```
//-- **************************
//-- CLASS:     ObservationDB
//-- AUTHOR:    Paul McKillop
//-- CREATED:   19 January 2019
//-  PURPOSE:   Retrieve and store Observation data
//-- **************************


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

namespace ProductPerformancePrototype
{
    public class ObservationDB
    {
        /// <summary>
        /// Write an observation class dataset to the text file
        /// </summary>
        /// <param name="myObservation"></param>
        /// <returns>Bool to indicate success of the write process</returns>
        public static bool WriteObservation(Observation myObservation)
        {
            //-- Get the data into the right format for writing to text file
            string observationData = myObservation.AllObervationData();

            //-- Use StreamWriter to write the data
            using (StreamWriter writer = new StreamWriter(@"D:
              \observations.txt", true))
            {
                //-- Write the line of data
                writer.WriteLine(observationData);

                //-- Close the StreamWriter
                writer.Close();
                //-- Dispose of it from memory. Shouldn't be needed
                //-- because of the 'using()' method but extra precaution
                writer.Dispose();

            }
            //-- got this far
            return true;
        }



        //-- Import the data in the text file into a DataTable
        public static DataTable GetAllObservationsData()
        {
            //-- Object to hold the data
            var dt = new DataTable();

            //-- Get it
            dt = ImportData.GetTextFileData(@"D:\observations.txt");

            //-- Return it in the method
            return dt;
        }


        //-- Get all observations unformatted
        public static List<string> GetAllObservationsUnformatted()
        {
            List<string> tempList = new List<string>();

            //-- object for data
            var dt = new DataTable();

            //-- Populate
            dt = ImportData.GetTextFileData(@"D:\observations.txt");

            //-- loop and fill the list
            foreach (DataRow row in dt.Rows)
            {
                var currentObservation = new Observation()
                {
                    Person = row.Field<string>(0),
                    ProductType = row.Field<string>(1),
                    Product = row.Field<string>(2),
                    ProductSize = row.Field<string>(3),
                    FirstUse = row.Field<string>(4),
                    Continuous = row.Field<string>(5),
                    Observation1 = row.Field<string>(6),
                    Observation2 = row.Field<string>(7),
                    Observation3 = row.Field<string>(8)
                };

                //-- add
                tempList.Add(currentObservation.AllObervationData());
            }

            return tempList;
        }


        public static List<string> GetAllObservationsShort()
        {
            List<string> tempList = new List<string>();

            //-- object for data
            var dt = new DataTable();

            //-- Populate
            dt = ImportData.GetTextFileData(@"D:\observations.txt");

            //-- loop and fill the list
            foreach (DataRow row in dt.Rows)
            {
                var currentObservation = new Observation()
                {
                    Person = row.Field<string>(0),
                    ProductType = row.Field<string>(1),
                    Product = row.Field<string>(2),
                    ProductSize = row.Field<string>(3),
                    FirstUse = row.Field<string>(4),
                    Continuous = row.Field<string>(5),
                    Observation1 = row.Field<string>(6),
                    Observation2 = row.Field<string>(7),
                    Observation3 = row.Field<string>(8)
                };

                //-- add
                tempList.Add(currentObservation.ObservationDataShort());
            }

            return tempList;
        }

























        public static List<string> ObservationsFormattedPasting()
        {
            List<string> tempList = new List<string>();

            return tempList;
        }

        public static List<string> ObservationsUnformattedPasting()
        {
            List<string> tempList = new List<string>();

            return tempList;
        }
    }
}

```