```vb
Option Explicit On
Option Strict On
Option Compare Binary
Public Class RentalForm

    Public TotalCustomers As Integer
    Public TotalDistance As Decimal
    Public TotalCost As Decimal

    'Code that Loads when program starts
    Private Sub RentalForm_Load(sender As Object, e As EventArgs) Handles
      MyBase.Load
        ResetAll()
        SummaryButton.Enabled = False
    End Sub

    'Clear Button Click
    Private Sub ClearButton_Click(sender As Object, e As EventArgs) Handles
      ClearButton.Click
        ResetAll()
    End Sub

    'Calculate Button Click
    Private Sub CalculateButton_Click(sender As Object, e As EventArgs) Handles
      CalculateButton.Click
        Dim tempNumber As Integer = 15

        'If Statement when true calculates the input data. If checks fail then
          the user is given a error message box
        If ValidateNameTextBox() = True And ValidateAddressTextBox() = True And
          ValidateCityTextBox() And ValidateStateTextBox() = True And
          ValidateZipCodeTextBox() = True And ValidateOdometer() = True And
          ValidateDaysTextBox() = True Then
            DistanceBox.Text = KilometersToMiles().ToString & " mi"
            MileageBox.Text = MileageCharge().ToString("C")
            DayBox.Text = DaysCharge().ToString("C")
            MinusBox.Text = Discount().ToString("C")
            YouOweBox.Text = TotalWithDiscount().ToString("C")
            UserMessages(False, "", True)
            TotalCustomers = Customers()
            TotalDistance = Distance()
            TotalCost = Cost()
        Else
            MsgBox(UserMessages(True, "", False))
            UserMessages(False, "", True)
        End If

        'Enables the Summary Button if data is input through the calculate
          button.
        If TotalDistance > 0 And TotalCost > 0 Then
            SummaryButton.Enabled = True
        End If
```

```
45
46      End Sub
47
48      'Series of Validate Data functions used to analyze the input from the user
49      Function ValidateOdometer() As Boolean
50          Dim userMessage As String
51          Dim trueOrFalse As Boolean = True
52
53          Try
54              If CDec(BeginOdometerTextBox.Text) > CDec(EndOdometerTextBox.Text)
                  Then
55                  userMessage = "Beginning Odometer Must be Smaller than Ending
                      Odometer."
56                  BeginOdometerTextBox.Text = ""
57                  EndOdometerTextBox.Text = ""
58                  trueOrFalse = False
59              End If
60          Catch ex As Exception
61              userMessage = "Odometer entries must be numbers."
62              BeginOdometerTextBox.Text = ""
63              EndOdometerTextBox.Text = ""
64              trueOrFalse = False
65          End Try
66
67          UserMessages(True, userMessage, False)
68
69          Return trueOrFalse
70      End Function
71
72      Function ValidateNameTextBox() As Boolean
73          Dim userMessage As String
74          Dim trueOrFalse As Boolean = True
75
76          If NameTextBox.Text = "" Then
77              userMessage = "Please enter a Name."
78              UserMessages(True, userMessage, False)
79              NameTextBox.Focus()
80              trueOrFalse = False
81          End If
82
83          Return trueOrFalse
84      End Function
85
86      Function ValidateAddressTextBox() As Boolean
87          Dim userMessage As String
88          Dim trueOrFalse As Boolean = True
89
90          If AddressTextBox.Text = "" Then
91              userMessage = "Please enter an Address."
92              UserMessages(True, userMessage, False)
93              NameTextBox.Focus()
94              trueOrFalse = False
```

```
95          End If
96
97          Return trueOrFalse
98      End Function
99
100     Function ValidateCityTextBox() As Boolean
101         Dim userMessage As String
102         Dim trueOrFalse As Boolean = True
103
104         If CityTextBox.Text = "" Then
105             userMessage = "Please enter a City."
106             UserMessages(True, userMessage, False)
107             NameTextBox.Focus()
108             trueOrFalse = False
109         End If
110
111         Return trueOrFalse
112     End Function
113
114     Function ValidateStateTextBox() As Boolean
115         Dim userMessage As String
116         Dim trueOrFalse As Boolean = True
117
118         If StateTextBox.Text = "" Then
119             userMessage = "Please enter a State."
120             UserMessages(True, userMessage, False)
121             NameTextBox.Focus()
122             trueOrFalse = False
123         End If
124
125         Return trueOrFalse
126     End Function
127
128     Function ValidateZipCodeTextBox() As Boolean
129         Dim userMessage As String
130         Dim zipCode As Integer
131         Dim trueOrFalse As Boolean = True
132
133         Try
134             zipCode = CInt(ZipCodeTextBox.Text)
135         Catch ex As Exception
136             userMessage = "Please enter a valid Zip Code."
137             UserMessages(True, userMessage, False)
138             ZipCodeTextBox.Text = ""
139             ZipCodeTextBox.Focus()
140             trueOrFalse = False
141         End Try
142
143         Return trueOrFalse
144     End Function
145
146     Function ValidateDaysTextBox() As Boolean
```

```
147         Dim userMessage As String
148         Dim trueOrFalse As Boolean = True
149
150         Try
151             If CInt(DaysTextBox.Text) > 45 Or CInt(DaysTextBox.Text) < 1 Then
152                 userMessage = "Please enter days between 1 and 45."
153                 UserMessages(True, userMessage, False)
154                 DaysTextBox.Text = ""
155                 trueOrFalse = False
156             End If
157
158         Catch ex As Exception
159             userMessage = "Days must be numeric."
