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

namespace ClassLibrary
{
    public static class ThermometerClass
    {
        // private static fields, so reference can only be done in the library.
        private static double _CelsiusTemperature;
        private static double _FahrenheitTemperature;

        // static constructor sets the original values of temperatures.
        static ThermometerClass()
        {
            _CelsiusTemperature = 0;
            _FahrenheitTemperature = 32;
        }

        //========================= Get Temperatures / Set Temperature Values =========================
        /* two static properties, one called CelsiusTemperature  and the other FahrenheitTemperature
           (gets the field values and returns new set field values).*/
        public static double CelsiusTemperature
        {
            get
            { return _CelsiusTemperature; }

            set
            { _CelsiusTemperature = value; }
        }

        public static double FahrenheitTemperature
        {
            get
             { return _FahrenheitTemperature; }

            set
            { _FahrenheitTemperature = value; }
        }
```

```csharp
//========================= Add / Pass Temperatures to convertToDouble =========================
/* Two static methods, one called AddCelsius  and the other AddFahrenheit, both of which take in a
   STRING. (Call convert to double, then append temperature, then call to change the other
   temperature.)*/
public static void AddCelsius(string CelsiusTemperature1)
{
    CelsiusTemperature += ConvertInputToDouble(CelsiusTemperature1);
    CelsiusToFahrenheit(CelsiusTemperature.ToString());

}

public static void AddFahrenheit(string FahrenheitTemperature1)
{
    FahrenheitTemperature += ConvertInputToDouble(FahrenheitTemperature1);
    FahrenheitToCelsius(FahrenheitTemperature.ToString());

}

//========================= Convert to Double / Catch Exceptions =============================
/* Initialize and parse the variable temp to double, catch the exeption while displaying
   text in the exception box (message is sent to the console program.cs), console window
   and return the current temperatures.*/
private static double ConvertInputToDouble(string pTempString)
{
    double temp = 0.0;

    try
      { temp = Double.Parse(pTempString); }

    catch (Exception e)
      { throw new ApplicationException("That was not a valid temperature."); }

    return temp;
}

//================== Calculate ConvertTodouble, CelsiusTemperature1 with formula ==================
/* Convert the string to a double, and BOTH of these 2 methods update BOTH of the fields.
   Call Method ConvertInputToDouble to convert to double while calculating temperatures.*/
private static void CelsiusToFahrenheit(string CelsiusTemperature1)
   { FahrenheitTemperature = (ConvertInputToDouble(CelsiusTemperature1) * 9 / 5) + 32; }

private static void FahrenheitToCelsius(string FahrenheitTemperature)
   { CelsiusTemperature = (ConvertInputToDouble(FahrenheitTemperature) - 32) * 5 / 9; }
    }
}
```

```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ClassLibrary;

namespace Thermometer
{
    class Program
    {
        static void Main(string[] args)
        {
            /*Write a test class that creates an instance of a thermometer,
              displays the current Celsius and Fahrenheit temperature.*/

            Console.Write("\nThis is a test of the Thermometer Class\n");

            //======================= Set 1 c = 0, f = 32 ===============================
            Console.Write("\nThe current temp is: {0}\u00B0C\nThe current temp is: {1}\u00B0F\n"
                        , ThermometerClass.CelsiusTemperature, ThermometerClass.FahrenheitTemperature);

            //======================= Set 2 c = 50, f = 122 =============================
            // Adds 50 degrees Celsius to it and then displays the Celsius and Fahrenheit temperature.
            ThermometerClass.AddCelsius("50"); // 0 + 50 = 50c
            Console.Write("\nThe current temp is: {0}\u00B0C\nThe current temp is: {1}\u00B0F\n"
                        , ThermometerClass.CelsiusTemperature, ThermometerClass.FahrenheitTemperature);

            //======================= Set 3 c = 36.11111111, f = 97 =====================
            // Adds -25 degrees Fahrenheit to it and then displays the Celsius and Fahrenheit temperature.
            ThermometerClass.AddFahrenheit("-25"); // 122 -25 = 97f
            Console.Write("\nThe current temp is: {0}\u00B0C\nThe current temp is: {1}\u00B0F\n"
                        , ThermometerClass.CelsiusTemperature, ThermometerClass.FahrenheitTemperature);


            //=========================================================================//
```

```csharp
                    //==================== Try Catch with current values displayed =================
                    /* Allow the user to change the value. Call the Methods ThermometerClass.CelsiusTemperature,
                       FahrenheitTemperature in the ClassLibrary to get the current temperature states while writing
                       them out to the console. Use try catch to catch any exceptions and throw an application
                       exception.*/
                    try
                    {
                        Console.Write("\nEnter a Celsius value to change the temperature by: ");
                        ThermometerClass.AddCelsius(Console.ReadLine());
                        Console.Write("\nThe current temp is: {0}\u00B0C\nThe current temp is: {1}\u00B0F\n" +
                                    "\nPress any key to exit the program."
                                , ThermometerClass.CelsiusTemperature, ThermometerClass.FahrenheitTemperature);

                        Console.ReadKey(); // Pause before ending.

                    }/* If so, Catch the exception and write out the current temperature state that was not changed
                        (ThermometerClass.Method). Send the Exception Message from the ClassLibrary to the string
                        And Application Exception box that is being thrown. Display the current values of both
                        Celsius and Fahrenheit Temperatures.*/
                    catch (Exception e)
                    {
                        Console.Write("\n"+e.Message+"\nThe temperature setting was not changed.\n"+
                                    "The current temp is: {0}\u00B0C \nThe current temp is: {1}\u00B0F"
                                , ThermometerClass.CelsiusTemperature, ThermometerClass.FahrenheitTemperature);
                        {
                            throw new ApplicationException(e.Message); /* Solve the exception by throwing a
                                                                         new application exception.*/
                        }
                    }

                }

            }

        }
```