Chapter 8

Exercises 8.1

1.
```
Samuel Alito,NJ
Henry Baldwin,PA
```

3.
```
Alito was appointed by Bush
Baldwin was appointed by Jackson
```

5.
```
Alito,Samuel,2006
Baldwin,Henry,1830
```

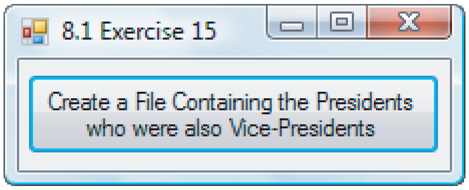
7. The new file contains the full names of the justices whose last name begins with the letter B and the years they were appointed to the court. The justices are ordered by the year they were appointed.

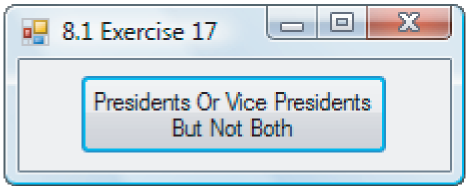
9. The new file is the same as the original file except that the last three fields have been deleted from each record.

11. The new file contains the names of the people who subscribe to either the New York Times or the Wall Street Journal, or both.

13. The new file contains the names of the people who subscribe to the New York Times but not the Wall Street Journal.

15.
```
Private Sub btnBoth_Click(...) Handles btnBoth.Click
  'Create a file of presidents who were also vice presidents
  Dim vicePres() As String = IO.File.ReadAllLines("VPres.txt")
  Dim presidents() As String = IO.File.ReadAllLines("USPres.txt")
  Dim both() As String = presidents.Intersect(vicePres).ToArray
  IO.File.WriteAllLines("Both.txt", both)
  MessageBox.Show(both.Count & " presidents", "File Created")
End Sub
```

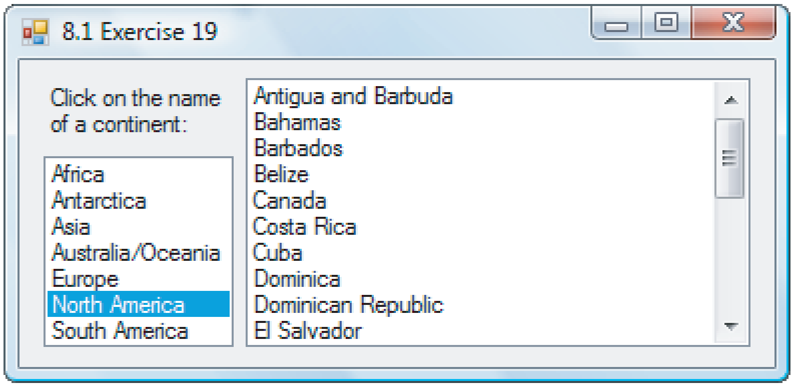

8.1 Exercise 15
Create a File Containing the Presidents who were also Vice-Presidents



17.
```
Private Sub btnXor_Click(...) Handles btnXor.Click
  'Create a file of people who were pres or VP but not both
  Dim vicePres() As String = IO.File.ReadAllLines("VPres.txt")
  Dim presidents() As String = IO.File.ReadAllLines("USPres.txt")
  Dim eitherOr() As String = presidents.Union(vicePres).ToArray
  Dim both() As String = presidents.Intersect(vicePres).ToArray
  Dim exclusiveOr() As String = eitherOr.Except(both).ToArray
  IO.File.WriteAllLines("Xor.txt", exclusiveOr)
  MessageBox.Show(exclusiveOr.Count &
      " presidents or vice presidents, but not both", "File Created")
End Sub
```

19.
```
Dim countries() As String = IO.File.ReadAllLines("UN.txt")

'lstContinents was filled at design time
Private Sub lstContinents_SelectedIndexChanged(...) Handles _
                      lstContinents.SelectedIndexChanged
  Dim selectedContinent As String = lstContinents.Text
  If selectedContinent = "Antarctica" Then
```

```
    lstCountries.DataSource = Nothing
    MessageBox.Show("There are no countries in Antarctica.")
  Else
    Dim query = From nation In countries
                Let data = nation.Split(","c)
                Let name = data(0)
                Let continent = data(1)
                Where continent = selectedContinent
                Select name
    lstCountries.DataSource = query.ToList
    lstCountries.SelectedItem = Nothing
  End If
End Sub
```

