```
 1  Option Explicit On
 2  Option Strict On
 3
 4  Public Class Form1
 5      Dim summary As New List(Of String)
 6
 7      'Private Function HorizontalBorder(length As Integer, characterSet As
          String) As String
 8      '    Dim tempstring As String = ""
 9      '    For i = 0 To length
10      '        tempstring &= characterSet
11      '    Next
12
13      '    Return tempstring
14      'End Function
15      Private Function ColumnHeaders(firstColumn As String, secondColumn As
          String, thirdColumn As String, fourthColumn As String) As String
16          'Description, Unit, Rate, Subtotal
17          Dim columnWidth As Integer = 20
18          firstColumn = "| " & firstColumn
19          secondColumn = "| " & secondColumn
20          thirdColumn = "| " & thirdColumn
21          fourthColumn = "| " & fourthColumn
22
23          Return firstColumn.PadRight(columnWidth) & secondColumn.PadRight
              (columnWidth) &
24           thirdColumn.PadRight(columnWidth) & fourthColumn.PadRight(columnWidth
              - 1) & "|"
25
26      End Function
27      Private Sub clearData()
28          ' []update to clear summary?
29          stockCodeTextBox.Clear()
30          mondayValMaskedTextBox.Clear()
31          tuesdayValMaskedTextBox.Clear()
32          wednesdayValMaskedTextBox.Clear()
33          stockListBox.Items.Clear()
34
35      End Sub
36
37      Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs)
          Handles ExitToolStripMenuItem.Click
38          Me.Close()
39      End Sub
40
41      Private Sub CalculateToolStripMenuItem_Click(sender As Object, e As
          EventArgs) Handles CalculateToolStripMenuItem.Click,
42          CalculateToolStripMenuItem1.Click
43          'Validates data entered and returns messagebox with any fields that are
```

```
            invalid,
44          'selecting the first in the tab order.
45          Dim errorSummary As String = ""
46          Dim firstDayChange As Double
47          Dim secondDayChange As Double
48          Dim decision As String
49          Static buyCount As Integer = 0
50          Static sellCount As Integer = 0
51          Static holdCount As Integer = 0
52
53
54          Try
55              Double.Parse(wednesdayValMaskedTextBox.Text)
56          Catch ex As Exception
57              errorSummary = "Please enter a valid Price in Wednesday." &
                  vbNewLine & errorSummary
58              wednesdayValMaskedTextBox.Select()
59          End Try
60          Try
61              Double.Parse(tuesdayValMaskedTextBox.Text)
62          Catch ex As Exception
63              errorSummary = "Please enter a valid Price in Tuesday." & vbNewLine
                   & errorSummary
64              tuesdayValMaskedTextBox.Select()
65          End Try
66          Try
67              Double.Parse(mondayValMaskedTextBox.Text)
68          Catch ex As Exception
69              errorSummary = "Please enter a valid Price in Monday." & vbNewLine
                  & errorSummary
70              mondayValMaskedTextBox.Select()
71          End Try
72          If stockCodeTextBox.Text = "" Then
73              errorSummary = "Please enter the stock code you would like to
                  calculate." _
74                  & vbNewLine & errorSummary
75              stockCodeTextBox.Select()
76          End If
77
78          'errorSummary is updated only in catch statements, and if it is a
              summary is shown in the
79          'Message Box
80          If errorSummary <> "" Then
81              MessageBox.Show(errorSummary)
82          End If
83
84          'If errorSummary was blank then the calculations are performed and
              added to the listbox.
85          'The function HoldBuySell() checks the decision, and chooses the case
```

```
            to run.