160             UserMessages(True, userMessage, False)
161             DaysTextBox.Text = ""
162             DaysTextBox.Focus()
163             trueOrFalse = False
164         End Try
165
166         Return trueOrFalse
167     End Function
168
169     'Function used to Construct a single user message with all error messages
170     Function UserMessages(addMessage As Boolean, message As String, clearMessage
          As Boolean) As String
171         Static formattedMessages As String
172
173         If clearMessage = True Then
174             formattedMessages = ""
175         ElseIf addMessage = True Then
176             formattedMessages &= message & vbNewLine
177         End If
178
179         Return formattedMessages
180     End Function
181
182     Function KilometersToMiles() As Decimal
183         Dim miles As Decimal
184
185         Try
186             miles = CDec(EndOdometerTextBox.Text) - CDec
                  (BeginOdometerTextBox.Text)
187         Catch
188         End Try
189
190         If KilometersradioButton.Checked = True Then
191             Try
192                 miles *= CDec(0.62)
193             Catch
194             End Try
195         End If
196
```

```
197         Return miles
198     End Function
199
200     Function MileageCharge() As Decimal
201         Dim milesCharge As Decimal
202         Dim miles As Decimal = KilometersToMiles()
203         Const REGULARRATE = 0.12D
204         Const LOWRATE = 0.1D
205         Const FREERATE = 0D
206
207         'Mileage Charge
208         'First 200 miles driven are always free.
209         'All miles between 201 And 500 inclusive are .12 cents per mile.
210         'miles greater than 500 are charged at .10 cents per mile.
211         Select Case miles
212             Case < 201
213                 milesCharge = miles * FREERATE
214             Case > 500
215                 milesCharge = 300 * REGULARRATE
216                 milesCharge += (miles - 500) * LOWRATE
217             Case Else
218                 milesCharge = (miles - 200) * REGULARRATE
219         End Select
220
221         Return milesCharge
222
223     End Function
224
225     Function DaysCharge() As Integer
226         Const DAYPRICE As Integer = 15
227
228         Try
229             DaysCharge = CInt(DaysTextBox.Text) * DAYPRICE
230         Catch
231         End Try
232
233         Return DaysCharge
234     End Function
235
236     Function TotalWithoutDiscount() As Decimal
237
238         TotalWithoutDiscount = MileageCharge() + DaysCharge()
239
240         Return TotalWithoutDiscount
241     End Function
242
243     Function Discount() As Decimal
244         Dim totalDiscount As Decimal
245         Const AAARATE = 0.05D
246         Const SENIORATE = 0.03D
247
248         '2 discounts, AAA at 5% and Senior at 3%
```

```
249         If AAAcheckbox.Checked = True Then
250             totalDiscount = TotalWithoutDiscount() * AAARATE
251         End If
252
253         If Seniorcheckbox.Checked = True Then
254             totalDiscount += TotalWithoutDiscount() * SENIORATE
255         End If
256
257         Return totalDiscount
258     End Function
259
260     'Function used to combine all cost into a total
261     Function TotalWithDiscount() As Decimal
262
263         TotalWithDiscount = MileageCharge() + DaysCharge() - Discount()
264
265         Return TotalWithDiscount
266     End Function
267
268     Sub ResetAll()
269
270         'Clear Everything on the Form
271         NameTextBox.Text = ""
272         AddressTextBox.Text = ""
273         CityTextBox.Text = ""
274         StateTextBox.Text = ""
275         ZipCodeTextBox.Text = ""
276         BeginOdometerTextBox.Text = ""
277         EndOdometerTextBox.Text = ""
278         DaysTextBox.Text = ""
279         MilesradioButton.Checked = True
280         AAAcheckbox.Checked = False
281         Seniorcheckbox.Checked = False
282         DistanceBox.Text = ""
283         MileageBox.Text = ""
284         DayBox.Text = ""
285         MinusBox.Text = ""
286         YouOweBox.Text = ""
287
288     End Sub
289
290     'Exit Button with warning message before closing
291     Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles
          ExitButton.Click
292         Dim msg = "Are you sure you want to close the program?"
293         'Uses MsgBox to display a warning before closing the program.
294         Dim style = MsgBoxStyle.YesNo Or MsgBoxStyle.DefaultButton2 Or
              MsgBoxStyle.Question
295         Dim title = "Car Rental"
296         Dim response = MsgBox(msg, style, title)
297
298         If response = MsgBoxResult.Yes Then
```

```
299             Me.Close()
300         End If
301
302     End Sub
303
304     Private Sub SummaryButton_Click(sender As Object, e As EventArgs) Handles
          SummaryButton.Click
305
306         'Displays strings with the public variables that contain totals from each
              form
307         MsgBox("Total Customers: " & TotalCustomers & vbNewLine & "Total
              Distance: " & TotalDistance & vbNewLine & "Total Charges: " &
              TotalCost.ToString("C"))
308
309     End Sub
310
311     Sub Summary(Customers As Integer, Distance As Decimal, Cost As Decimal)
312         Static totalCustomers As Integer
313         Static totalDistance As Decimal
314         Static totalCost As Decimal
315
316         totalCustomers += Customers
317         totalDistance += Distance
318         totalCost += Cost
319
320     End Sub
321
322     Function Customers() As Integer
323         Static total As Integer
324
325         total += 1
326
327         Return total
328     End Function
329
330     Function Distance() As Decimal
331         Dim stringLength As Integer = Len(DistanceBox.Text)
332         Dim subtractedStringLength As String = LSet(DistanceBox.Text,
              stringLength - 3)
333         Static total As Decimal
334
335         total += CDec(subtractedStringLength)
336
337         Return total
338     End Function
339
340     Function Cost() As Decimal
341         Static total As Decimal
342
343         total += CDec(YouOweBox.Text)
344
345         Return total
```

```
346         End Function
347
348     End Class
349
```