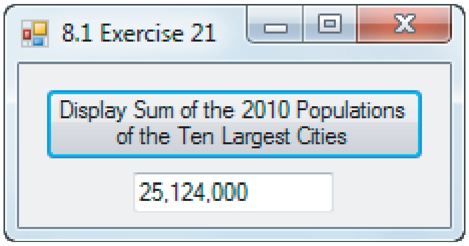


Africa
Antarctica
Asia
Australia/Oceania
Europe
North America
South America

Antigua and Barbuda
Bahamas
Barbados
Belize
Canada
Costa Rica
Cuba
Dominica
Dominican Republic
El Salvador

21.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim cities() As String = IO.File.ReadAllLines("Cities.txt")
  Dim query = From city In cities
              Let data = city.Split(","c)
              Let pop2010 = CDbl(data(3))
              Order By pop2010 Descending
              Select pop2010
  Dim pops() As Double = query.ToArray
  ReDim Preserve pops(9)
  txtOutput.Text = FormatNumber(100000 * pops.Sum, 0)
End Sub
```

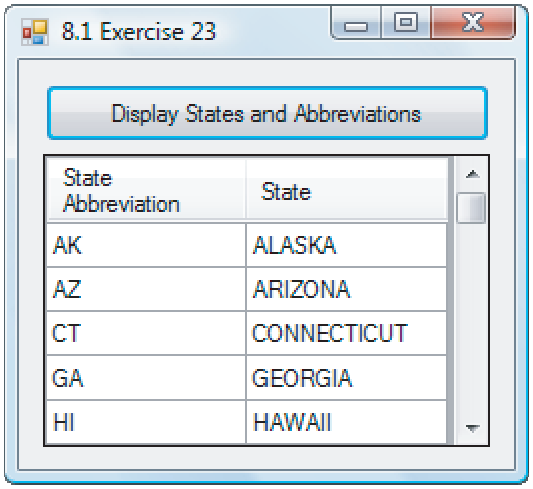

8.1 Exercise 21
Display Sum of the 2010 Populations of the Ten Largest Cities

25,124,000

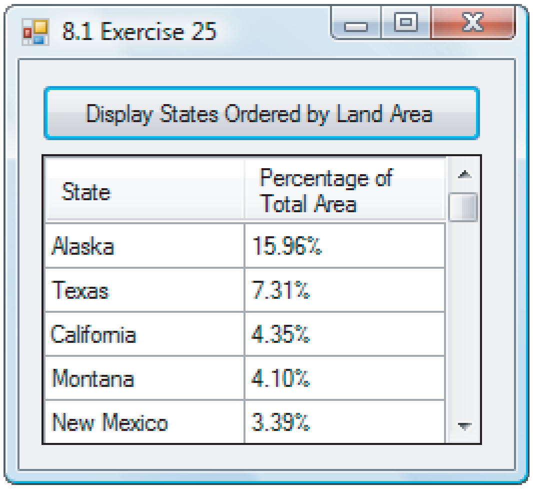
23.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim states() As String = IO.File.ReadAllLines("USStates.txt")
  Dim query = From line In states
              Let name = line.Split(","c)(0).ToUpper
              Let abbrev = line.Split(","c)(1)
              Where Not name.StartsWith(abbrev)
              Order By name Ascending
              Select abbrev, name
```

```
    dgvOutput.DataSource = query.ToList
    dgvOutput.CurrentCell = Nothing
    dgvOutput.Columns("abbrev").HeaderText = "State Abbreviation"
    dgvOutput.Columns("name").HeaderText = "State"
End Sub
```



State Abbreviation	State
AK	ALASKA
AZ	ARIZONA
CT	CONNECTICUT
GA	GEORGIA
HI	HAWAII



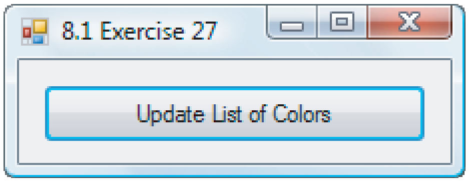
State	Percentage of Total Area
Alaska	15.96%
Texas	7.31%
California	4.35%
Montana	4.10%
New Mexico	3.39%

