```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Text;
 4  using System.Data;
 5
 6  /*
 7   * Title:   Lists
 8   * Author:  Paul McKillop
 9   * Date:    17 March 2020
10   * Purpose: Prov ide lists of items for the interface
11   */
12
13  namespace GymTrackingV
14  {
15      public class Lists
16      {
17          internal static string dataPath = @"C:\gymdata.txt";
18
19          /// <summary>
20          /// Get list of machines from the text file database
21          /// </summary>
22          /// <returns>List<string></returns>
23          public static List<string> Machines()
24          {
25
26              //-- Handler variables
27              var path = dataPath;
28              var tempList = new List<string>();
29              var dt = new DataTable();
30
31              //-- Get the data from the database
32              dt = ImportData.GetTextFileData(path);
33
34              //-- Use a try .. catch block to trap errors found
35              try
36              {
37                  //-- Iterate through the data rows to find unique machine
                      names
38                  foreach (DataRow row in dt.Rows)
39                  {
40                      //-- use an object to hold the data from each row of the
                        text file as it is found for processing.
41                      //-- Intellisense will know valid object fields
42                      //-- Use the column indeces to reference the correct Field
                         item in the row of data
43                      var lineData = new MachineData
44                      {
45                          MachineName = row.Field<string>(0),
46                          Level = row.Field<string>(1)
47                      };
48
49                      if (!Utility.StringFound(tempList, lineData.MachineName))
50                      {
51                          tempList.Add(lineData.MachineName);
52                      }
53                  }
```

```
54              }
55              catch (Exception)
56              {
57
58                  throw;
59              }
60
61              //-- return the populated list of MachineName items
62              return tempList;
63          }
64
65          /// <summary>
66          /// Get list of levels from the text file database.
67          /// </summary>
68          /// <returns>List<string></string></returns>
69          public static List<string> Levels()
70          {
71              //-- for structure and purpose comments refwer to procedure
                  Machines() above
72              var path = dataPath;
73              var tempList = new List<string>();
74              var dt = new DataTable();
75
76              //-- Get the data from the database
77              dt = ImportData.GetTextFileData(path);
78
79              //-- Use a try .. catch block to trap errors found
80              try
81              {
82                  //-- Iterate through the data rows to find unique machine
                      names
83                  foreach (DataRow row in dt.Rows)
84                  {
85                      //-- use an object to hold the data from each row of the
                      text file as it is found for processing.
86                      //-- Intellisense will know valid object fields
87                      //-- Use the column indeces to reference the correct Field
                       item in the row of data
88                      var lineData = new MachineData
89                      {
90                          MachineName = row.Field<string>(0),
91                          Level = row.Field<string>(1)
92                      };
93
94                      if (!Utility.StringFound(tempList, lineData.Level))
95                      {
96                          tempList.Add(lineData.Level);
97                      }
98                  }
99              }
100             catch (Exception)
101             {
102
103                 throw;
104             }
105
```

```
106             //-- return the populated list of Level items
107             return tempList;
108         }
109     }
110 }
111
```