```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ProductPerformance
{
    public class Lists
    {
        //-- Product Types - no filter required
        public static List<string> GetProductTypes()
        {

            //- Return the method
            return ProductCatalogueDB.GetProductTypes();
        }

        //-- Products - filter by ProductType
        public static List<string> GetProductsByType(string productType)
        {
            //-- return the method
            return ProductCatalogueDB.ProductsByType(productType);
        }

        #region Short Person Data from Database

        /// <summary>
        /// Short Data of Persons
        /// </summary>
        /// <returns>List of stringsreturns>
        public static List<string> ShortDataAllPersonsInDb()
        {
            //-- Create list to hole items to be returned
            List<string> tempList = new List<string>();

            //-- Loop through data in the text file
            using (StreamReader reader = new StreamReader(@"D:\persons.txt"))
            {
                //-- loop while data in reader
                while (true)
                {
                    //-- Get data in the line
                    string line = reader.ReadLine();
                    //-- drop if no line data
                    if (line == null)
                    {
                        break;
                    }

                    //-- split the line by comma separation
                    string[] fields = line.Split(',');
                    //-- initialise person object
                    Person person = new Person()
                    {
```

```
57                          Forename = fields[0],
58                          Surname = fields[1],
59                          Postcode = fields[4]
60                      };
61                      //- get short data as string
62                      string personShort = person.PersonShortData();
63                      //-- Add tto list
64                      tempList.Add(personShort);
65                  }
66
67                  //-- Return the method
68                  return tempList;
69              }
70          }
71
72          #endregion
73
74          #region String is found in list
75
76          /// <summary>
77          /// Check if a string is already in a list
78          /// </summary>
79          /// <param name="listToSearch"></param>
80          /// <param name="stringToFind"></param>
81          /// <returns>Boolean</returns>
82          public static bool StringFound(List<string> listToSearch, string
              stringToFind)
83          {
84              //-- tracker variable
85              bool stringFound = false;
86              //-- Loop through all
87              foreach (string value in listToSearch)
88              {
89                  if (value == stringToFind)
90                  {
91                      stringFound = true;
92                      return stringFound;
93                  }
94              }
95
96              //-- return true or false: in list, or not
97              return stringFound;
98          }
99
100         #endregion
101
102         //-------------------------------------------------------------------
              -
103         //-- Static lists follow: Items do not change
104         //-- These are what are called (hard-code) literals
105         //-- Not best practice but quicker
106         //--
              ------------------------------------------------------------------
107
108         public static List<string> ProductSizes()
109         {
```

```
110             List<string> sizes = new List<string>();
111
112             sizes.Add("250ml");
113             sizes.Add("500ml");
114             sizes.Add("750ml");
115             sizes.Add("1 Litre");
116             sizes.Add("Other");
117             return sizes;
118         }
119
120         //-- Observation 1 is mandatory so 'none' is not an option
121         public static List<string> ObservationOneOptions()
122         {
123             //-- Working list for string to be returned
124             List<string> tempList = new List<string>
125             {
126                 "Very poor",
127                 "Poor",
128                 "Okay",
129                 "Quite good",
130                 "Good",
131                 "Works well",
132                 "Works very well",
133                 "Really good",
134                 "Excellent",
135                 "Other"
136             };
137
138             //- Return the method
139             return tempList;
140         }
141
142         //-- Other observations list
143         public static List<string> ObservationOtherOptions()
144         {
145             //-- Working list for string to be returned
146             List<string> tempList = new List<string>
147             {
148                 "None",
149                 "Very poor",
150                 "Poor",
151                 "Okay",
152                 "Quite good",
153                 "Good",
154                 "Works well",
155                 "Works very well",
156                 "Really good",
157                 "Excellent",
158                 "Other"
159             };
160
161             //- Return the method
162             return tempList;
163         }
164     }
165 }
```