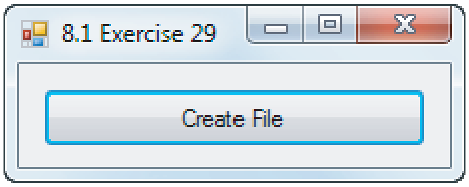
```
25. Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
      Dim states() As String = IO.File.ReadAllLines("USStates.txt")
      Dim query1 = From line In states
                   Let area = CInt(line.Split(","c)(2))
                   Select area
      Dim totalArea = query1.Sum
      Dim query2 = From line In states
                   Let name = line.Split(","c)(0)
                   Let area = CInt(line.Split(","c)(2))
                   Let percentArea = FormatPercent(area / totalArea)
                   Order By area Descending
                   Select name, percentArea
      dgvOutput.DataSource = query2.ToList
      dgvOutput.CurrentCell = Nothing
      dgvOutput.Columns("name").HeaderText = "State"
      dgvOutput.Columns("percentArea").HeaderText =
                                  "Percentage of Total Area"
    End Sub
```

```
27. Private Sub btnUpdate_Click(...) Handles btnUpdate.Click
      Dim colors() As String = IO.File.ReadAllLines("Pre1990Colors.txt")
      Dim retired() As String = IO.File.ReadAllLines("RetiredColors.txt")
      Dim added() As String = IO.File.ReadAllLines("AddedColors.txt")
      Dim tempArray() As String = colors.Except(retired).ToArray
      Dim query = From color In tempArray.Concat(added)
                  Order By color
                  Select color
      IO.File.WriteAllLines("NewColors.txt", query)
    End Sub
```

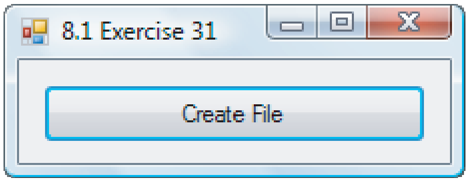
```
29. Private Sub btnCreate_Click(...) Handles btnCreate.Click
      Dim justices() As String = IO.File.ReadAllLines("Justices.txt")
```

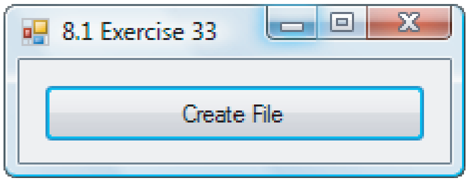
```
  Dim query = From justice In justices
              Let data = justice.Split(","c)
              Let firstName = data(0)
              Let secondName = data(1)
              Let pres = data(2)
              Let yrAppt = data(4)
              Let yrLeft = data(5)
              Select firstName & "," & secondName & "," & pres & "," &
                     yrAppt & "," & yrLeft
  IO.File.WriteAllLines("JusticesNoState.txt", query)
End Sub
```





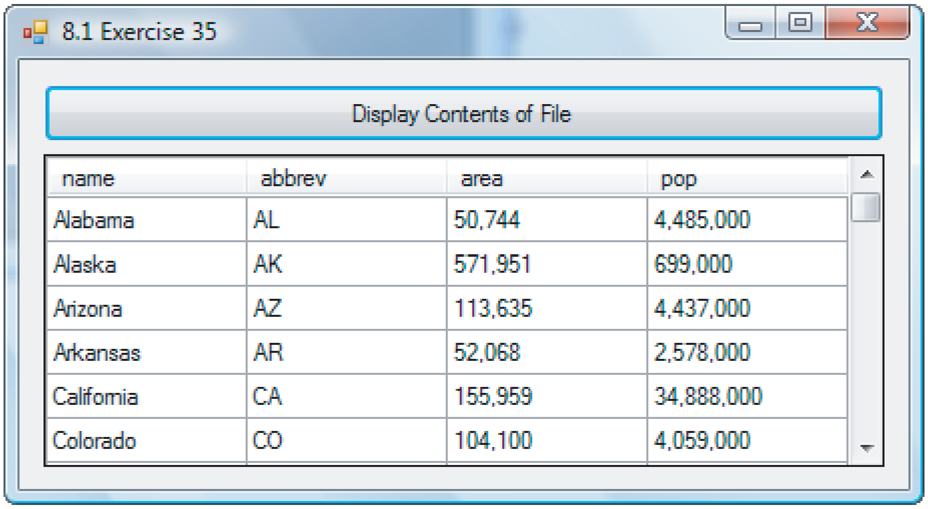
31.
```
Private Sub btnCreate_Click(...) Handles btnDisplay.Click
  'query1: all states; query2: states with justices
  Dim states() As String = IO.File.ReadAllLines("USStates.txt")
  Dim justices() As String = IO.File.ReadAllLines("Justices.txt")
  Dim query1 = From state In states
               Let abbrev = state.Split(","c)(1)
               Select abbrev
  Dim query2 = From justice In justices
               Let state = justice.Split(","c)(3)
               Select state
  IO.File.WriteAllLines("NoJustices.txt", query1.Except(query2))
End Sub
```





