

```
1 'Doug Cluff
2 'RCET0265
3 'Asg4-6
4 'Rental Car Form
5 'https://github.com/clufdoug/DACVSF19/tree/master/Assignments/Asg4-6
6
7 Option Strict On
8 Option Explicit On
9
10 Public Class RentalForm
11
12     Dim beginOdometer As Double
13     Dim endOdometer As Double
14     Dim custName As String
15     Dim custAddress As String
16     Dim custCity As String
17     Dim custState As String
18     Dim custZip As Integer
19     Dim days As Integer
20     Dim totalcustomers As Integer
21     Dim totalmiles As Double
22     Dim totalsales As Double
23
24     Private Sub RentalForm_Load(sender As Object, e As EventArgs) Handles MyBase.Load
25         'Disables the Summary Button until input has been validated
26         SummaryButton.Enabled = False
27         NameTextBox.Text = "Tim Rossiter"
28         AddressTextBox.Text = "2019 Mr.Roboto LN."
29         CityTextBox.Text = "Chubbuck"
30         StateTextBox.Text = "Idaho"
31         ZipCodeTextBox.Text = "83202"
32     End Sub
33
34     Private Sub calculateButton_Click(sender As Object, e As EventArgs) Handles CalculateButton.Click
35         'Handles button click to begin validation protocols
36         ValidateInput()
37     End Sub
38
39     Private Sub clearButton_Click(sender As Object, e As EventArgs) Handles ClearButton.Click
40         'Handles button click to clear all boxes and calculated values
41         NameTextBox.Clear()
42         AddressTextBox.Clear()
43         CityTextBox.Clear()
44         StateTextBox.Clear()
45         ZipCodeTextBox.Clear()
46         BeginOdometerTextBox.Clear()
47         EndOdometerTextBox.Clear()
48         DaysTextBox.Clear()
49         milesDrivenLabel.Text = ""
```

```
50     mileChargeLabel.Text = ""
51     dayChargeLabel.Text = ""
52     discountLabel.Text = ""
53     youOweLabel.Text = ""
54     milesRadioButton.Enabled = True
55     kilometersRadioButton.Enabled = False
56     aaaCheckBox.Enabled = False
57     seniorCheckBox.Enabled = False
58 End Sub
59
60 Private Sub ValidateInput()
61     'Validates the input of the text boxes
62     'All errorMessage values are combined and listed on separate lines
63     'boxFocus determined that the first box with the error message will be selected
64
65     Dim errorMessage As String
66     Dim boxFocus As Object
67     boxFocus = ""
68     errorMessage = ""
69
70     If NameTextBox.Text = "" Then
71         errorMessage = "Customer name may not be blank" & vbNewLine
72         boxFocus = NameTextBox.Focus()
73     End If
74     If AddressTextBox.Text = "" Then
75         errorMessage += "Address may not be blank" & vbNewLine
76         If boxFocus Is "" Then
77             boxFocus = AddressTextBox.Focus()
78         End If
79     End If
80     If CityTextBox.Text = "" Then
81         errorMessage += "City may not be blank" & vbNewLine
82         If boxFocus Is "" Then
83             boxFocus = CityTextBox.Focus()
84         End If
85     End If
86     If StateTextBox.Text = "" Then
87         errorMessage += "State may not be blank" & vbNewLine
88         If boxFocus Is "" Then
89             boxFocus = StateTextBox.Focus()
90         End If
91     End If
92
93     End If
94     Try
95         Integer.Parse(ZipCodeTextBox.Text)
96     Catch ex As Exception
97         errorMessage += "Zip code must be a number" & vbNewLine
98         ZipCodeTextBox.Clear()
99         If boxFocus Is "" Then
100             boxFocus = ZipCodeTextBox.Focus()
```

```
101         End If
102     End Try
103     Try
104         Integer.Parse(BeginOdometerTextBox.Text)
105     Catch ex As Exception
106         errorMessage += "Beginning Odometer Reading must be a number" &
vbNewLine
107         BeginOdometerTextBox.Clear()
108         If boxFocus Is "" Then
109             boxFocus = BeginOdometerTextBox.Focus()
110         End If
111     End Try
112     Try
113         Integer.Parse(EndOdometerTextBox.Text)
114     Catch ex As Exception
115         errorMessage += "Ending Odometer Reading must be a number" &
vbNewLine
116         EndOdometerTextBox.Clear()
117         If boxFocus Is "" Then
118             boxFocus = EndOdometerTextBox.Focus()
119         End If
120     End Try
121     Try
122         Integer.Parse(DaysTextBox.Text)
123     Catch ex As Exception
124         errorMessage += "Number of Days must be a number" & vbNewLine
125         DaysTextBox.Clear()
126         If boxFocus Is "" Then
127             boxFocus = DaysTextBox.Focus()
128         End If
129     End Try
130
131     If errorMessage <> "" Then
132         MessageBox.Show(errorMessage, "Input Error")
133         boxFocus.ToString()
134     Else
135         'If all checked values pass validation the following subs and
functions
136         'will calculate And return values.
