```
/*-------------------------------------------------------------------------------------------
 * Name:        Dan Cassidy
 * Date:        2015-06-02
 * Assignment:  cView-P1
 * Source File: CViewData.cs
 * Class:       CSCI-C 490, C# Programming, MoWe 08:00
 * Purpose:     Contains the basic class for the cView program, along with some supporting methods.
-------------------------------------------------------------------------------------------*/

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

namespace cView_P1_DanCassidy
{
    class CViewData
    {
        //Basic properties of the class.
        public string Name { get; set; }
        public string Address { get; set; }
        public string City { get; set; }
        public string State { get; set; }
        public string ZIPCode { get; set; }
        public string PhoneNumber { get; set; }

        //Easily accessible string showing the data order in the ToString() method.
        private const string HEADER = "Business Name, Address, City, State, ZIP Code [Phone Number]";

        //Read-only accessor for the Header property that just uses the HEADER constant.
        public static string Header
        {
            get
            {
                return HEADER;
            }
        }

        /*-------------------------------------------------------------------------------------------
         * Method:  Contains
         * Purpose: Search this object for a string, optionally with case sensitivity.
         * Input:   string toSearchFor, representing the string that will be searched for.
         * Input:   (Optional) bool caseInsensitive, determines whether the search will be case
         *          sensitive or case insensitive.  Default is case insensitive.
         * Output:  bool representing whether the specified string was found in the object.
        -------------------------------------------------------------------------------------------*/
        public bool Contains(string toSearchFor, bool caseInsensitive = true)
        {
            //Determine whether to use case sensitive or insensitive searching.
            switch (caseInsensitive)
            {
                case false:
                    //Case sensitive searching.
                    if (Name.Contains(toSearchFor) || Address.Contains(toSearchFor) ||
                        City.Contains(toSearchFor) || State.Contains(toSearchFor) ||
                        ZIPCode.Contains(toSearchFor) || PhoneNumber.Contains(toSearchFor))
                    {
                        //Found it.
                        return true;
                    }
                    break;

                case true:
                default:
                    //Case insensitive searching. Basic code idea from Stack Overflow.
```

```
67                      //http://stackoverflow.com/a/444818
68                      if (Name.IndexOf(toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0 ||
69                          Address.IndexOf(toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0 ||
70                          City.IndexOf(toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0 ||
71                          State.IndexOf(toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0 ||
72                          ZIPCode.IndexOf(toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0 ||
73                          PhoneNumber.IndexOf(toSearchFor, StringComparison.OrdinalIgnoreCase) >= 0)
74                      {
75                          //Found it.
76                          return true;
77                      }
78                      break;
79              }
80
81              //If the specified string cannot be found in this object, return false.
82              return false;
83          }
84
85          /*-------------------------------------------------------------------------------------------
86           * Method:  ToString
87           * Purpose: Override of the ToString() method. Formats the return value so it looks pretty.
88           * Input:   Nothing
89           * Output:  String object containing serialized object data.
90          -------------------------------------------------------------------------------------------*/
91          public override string ToString()
92          {
93              return Name + ", " + Address + ", " + City + ", " +
94                  State + ", " + ZIPCode + " [" + PhoneNumber + "]";
95          }
96      }
97  }
98
```