33.
```
Dim justices() As String = IO.File.ReadAllLines("Justices.txt")

Private Sub Create_Click(...) Handles Create.Click
  Dim query = From line In justices
              Let state = line.Split(","c)(3)
              Order By state Ascending
              Select state & "," & NumberOfJustices(state)
              Distinct
  IO.File.WriteAllLines("NewFile.txt", query)
End Sub

Function NumberOfJustices(ByVal state As String) As Integer
  Dim query = From line In justices
              Let place = line.Split(","c)(3)
              Where place = state
              Select place
  Return query.Count
End Function
```

35.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From state In IO.File.ReadAllLines("USStates.txt")
              Let data = state.Split(","c)
              Let name = data(0)
              Let abbrev = data(1)
              Let area = FormatNumber(data(2), 0)
              Let pop = FormatNumber(data(3), 0)
              Order By name
              Select name, abbrev, area, pop
  dgvStates.DataSource = query.ToList
  dgvStates.CurrentCell = Nothing
End Sub
```



name	abbrev	area	pop
Alabama	AL	50,744	4,485,000
Alaska	AK	571,951	699,000
Arizona	AZ	113,635	4,437,000
Arkansas	AR	52,068	2,578,000
California	CA	155,959	34,888,000
Colorado	CO	104,100	4,059,000

Exercises 8.2

1. `Hello`   3. `Bon Jour`   5. `You must enter a number.`

7. `Error occurred.`   9. `File Ages.txt contains an invalid age.`

11. The file Welcome.txt is created and has the following lines:

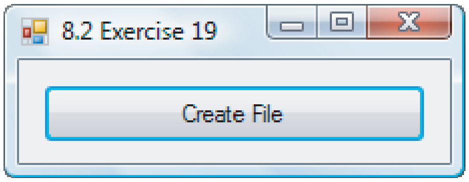
```
Hello
Bon Jour
```

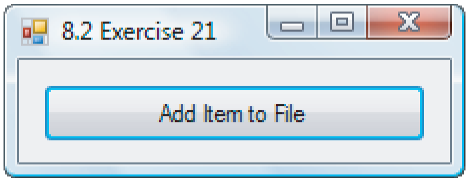
13. The filespec `Greetings.txt` should be delimited with quotation marks.

15. There should be no quotations around the variable `name` as the argument to the CreateText method.

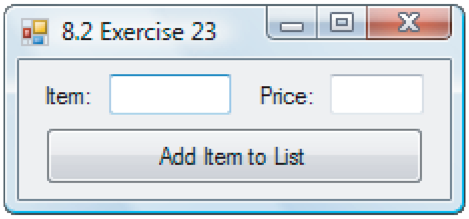
17. The variable *`age`* is declared within the Try-Catch-Finally block. Therefore it has block-level scope and is not available below the line `End Try`.