137         MilesDriven()
138         MileCharge()
139         DayCharge()
140         Discount()
141         YouOwe()
142         Total()
143         SummaryButton.Enabled = True
144     End If
145
146 End Sub
147 Private Sub SummaryButton_Click(sender As Object, e As EventArgs) Handles
SummaryButton.Click
148     'Displays the number of customers and the daily totals in a messagebox
```

```
149     MessageBox.Show("Daily Totals" & vbNewLine & "Customers = " & _  
        totalcustomers & vbNewLine _  
150         & "Miles = " & totalmiles & vbNewLine _  
151         & "Sales = " & totalsales.ToString("C"))  
152 End Sub  
153  
154 Private Sub ExitButton_Click(sender As Object, e As EventArgs) Handles  
    ExitButton.Click  
155     'Exits the program  
156     If MessageBox.Show("Do you want to Exit?" & vbNewLine & "All Data Will be  
        lost.", "Exit Program?",  
157         MessageBoxButtons.OKCancel) = Windows.Forms.DialogResult.OK Then  
158         Me.Close()  
159     End If  
160  
161 End Sub  
162  
163 Private Function MilesDriven() As Double  
164     'Validates the miles driven and converts if data is in Kilometers.  
165     Try  
166         beginOdometer = Double.Parse(BeginOdometerTextBox.Text)  
167     Catch ex As Exception  
168         MessageBox.Show("Odometer Values must be a number", "Input Error")  
169         BeginOdometerTextBox.Clear()  
170         BeginOdometerTextBox.Focus()  
171     End Try  
172     Try  
173         endOdometer = Double.Parse(EndOdometerTextBox.Text)  
174     Catch ex As Exception  
175         MessageBox.Show("Odometer Values must be a number", "Input Error")  
176         EndOdometerTextBox.Clear()  
177         EndOdometerTextBox.Focus()  
178     End Try  
179     If kilometersRadioButton.Checked = True Then  
180         MilesDriven = ((endOdometer - beginOdometer) * 0.62)  
181     ElseIf milesRadioButton.Checked = True Then  
182         MilesDriven = endOdometer - beginOdometer  
183     End If  
184     milesDrivenLabel.Text = MilesDriven.ToString  
185     'Console.WriteLine(milesDriven)  
186 End Function  
187  
188 Private Function MileCharge() As Double  
189     'Calculates the rate based on miles traveled  
190     If MilesDriven() <= 200 Then  
191         MileCharge = 0  
192     ElseIf MilesDriven() > 200 And MilesDriven() <= 500 Then  
193         MileCharge = (MilesDriven() - 200) * 0.12  
194     ElseIf MilesDriven() > 500 Then  
195         MileCharge = ((MilesDriven() - 500) * 0.1) + (300 * 0.12)  
196     End If  
197     mileChargeLabel.Text = MileCharge.ToString("C")
```

```
198     'Console.WriteLine(mileCharge)
199
200 End Function
201
202 Private Function DayCharge() As Double
203     'Validates the number of days and charges based on a $15 daily charge
204     If Integer.Parse(DaysTextBox.Text) = 0 Or Integer.Parse(DaysTextBox.Text) > 45 Then
205         MessageBox.Show("Days must be greater than 0 and less than 45", "Input Error")
206         DaysTextBox.Clear()
207         DaysTextBox.Focus()
208     Else
209         DayCharge = Integer.Parse(DaysTextBox.Text) * 15.0
210     End If
211     dayChargeLabel.Text = DayCharge.ToString("C")
212     Console.WriteLine(DayCharge)
213 End Function
214
215 Private Function Discount() As Double
216     'Applies the any applicable discounts to the total
217     Discount = 0
218     If aaaCheckBox.Checked = True Then
219         Discount = (0.05 * (MileCharge() + DayCharge()))
220     End If
221     If seniorCheckBox.Checked = True Then
222         Discount += (0.03 * (MileCharge() + DayCharge()))
223     End If
224     discountLabel.Text = Discount.ToString("C")
225     Console.WriteLine(Discount)
226 End Function
227
228 Private Function YouOwe() As Double
229     'Returns the total charge minus any discounts
230     YouOwe = MileCharge() + DayCharge() - Discount()
231     youOweLabel.Text = YouOwe.ToString("C")
232     Console.WriteLine(YouOwe)
233 End Function
234
235 Private Sub Total()
236     totalcustomers += 1
237     totalmiles += MilesDriven()
238     totalsales += YouOwe()
239 End Sub
240
241 End Class
242
```