86          If errorSummary = "" Then
87              firstDayChange = Double.Parse(tuesdayValMaskedTextBox.Text) -
                  Double.Parse(mondayValMaskedTextBox.Text)
88              secondDayChange = Double.Parse(wednesdayValMaskedTextBox.Text) -
                  Double.Parse(tuesdayValMaskedTextBox.Text)
89              'HoldBuySell(firstDayChange, secondDayChange)
90              decision = CStr(HoldBuySell(firstDayChange, secondDayChange))
91              'Console.WriteLine(firstDayChange)
92              'Console.WriteLine(secondDayChange)
93              'Console.WriteLine(decision)
94              'Console.WriteLine(ColumnHeaders(stockCodeTextBox.Text,
                  firstDayChange.ToString("C"), secondDayChange.ToString("C"),
                  decision))
95              Select Case decision
96                  Case "Buy"
97                      'Inserts at the top of the Buy section of the columns, with
                         ColumnHeaders
98                      'formatting the output.
99                      summary.Insert(0, ColumnHeaders(stockCodeTextBox.Text,
                        firstDayChange.ToString("C"), secondDayChange.ToString
                        ("C"), decision))
100                     'inserts the current value in the first place
101                     stockListBox.Items.Insert(0, ColumnHeaders
                        (stockCodeTextBox.Text, firstDayChange.ToString("C"),
                        secondDayChange.ToString("C"), decision))
102                     'Increments a count to tell the Sell case where to insert.
103                     buyCount += 1
104
105
106                 Case "Sell"
107                     'Inserts at the top of the Sell section of the columns,
                        with ColumnHeaders
108                     'formatting the output.
109                     summary.Insert(buyCount, ColumnHeaders
                        (stockCodeTextBox.Text, firstDayChange.ToString("C"),
                        secondDayChange.ToString("C"), decision))
110                     'Inserts the Sell at the end of the Buy column
111                     stockListBox.Items.Insert(buyCount, ColumnHeaders
                        (stockCodeTextBox.Text, firstDayChange.ToString("C"),
                        secondDayChange.ToString("C"), decision))
112                     'Increments a count to tell the Hold case where to insert.
113                     sellCount += 1
114                 Case "Hold"
115                     'Inserts at the top of the Hold section of the columns,
                        with ColumnHeaders
116                     'formatting the output.  It does this by adding both counts
                         together to set the
117                     'insert point.
```

```
118                     summary.Insert((buyCount + sellCount), ColumnHeaders
                        (stockCodeTextBox.Text, firstDayChange.ToString("C"),
                        secondDayChange.ToString("C"), decision))
119                     stockListBox.Items.Insert((buyCount + sellCount),
                        ColumnHeaders(stockCodeTextBox.Text,
                        firstDayChange.ToString("C"), secondDayChange.ToString
                        ("C"), decision))
120             End Select
121
122         End If
123
124     End Sub
125     Private Function HoldBuySell(firstDayChange As Double, secondDayChange As
          Double) As Object
126         'reads in the firstday to secondday change from calculate and returns
              buy or sell if the value
127         'has changed according to Justine's Algorithm.  Otherwise returns hold.
128
129         Dim result As String
130         If firstDayChange > 0 And secondDayChange > firstDayChange Then
131             result = "Buy"
132         ElseIf firstDayChange < 0 And secondDayChange < firstDayChange Then
133             result = "Sell"
134         Else
135             result = "Hold"
136         End If
137
138         Return result
139
140     End Function
141
142     Private Sub ClearToolStripMenuItem_Click(sender As Object, e As EventArgs)
          Handles ClearToolStripMenuItem.Click,
143                     ClearToolStripMenuItem1.Click()
144         'Inserts at the top of the Sell section of the columns, with
              ColumnHeaders
145         'formatting the output.
146         clearData()
147         stockListBox.Items.Clear()
148         summary.Clear()
149
150     End Sub
151
152     Private Sub SummaryToolStripMenuItem_Click(sender As Object, e As
          EventArgs) Handles SummaryToolStripMenuItem.Click,
          SummaryToolStripMenuItem1.Click
153         'Loops through and adds all stock prices calculated to a String, then
              uses a message box to
154         'show a summary.
```

```
155         Dim msgSummary As String
156         For i = 0 To (summary.Count - 1)
157             Console.WriteLine(summary(i))
158             msgSummary &= summary(i) & vbNewLine
159         Next
160         MessageBox.Show(msgSummary)
161     End Sub
162
163 End Class
164
```