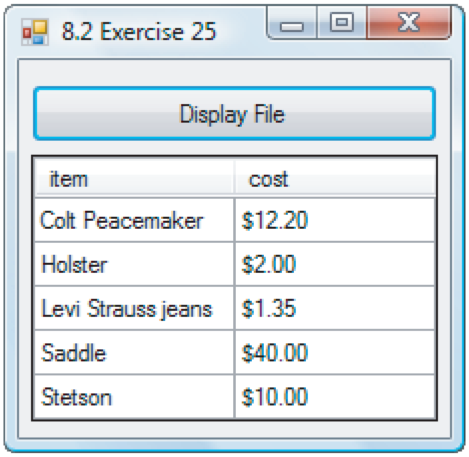
19.
```
Private Sub btnCreate_Click(...) Handles btnCreate.Click
  'Create a text file and populate it
  Dim sw As IO.StreamWriter = IO.File.CreateText("Cowboy.txt")
  sw.WriteLine("Colt Peacemaker,12.20")
  sw.WriteLine("Holster,2.00")
  sw.WriteLine("Levi Strauss jeans,1.35")
  sw.WriteLine("Saddle,40.00")
  sw.WriteLine("Stetson,10.00")
  sw.Close()  'Always close the writer when finished.
  MessageBox.Show("The file has been created.", "Done")
End Sub
```





21.
```
Private Sub btnAdd_Click(...) Handles btnAdd.Click
  'Append item to a text file
  Dim sw As IO.StreamWriter = IO.File.AppendText("Cowboy.txt")
  sw.WriteLine("Winchester Rifle,20.50")
  sw.Close()
  MessageBox.Show("The item has been added to the file.", "DONE")
End Sub
```





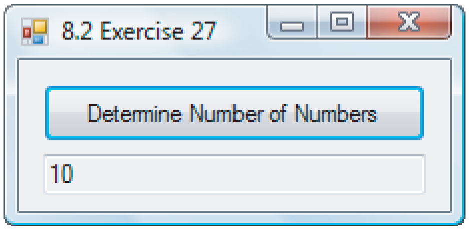
item	cost
Colt Peacemaker	$12.20
Holster	$2.00
Levi Strauss jeans	$1.35
Saddle	$40.00
Stetson	$10.00

23.
```
Private Sub btnAdd_Click(...) Handles btnAdd.Click
  Dim item As String
  Dim price As Double
  Dim sr As IO.StreamReader = IO.File.OpenText("Cowboy.txt")
  Dim sw As IO.StreamWriter = IO.File.CreateText("Cowboy2.txt")
  Dim flag As Boolean = False
  Dim line As String = ""
  Dim data() As String
  Do While (line < txtItem.Text) And (Not sr.EndOfStream)
    line = sr.ReadLine
    data = line.Split(","c)
    item = data(0)
    price = CDbl(data(1))
    If item > txtItem.Text Then
      sw.WriteLine(txtItem.Text & "," & FormatNumber(txtPrice.Text))
      'Set flag to True so we don't add it again at the end
      flag = True
    End If
    sw.WriteLine(line)
  Loop
  Do Until sr.EndOfStream
    line = sr.ReadLine
    sw.WriteLine(line)
  Loop
  If Not flag Then
    sw.WriteLine(txtItem.Text & "," & FormatNumber(txtPrice.Text))
  End If
```

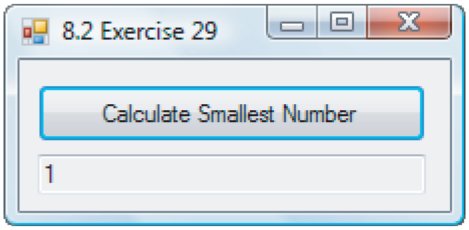
```
      sr.Close()
      sw.Close()
      MessageBox.Show("Item added to Cowboy2.txt")
      txtItem.Clear()
      txtPrice.Clear()
    End Sub
```

