```
 1  namespace ZebCarRental
 2  {
 3      public partial class Form1 : Form
 4      {
 5
 6          public Form1()
 7          {
 8              InitializeComponent();
 9          }
10
11          private string carType;
12          const string SEDAN = "Sedan";
13          const string SUV = "SUV";
14          const string COMP = "Compact";
15          const double taxMin = 0.0;
16          const double taxMax = 10.0;
17          private string logFile = "Rental Log File.txt";
18          internal string cfgFile = "Configuration.txt";
19          private double sedanRate;
20          private double suvRate;
21          private double compRate;
22          private double taxRate;
23          private double MIN_Rate = -1;
24          const int LISTBOX = 1;
25          const int LOGFILE = 2;
26          const int BOTH = 3;
27          // ica 9, declare form to object
28          private Form2 sf;
29
30          internal double SedanRate
31          {
32              get { return sedanRate; }
33              set
34              {
35                  if (value > MIN_Rate)
36                  {
37                      sedanRate = value;
38                  }
39              }
40
41          }
42          internal double SuvRate
43          {
44              get { return suvRate; }
45              set
46              {
47                  if (value > MIN_Rate)
48                  {
49                      suvRate = value;
```

```
50                  }
51              }
52          }
53          internal double CompRate
54          {
55              get { return compRate; }
56              set
57              {
58                  if (value > MIN_Rate)
59                  {
60                      compRate = value;
61                  }
62              }
63          }
64          private void btnQuit_Click(object sender, EventArgs e)
65          {
66
67              DialogResult ButtonSelected;
68              ButtonSelected = MessageBox.Show(
69                  "Do you really want to Quit?", "Exiting...",
70                  MessageBoxButtons.YesNo,
71                  MessageBoxIcon.Question);
72              if (ButtonSelected == DialogResult.Yes)
73              {
74                  this.Close();
75              }
76          }
77
78          private void btnClear_Click(object sender, EventArgs e)
79          {
80              txtName.Clear();
81              txtDays.Clear();
82              txtRate.Clear();
83              lstOut.Items.Clear();
84              txtName.Focus();
85              rdoSedan.Checked = true;
86          }
87
88          private void btnCalc_Click(object sender, EventArgs e)
89          {
90              //  declare/read variables from txtbx
91              string custName;
92              int totalDays;
93              double rateDaily, costCar;
94              bool totalDaysValid, rateDailyValid;
95              double carTypeRate = 0;
96              custName = txtName.Text.Trim();
97
98              // pretend widget name is a name of a person
```

```
 99              // this code is not required for your project
100              // but you may want to use it if you have a customer name
101
102              string fName, lName;
103              int posSpace;
104
105              posSpace = custName.IndexOf(" ");
106              if (posSpace != -1)
107              {
108                  fName = custName.Substring(0, posSpace);
109                  lName = custName.Substring(posSpace).Trim();
110                  lstOut.Items.Add("First Name is " + fName);
111                  lstOut.Items.Add("Last Name is " + lName);
112              }
113
114
115              //      double rateTax, costTax, totalCost  ;
116              //  input
117              custName = txtName.Text;
118              totalDaysValid = int.TryParse(txtDays.Text, out totalDays);
119              rateDailyValid = double.TryParse(txtRate.Text, out rateDaily);
120              //processing
121              if (totalDaysValid && rateDailyValid)
122              {
123                  switch (carType)
124                  {
125                      case SEDAN:
126                          carTypeRate = sedanRate;
127                          break;
128                      case SUV:
129                          carTypeRate = suvRate;
130                          break;
131                      case COMP:
132                          carTypeRate = compRate;
133                          break;
134                      default:
135                          lstOut.Items.Add("This shouldn't happen.");
136                          break;
137
138
139                  }
140                  costCar = totalDays * rateDaily;
141                  //      costTax = costCar * rateTax;
142                  //      totalCost = costCar + costTax;
143
144                  //  output
145                  outputTrans("*************** Beginning of Transaction " +
                       DateTime.Now.ToString("G") + "  *****************",
                       LOGFILE);
```

```
146                 outputTrans("Customer Name: " + custName,BOTH);
147                 outputTrans("Days entered is: " + totalDays, BOTH);
148                 //      lstOut.Items.Add("Vehicle type selected:");
149                 outputTrans("Selected vehicle type rate: " +
                      rateDaily.ToString("C"), BOTH);
150                 outputTrans("Vehicle rental cost: " + costCar.ToString
                      ("C"), BOTH);
151                 //      lstOut.Items.Add("Tax rate: ");
152                 //      lstOut.Items.Add("Tax charge: ");
153                 //      lstOut.Items.Add("Cost with tax: ");
154
155             }
156
157             else
158             {
159                 if (!totalDaysValid)
160                 {
161                     lstOut.Items.Add("Total Days should be whole
                       number.");
162                 }
163                 if (!rateDailyValid)
