

```
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 ColumnHeaders(firstColumn As String, secondColumn As String,
8         thirdColumn As String, fourthColumn As String) As String
9         'Updates the formatting of the stock picks.
10        'Description, Unit, Rate, Subtotal
11        Dim columnWidth As Integer = 20
12        firstColumn = "|" & firstColumn
13        secondColumn = "|" & secondColumn
14        thirdColumn = "|" & thirdColumn
15        fourthColumn = "|" & fourthColumn & "|"
16
17        Return firstColumn.PadRight(columnWidth) & secondColumn.PadRight(columnWidth) &
18            thirdColumn.PadRight(columnWidth) & fourthColumn.PadRight(columnWidth) & " |"
19
20    End Function
21    Private Sub clearData()
22        'Clears Data
23        stockCodeTextBox.Clear()
24        mondayValMaskedTextBox.Clear()
25        tuesdayValMaskedTextBox.Clear()
26        wednesdayValMaskedTextBox.Clear()
27        stockListBox.Items.Clear()
28    End Sub
29
30    Private Sub ExitToolStripMenuItem_Click(sender As Object, e As EventArgs)
31        Handles ExitToolStripMenuItem.Click
32        Me.Close()
33    End Sub
34
35    Private Sub CalculateToolStripMenuItem_Click(sender As Object, e As EventArgs)
36        Handles CalculateToolStripMenuItem.Click, CalculateToolStripMenuItem1.Click
37        'Validates data entered and returns messagebox with any fields that are invalid,
38        'selecting the first in the tab order.
39        Dim errorSummary As String = ""
40        Dim firstDayChange As Double
41        Dim secondDayChange As Double
42        Dim decision As String
43        Static buyCount As Integer = 0
```

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

```
85         decision = CStr(HoldBuySell(firstDayChange, secondDayChange))
86
87     Select Case decision
88         Case "Buy"
89             'Inserts at the top of the Buy section of the columns, with
90             ColumnHeaders
91             'formatting the output.
92             summary.Insert(0, ColumnHeaders(stockCodeTextBox.Text,
93             firstDayChange.ToString("C"),
94             secondDayChange.ToString("C"), decision))
95             'inserts the current value in the first place
96             stockListBox.Items.Insert(0, ColumnHeaders
97             (stockCodeTextBox.Text, firstDayChange.ToString("C"),
98             secondDayChange.ToString("C"), decision))
99             'Increments a count to tell the Sell case where to insert.
100            buyCount += 1
101
102        Case "Sell"
103            'Inserts at the top of the Sell section of the columns,
104            with ColumnHeaders
105            'formatting the output.
106            summary.Insert(buyCount, ColumnHeaders
107            (stockCodeTextBox.Text, firstDayChange.ToString("C"),
108            secondDayChange.ToString("C"), decision))
109            'Inserts the Sell at the end of the Buy column
110            stockListBox.Items.Insert(buyCount,
111            ColumnHeaders(stockCodeTextBox.Text,
112            firstDayChange.ToString("C"),
113            secondDayChange.ToString("C"), decision))
114            'Increments a count to tell the Hold case where to insert.
115            sellCount += 1
116        Case "Hold"
117            'Inserts at the top of the Hold section of the columns,
118            with ColumnHeaders
119            'formatting the output. It does this by adding both counts
120            together to set the
121            'insert point.
122            summary.Insert((buyCount + sellCount),
123            ColumnHeaders(stockCodeTextBox.Text,
124            firstDayChange.ToString("C"),
125            secondDayChange.ToString
126            ("C"), decision))
127            stockListBox.Items.Insert((buyCount + sellCount),
128            ColumnHeaders(stockCodeTextBox.Text,
129            firstDayChange.ToString("C"),
130            secondDayChange.ToString
131            ("C"), decision))
132    End Select
```

```

122
123     End If
124
125 End Sub
126 Private Function HoldBuySell(firstDayChange As Double, secondDayChange As
    Double) As Object
127     'reads in the firstday to secondday change from calculate and returns
    buy or sell if the value
128     'has changed according to Justine's Algorithm. Otherwise returns hold.
129
130     Dim result As String
131     If firstDayChange > 0 And secondDayChange > firstDayChange Then
132         result = "Buy"
133     ElseIf firstDayChange < 0 And secondDayChange < firstDayChange Then
134         result = "Sell"
135     Else
136         result = "Hold"
137     End If
138
139     Return result
140
141 End Function
142
143 Private Sub ClearToolStripMenuItem_Click(sender As Object, e As EventArgs)
144     - Handles ClearToolStripMenuItem.Click, ClearToolStripMenuItem1.Click
145     'Inserts at the top of the Sell section of the columns, with
    ColumnHeaders
146     'formatting the output.
147     clearData()
148     stockListBox.Items.Clear()
149     summary.Clear()
150
151 End Sub
152
153 Private Sub SummaryToolStripMenuItem_Click(sender As Object, e As
    EventArgs) _
154     Handles SummaryToolStripMenuItem.Click, SummaryToolStripMenuItem1.Click
155     'Loops through and adds all stock prices calculated to a String, then
    uses a message box to
156     'show a summary.
157     Dim msgSummary As String =
158     "*****" &
    vbNewLine &
159     "*****Justine's Stock
    Picks*****" & vbNewLine &
160     "*****" &

```

```
        vbNewLine
161         For i = 0 To (summary.Count - 1)
162             'Console.WriteLine(summary(i))
163             msgSummary &= summary(i) & vbNewLine
164         Next
165         MessageBox.Show(msgSummary)
166     End Sub
167
168 End Class
169
```