```
25. Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
      Dim sr As IO.StreamReader = IO.File.OpenText("Cowboy.txt")
      Dim lines(4) As String
      For i As Integer = 0 To 4
        lines(i) = sr.ReadLine
      Next
      Dim query = From line In lines
                  Let data = line.Split(","c)
                  Let item = data(0)
                  Let cost = FormatCurrency(data(1))
                  Select item, cost
      dgvOutput.DataSource = query.ToList
      dgvOutput.CurrentCell = Nothing
    End Sub
```

```
27. Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
      Dim sr As IO.StreamReader = IO.File.OpenText("Numbers.txt")
      Dim counter As Integer = 0
      Dim num As Double
      Do Until sr.EndOfStream
        num = CDbl(sr.ReadLine)
        counter += 1
      Loop
      txtOutput.Text = CStr(counter)
      sr.Close()
    End Sub
```



10

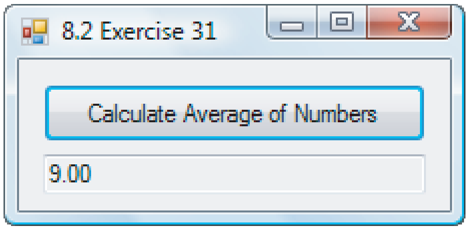


1

```
29. Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
      Dim sr As IO.StreamReader = IO.File.OpenText("Numbers.txt")
      Dim num As Double
      Dim min As Double = CDbl(sr.ReadLine)
      Do Until sr.EndOfStream
        num = CDbl(sr.ReadLine)
        If num < min Then
          min = num
        End If
      Loop
      txtOutput.Text = CStr(min)
      sr.Close()
    End Sub
```

```
31. Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
      Dim sr As IO.StreamReader = IO.File.OpenText("Numbers.txt")
```

```
 Dim counter As Integer = 0
 Dim total As Double = 0
 Dim num As Double
 Do Until sr.EndOfStream
  num = CDbl(sr.ReadLine)
  counter += 1
  total += num
 Loop
 txtOutput.Text = FormatNumber(total / counter)
 sr.Close()
End Sub
```



9.00

Exercises 8.3

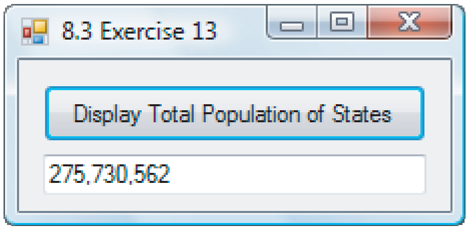
1. No **3.** No **5.** No **7.** No **9.** No

11.

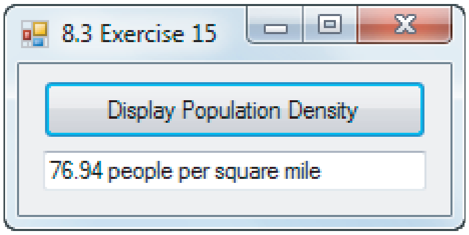
```
<?xml version='1.0'?>
<!-- This file contains the ages of the presidents when inaugurated.-->
<Presidents>
 <president>
  <name>George Washington</name>
  <ageAtInauguation>57</ageAtInauguation>
 </president>
 <president>
  <name>John Adams</name>
  <ageAtInauguation>61</ageAtInauguation>
 </president>
</Presidents>
```

13.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
 Dim stateData As XElement = XElement.Load("USStates.xml")
 Dim query = From st In stateData.Descendants("state")
             Let pop = CInt(st.<population>.Value)
             Select pop
 txtOutput.Text = FormatNumber(query.Sum, 0)
End Sub
```



275,730,562



76.94 people per square mile

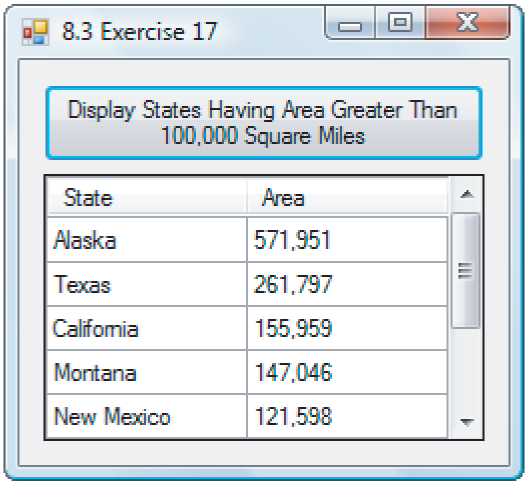
15.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
 Dim stateData As XElement = XElement.Load("USStates.xml")
```

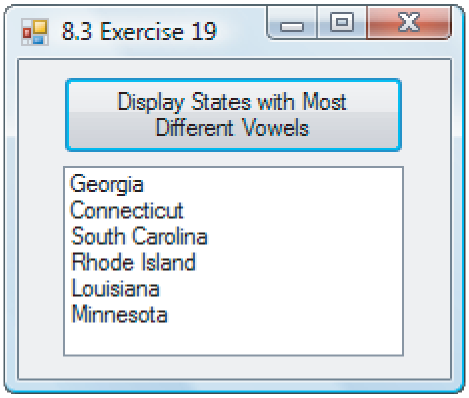
```
  Dim queryPop = From st In stateData.Descendants("state")
                 Let pop = CInt(st.<population>.Value)
                 Select pop
  Dim queryArea = From st In stateData.Descendants("state")
                  Let area = CInt(st.<area>.Value)
                  Select area
  txtOutput.Text = FormatNumber(queryPop.Sum / queryArea.Sum) &
                   " people per square mile"
End Sub
```

17.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim stateData As XElement = XElement.Load("USStates.xml")
  Dim query = From st In stateData.Descendants("state")
              Let name = st.<name>.Value
              Let area = CDbl(st.<area>.Value)
              Let formattedArea = FormatNumber(area, 0)
              Where area > 100000
              Order By area Descending
              Select name, formattedArea
  dgvStates.DataSource = query.ToList
  dgvStates.CurrentCell = Nothing
  dgvStates.Columns("name").HeaderText = "State"
  dgvStates.Columns("formattedArea").HeaderText = "Area"
End Sub
```


