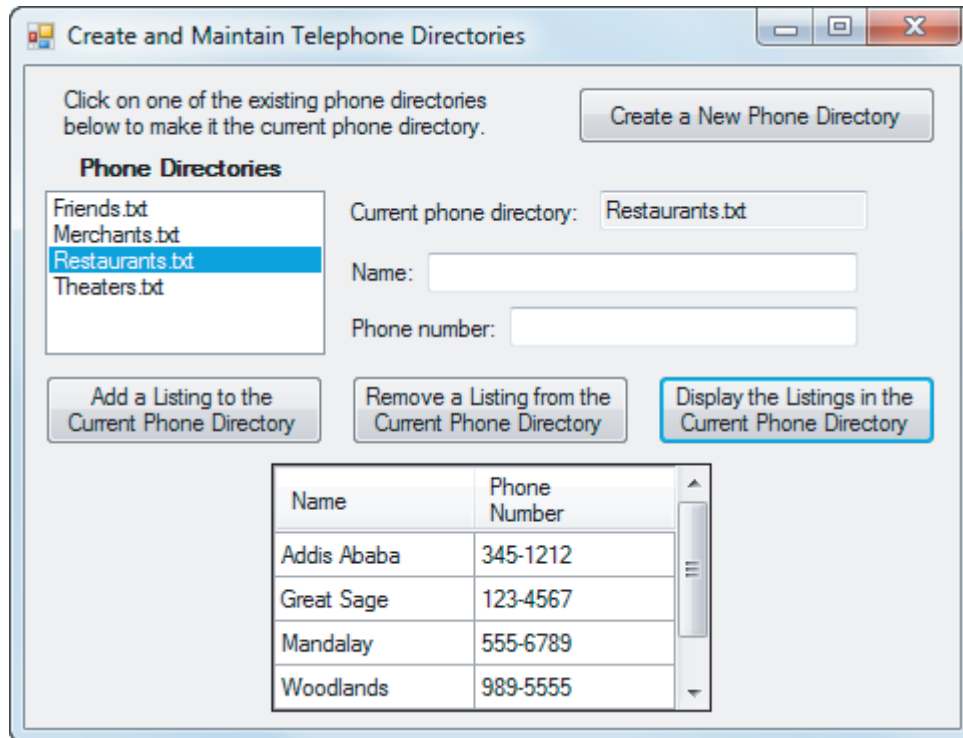


FIGURE 8.18 Sample output of Programming Project 4.

5. Each item in a supermarket is identified by its Universal Product Code (UPC), which consists of a sequence of 12 digits appearing below a rectangle of bars. See Fig. 8.19. The bars have these digits encoded in them so that the UPC can be read by an optical scanner. Let's refer to the UPC as  $d_1 d_2 d_3 d_4 d_5 d_6 d_7 d_8 d_9 d_{10} d_{11} d_{12}$ . The single digit on the left,  $d_1$ , identifies the type of product (for instance, 0 for general groceries, 2 for meat and produce, 3 for drug and health products, and 5 for coupons). The first set of five digits,  $d_2 d_3 d_4 d_5 d_6$ , identifies the manufacturer, and the second set of five digits,  $d_7 d_8 d_9 d_{10} d_{11}$ , identifies the product. The twelfth digit on the right,  $d_{12}$ , is a check digit. It is chosen so that

$$3 \cdot d_1 + d_2 + 3 \cdot d_3 + d_4 + 3 \cdot d_5 + d_6 + 3 \cdot d_7 + d_8 + 3 \cdot d_9 + d_{10} + 3 \cdot d_{11} + d_{12} \quad (*)$$

is a multiple of 10. For instance, for the UPC in Figure 8.19,

$$3 \cdot 0 + 7 + 3 \cdot 0 + 7 + 3 \cdot 3 + 4 + 3 \cdot 0 + 0 + 3 \cdot 9 + 0 \cdot 0 + 3 \cdot 3 + 4 = 40.$$



FIGURE 8.19 A Universal Product Code.



VideoNote  
DNA  
sequence  
data  
(Homework)

Since  $40 = 4 \cdot 10$ , 40 is a multiple of 10. In the event that the cashier has to enter the UPC manually and mistypes a digit, the above sum will not add up to a multiple of 10.

Write a program to simulate an automated check-out at a supermarket. A master file, called `UPC.txt`, should have a record for each item in the supermarket consisting of fields for the UPC, the name of the item, and the price of the item. For instance, the file might contain the following records:

```
037000004301,Jif Peanut Butter - 22 oz,2.29
070734000034,Celestial Seasonings Sleepytime Tea,2.59
099482403645,365 Soda Root Beer,.55
```

The program should allow the cashier to enter UPCs one at a time and should place the UPCs in a separate text file. Each UPC should be validated with the sum (\*) as soon as it is entered and should be reentered if the sum is not a multiple of 10. After all items have been processed, the program should use the two text files to display (in a list box) a receipt similar to the one in Fig. 8.20.

```
22-oz Jif Peanut Butter: $2.29
Celestial Seasonings Sleepytime Tea: $2.59
365 Soda Root Beer: $.55
Total: $5.43
```

FIGURE 8.20 Sample output of Programming Project 5.

6. The file `Baseball.xml` contains data about the performance of major league players in the 2009 regular season. (Only players with at least 350 at bats are included in the file.) Figure 8.21 shows the beginning of the file. Write a program using the file `Baseball.xml` that requests a team as input from a list and displays the players from that team whose batting average was above the average of his teammates' batting averages that are listed in the file. The players should be sorted in decreasing order by their batting averages. The output should display each player's full name and batting average. See Fig. 8.22.

```
<?xml version='1.0'?>
<!-- This file contains data on major league baseball players.-->
<major_league_baseball>
  <player>
    <name>Aaron Hill</name>
    <team>Blue Jays</team>
    <atBats>682</atBats>
    <hits>195</hits>
  </player>
```

FIGURE 8.21 XML file for Programming Project 6.

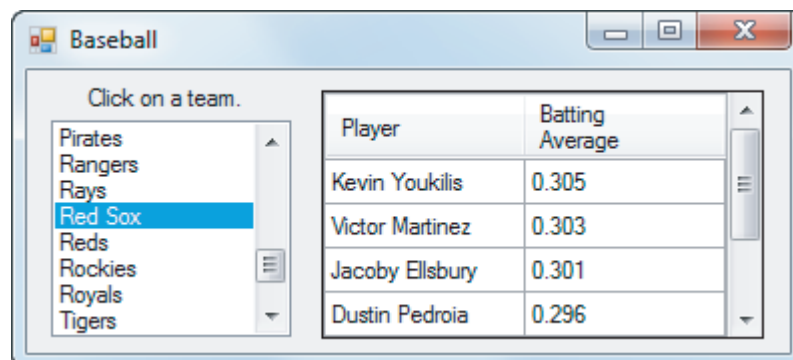


FIGURE 8.22 Possible outcome of Programming Project 6.