```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using System.Data;
5
6  /*
7   * Title:   MachineDataDb
8   * Author:  Paul McKillop
9   * Date:    17 March 2020
10  * Purpose: Get lists from the text file database
11  */
12
13 namespace GymTrackingV
14 {
15     public class MachineDataDb
16     {
17         /// <summary>
18         /// Get MachineData from the text file database
19         /// </summary>
20         /// <returns>DataTable</returns>
21         public static DataTable GetMachineData()
22         {
23             //-- handler variables
24             var path = Lists.dataPath;
25             var dt = new DataTable();
26
27             //--- Use try .. catch to trap errors
28             try
29             {
30                 dt = ImportData.GetTextFileData(path);
31             }
32             catch (Exception)
33             {
34
35                 throw;
36             }
37
38             //-- return the procedure as a DataTable
39             return dt;
40         }
41
42         /// <summary>
43         /// Get all data columns for a particular machine
44         /// </summary>
45         /// <param name="machineName"></param>
46         /// <returns>List<MachineData>()</MachineData></returns>
47         public static List<MachineData> GetIndividualMachineData(string
           machineName)
48         {
49             var tempList = new List<MachineData>();
50
51             var dt = MachineDataDb.GetMachineData();
52
53             foreach (DataRow row in dt.Rows)
54             {
55                 var data = new MachineData()
```

```
56                  {
57                      MachineName = row.Field<string>(0),
58                      Level = row.Field<string>(1),
59                      Rate = float.Parse(row.Field<string>(2))
60                  };
61
62                  if (data.MachineName == machineName)
63                  {
64                      tempList.Add(data);
65                  };
66              }
67
68              return tempList;
69          }
70
71          /// <summary>
72          /// Get the rate for a machine and level combination
73          /// </summary>
74          /// Params supplied at runtime
75          /// <param name="machineName">string</param>
76          /// <param name="level">string</param>
77          /// <returns>int</returns>
78          public static int GetRate(string machineName, string level)
79          {
80              float tempRate = 0;
81
82              List<MachineData> machineData = GetIndividualMachineData
                    (machineName);
83
84              foreach (MachineData data in machineData)
85              {
86                  if (data.Level == level)
87                  {
88                      tempRate = data.Rate;
89                  }
90              }
91
92              return Convert.ToInt32(tempRate);
93          }
94      }
95  }
96
```