8.3 Exercise 17
Display States Having Area Greater Than 100,000 Square Miles

State	Area
Alaska	571,951
Texas	261,797
California	155,959
Montana	147,046
New Mexico	121,598



```
Georgia
Connecticut
South Carolina
Rhode Island
Louisiana
Minnesota
```

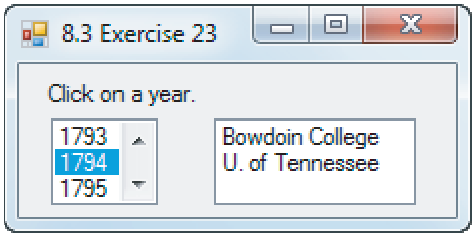
19.

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim stateData As XElement = XElement.Load("USStates.xml")
  Dim query1 = From st In stateData.Descendants("state")
               Let name = st.<name>.Value
               Let numVowels = NumberOfVowels(name)
               Order By numVowels Descending
               Select numVowels
  Dim maxVowels As Integer = query1.First
  Dim query2 = From st In stateData.Descendants("state")
               Let name = st.<name>.Value
               Where NumberOfVowels(name) = maxVowels
               Select name
  lstOutput.DataSource = query2.ToList
  lstOutput.SelectedItem = Nothing
End Sub
```

```
Function NumberOfVowels(ByVal word As String) As Integer
  word = word.ToUpper
  Dim num As Integer = 0
  If word.IndexOf("A") <> −1 Then
    num += 1
  End If
  If word.IndexOf("E") <> −1 Then
    num += 1
  End If
  If word.IndexOf("I") <> −1 Then
    num += 1
  End If
  If word.IndexOf("O") <> −1 Then
    num += 1
  End If
  If word.IndexOf("U") <> −1 Then
    num += 1
  End If
  Return num
End Function
```

21.
```
Dim colleges As XElement = XElement.Load("Colleges.xml")

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Display the colleges in Colleges.xml located in the given state
  Dim chosenState As String = mtbState.Text.ToUpper   'mask LL
  Dim query = From col In colleges.Descendants("college")
              Let name = col.<name>.Value
              Let state = col.<state>.Value
              Let yearFounded = col.<yearFounded>.Value
              Where state = chosenState
              Order By name Ascending
              Select name & " " & yearFounded
  lstColleges.DataSource = query.ToList
  lstColleges.SelectedItem = Nothing
End Sub
```

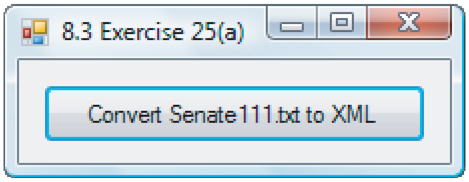
23.
```
Dim colleges As XElement = XElement.Load("Colleges.xml")

Private Sub frmColleges_Load(...) Handles MyBase.Load
  'Place the years for each college into the left list box
  Dim query = From col In colleges.Descendants("college")
              Let yearFounded = col.<yearFounded>.Value
              Order By yearFounded Ascending
              Select yearFounded
              Distinct
  For Each yr in query
    lstYears.Items.Add(yr)
  Next
End Sub

