Read/Write Text Files in C#

There are various ways to read and write text files using C#. The following examples shows how to use methods in the **System.IO.FILE** class to read and write text files.

```
class ReadFromFile
{
    static void Main()
    {
        // The files used in this example are created in the topic
        // How to: Write to a Text File. You can change the path and
        // file name to substitute text files of your own.

        // Example #1
        // Read the file as one string.
        string text = System.IO.File.ReadAllText(@"C:\Users\Public\TestFolder\WriteText.txt");

        // Display the file contents to the console. Variable text is a string.
        System.Console.WriteLine("Contents of WriteText.txt = {0}", text);

        // Example #2
        // Read each line of the file into a string array. Each element
        // of the array is one line of the file.
        string[] lines = System.IO.File.ReadAllLines(@"C:\Users\Public\TestFolder\WriteLines2.txt");

        // Display the file contents by using a foreach loop.
        System.Console.WriteLine("Contents of WriteLines2.txt = ");
        foreach (string line in lines)
        {
            // Use a tab to indent each line of the file.
            Console.WriteLine("\t" + line);
        }

        // Keep the console window open in debug mode.
        Console.WriteLine("Press any key to exit.");
        System.Console.ReadKey();
    }
}
```

```
class WriteTextFile
{
    static void Main()
    {

        // These examples assume a "C:\Users\Public\TestFolder" folder on your machine.
        // You can modify the path if necessary.

        // Example #1: Write an array of strings to a file.
        // Create a string array that consists of three lines.
        string[] lines = { "First line", "Second line", "Third line" };
        // WriteAllLines creates a file, writes a collection of strings to the file,
        // and then closes the file.  You do NOT need to call Flush() or Close().
        System.IO.File.WriteAllLines(@"C:\Users\Public\TestFolder\WriteLines.txt", lines);

        // Example #2: Write one string to a text file.
        string text = "A class is the most powerful data type in C#. Like a structure, " +
                      "a class defines the data and behavior of the data type. ";
        // WriteAllText creates a file, writes the specified string to the file,
        // and then closes the file.    You do NOT need to call Flush() or Close().
        System.IO.File.WriteAllText(@"C:\Users\Public\TestFolder\WriteText.txt", text);

        // Example #3: Write only some strings in an array to a file.
        // The using statement automatically flushes AND CLOSES the stream and calls
        // IDisposable.Dispose on the stream object.
        // NOTE: do not use FileStream for text files because it writes bytes, but StreamWriter
        // encodes the output as text.
        using (System.IO.StreamWriter file =
            new System.IO.StreamWriter(@"C:\Users\Public\TestFolder\WriteLines2.txt"))
        {
            foreach (string line in lines)
            {
                // If the line doesn't contain the word 'Second', write the line to the file.
                if (!line.Contains("Second"))
                {
                    file.WriteLine(line);
                }
            }
        }

        // Example #4: Append new text to an existing file.
        // The using statement automatically flushes AND CLOSES the stream and calls
        // IDisposable.Dispose on the stream object.
        using (System.IO.StreamWriter file =
            new System.IO.StreamWriter(@"C:\Users\Public\TestFolder\WriteLines2.txt", true))
        {
            file.WriteLine("Fourth line");
        }
    }
}
```