164                 {
165                     lstOut.Items.Add("Daily Rate must be typed as
                       '$1.23.");
166                 }
167             }
168
169
170
171
172
173         }
174
175         private void outputTrans (string msg, int outputType)
176
177         {
178             StreamWriter swLog;
179             if (outputType == LISTBOX || outputType == BOTH)
180             {
181                 lstOut.Items.Add(msg);
182             }
183             if (outputType == LOGFILE || outputType == BOTH)
184             {
185                 swLog = File.AppendText(logFile);
186                 swLog.WriteLine(msg);
187                 swLog.Close();
188             }
189         }
190         private void txtDays_Enter(object sender, EventArgs e)
```

```
191         {
192             txtDays.BackColor = Color.Ivory;
193         }
194
195         private void txtDays_Leave(object sender, EventArgs e)
196         {
197             txtDays.BackColor = SystemColors.Window;
198         }
199
200         private void txtRate_Enter(object sender, EventArgs e)
201         {
202             txtRate.BackColor = Color.Ivory;
203         }
204
205         private void txtRate_Leave(object sender, EventArgs e)
206         {
207             txtRate.BackColor = SystemColors.Window;
208         }
209
210         private void txtName_Enter(object sender, EventArgs e)
211         {
212             txtName.BackColor = Color.Ivory;
213         }
214
215         private void txtName_Leave(object sender, EventArgs e)
216         {
217             txtName.BackColor = SystemColors.Window;
218         }
219
220         private void Form1_Load(object sender, EventArgs e)
221         {
222             StreamReader srCFG;
223             //create sf, which is a form2 object
224             sf = new Form2(this);
225
226             rdoSedan.Checked = true;
227             bool fileWasNotFound = true;
228             do
229             {
230                 try
231                 {
232                     srCFG = File.OpenText(cfgFile);
233                     fileWasNotFound = false;
234                     try
235                     {
236                         SedanRate = double.Parse(srCFG.ReadLine());
237                         SuvRate = double.Parse(srCFG.ReadLine());
238                         CompRate = double.Parse(srCFG.ReadLine());
239                         srCFG.Close();
```

```
240                     }
241                     catch (FormatException ex)
242                     {
243                         lstOut.ForeColor = Color.Red;
244                         lstOut.Items.Add("File data  corrupted. Values
                        were set to defaults");
245                         lstOut.Items.Add(ex.Message);
246                         sedanRate = 40;
247                         suvRate = 60;
248                         compRate = 50;
249                         srCFG.Close();
250                     }
251                 }
252                 catch (FileNotFoundException ex)
253                 {
254                     MessageBox.Show(ex.Message + " Please enter a new file
                    name", "File Not Found");
255                     // OFD.InitialDirectory =
256                     OFD.Filter = "Text Files|*.txt| All Files |*.*";
257                     OFD.Title = "Open Configuration File";
258                     OFD.ShowDialog();
259                     cfgFile = OFD.FileName;
260                 }
261             } while (fileWasNotFound);
262 
263         }
264 
265         private void rdoSedan_CheckedChanged(object sender, EventArgs e)
266         {
267             if (rdoSedan.Checked)
268             {
269                 carType = SEDAN;
270             }
271         }
272 
273         private void rdoSUV_CheckedChanged(object sender, EventArgs e)
274         {
275             if (rdoSUV.Checked)
276             {
277                 carType = SUV;
278             }
279         }
280 
281         private void rdoCompact_CheckedChanged(object sender, EventArgs e)
282         {
283             if (rdoCompact.Checked)
284             {
285                 carType = COMP;
286             }
```

```
287         }
288
289         private void settingsToolStripMenuItem_Click(object sender,
              EventArgs e)
290         {
291             sf.txtSedanRate.Text = SedanRate.ToString();
292             sf.txtSUVRate.Text = SuvRate.ToString();
293             sf.txtCompRate.Text = CompRate.ToString();
294             sf.ShowDialog();
295         }
296
297         private void showLogToolStripMenuItem_Click(object sender,
              EventArgs e)
298         {
299             const int MAX_LINES = 2000;
300             string[] logLines = new string[MAX_LINES];
301             StreamReader sr = File.OpenText(logFile);
302             int numLines = 0;
303             while (!sr.EndOfStream)
304             {
305                 logLines[numLines] = sr.ReadLine();
306                 numLines++;
307             }
308             int begin = -2;
309             int end = 4;
310             for (int i = 0; i < numLines; i++)
311             {
312                 if (logLines[i] == "The vehicle type is " + carType)
313                 {
314                     for (int j = i + begin; j <= i + end; j++)
315                     {
316                         lstOut.Items.Add(logLines[j]);
317                     }
318                 }
319
320             }
321
322             double[] grades = new double[MAX_LINES];
323
324
325
326
327             sr.Close();
328         }
329
330         private void numberArrayTestToolStripMenuItem_Click(object sender,
               EventArgs e)
331         {
332             int[] numbers = new int[50];
```

```
333             for (int i = 0; i < 25; i++)
334             {
335                 numbers[i] = i;
336             }
337             lstOut.Items.Add(numbers.Average());
338         }
339     }
340 }
341
```