Private Sub lstYears_SelectedIndexChanged(...) Handles _
               lstYears.SelectedIndexChanged
  Dim chosenYear As String = lstYears.Text
  Dim query = From col In colleges.Descendants("college")
              Let name = col.<name>.Value
              Let yearFounded = col.<yearFounded>.Value
              Where yearFounded = chosenYear
              Select name
  lstColleges.DataSource = query.ToList
  lstColleges.SelectedItem = Nothing
End Sub
```

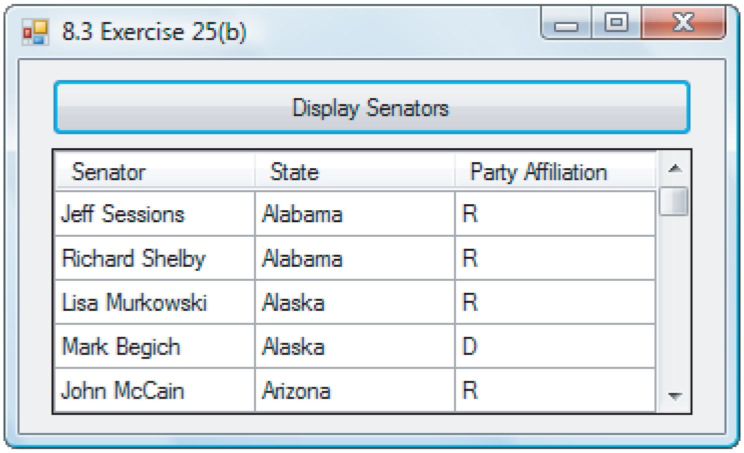


Bowdoin College
U. of Tennessee

25(a).

```
Private Sub btnSenate_Click(...) Handles btnSenate.Click
  Dim sr As IO.StreamReader = IO.File.OpenText("Senate111.txt")
  Dim sw As IO.StreamWriter = IO.File.CreateText("Senate111.xml")
  sw.WriteLine("<?xml version='1.0'?>")
  sw.WriteLine("<!-- This file contains data on the 111th Senate -->")
  sw.WriteLine("<Senate111>")
  Dim temp As String = ""
  Do Until sr.EndOfStream
    temp = sr.ReadLine
    sw.WriteLine("  <senator>")
    sw.WriteLine("    <name>" & temp.Split(","c)(0) & "</name>")
    sw.WriteLine("    <state>" & temp.Split(","c)(1) & "</state>")
    sw.WriteLine("    <party>" & temp.Split(","c)(2) & "</party>")
    sw.WriteLine("  </senator>")
  Loop
  sw.WriteLine("</Senate111>")
  sr.Close()
  sw.Close()
  MessageBox.Show("File Created")
End Sub
```





Senator	State	Party Affiliation
Jeff Sessions	Alabama	R
Richard Shelby	Alabama	R
Lisa Murkowski	Alaska	R
Mark Begich	Alaska	D
John McCain	Arizona	R

25(b).

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim senateData As XElement = XElement.Load("Senate111.XML")
  Dim query = From st In senateData.Descendants("senator")
              Let name = st.<name>.Value
              Let state = st.<state>.Value
              Let party = st.<party>.Value
              Order By state, name Ascending
              Select name, state, party
  dgvSenators.DataSource = query.ToList
  dgvSenators.CurrentCell = Nothing
  dgvSenators.Columns("name").HeaderText = "Senator"
  dgvSenators.Columns("state").HeaderText = "State"
  dgvSenators.Columns("party").HeaderText = "Party Affiliation"
End Sub
```

Chapter 9

Exercises 9.1

1. Chopin is deleted from the list.

3. The currently selected item in **lstBox,** Mozart, is deleted.

5. The item Haydn is inserted into **lstBox** between Chopin and Mozart.

7. The names in the list box will appear in descending alphabetical order.

9.
```
cboBox.Text = "Dante"
```

11.
```
cboBox.Items.Remove("Shakespeare")
```

13.
```
cboBox.Items.RemoveAt(cboBox.Items.Count − 1)
```

15.
```
Dim i As Integer = 0
Do While i < cboBox.Items.Count
  If CStr(cboBox.Items(i)).Substring(0, 1) = "M" Then
    cboBox.Items.RemoveAt(i)
  Else
    i += 1
  End If
Loop
```

17.
```
Private Sub btnSort_Click(...) Handles btnSort.Click
  Dim names() As String = IO.File.ReadAllLines("PopularName.txt")
  lstOutput.Sorted = True
  lstOutput.DataSource = names
  lstOutput.SelectedItem = Nothing
  For i As Integer = 0 To lstOutput.Items.Count − 1
    names(i) = CStr(lstOutput.Items(i))
  Next
  IO.File.WriteAllLines("SortedNames.txt", names)
  MessageBox.Show("The ordered file has been created.", "Done")
End Sub
```

23.
```
Dim ages() As String = IO.File.ReadAllLines("AgesAtInaugural.txt")

Private Sub frmPres_Load(...) Handles MyBase.Load
  Dim pres() As String = IO.File.ReadAllLines("USPres.txt")
  lstPres.DataSource = pres
End Sub
```