File I/O

Introduction

File I/O refers to the reading (input) and writing (output) of data to a file that is stored on the hard disk of your computer. C# treats a file as a sequential **stream** of bytes. C# contains a number of stream classes that are used for I/O operations. In this lab we will introduce two of these classes, the **StreamReader** class for reading text data from a file and the **StreamWriter** class for writing text data to a file.

The StreamReader class and the StreamWriter class contain methods that work identically to the way that they do for the Console class. For example, the StreamReader class uses the **ReadLine( )** method to read data from a file, in exactly the same way that the ReadLine( ) method reads data from the Console.

Declaring and Opening a Stream

Declaring a Stream Object: You declare a stream object just as you do any other variable. For example, to create a StreamReader object you would write

```
StreamReader theTextFile = new StreamReader;
```

In this case a StreamReader object is created and can be referenced by the variable *theTextFile*.

Opening a Stream: Before you can read from a file or write to a file, you must open the file. This can be done when the object is declared, by writing

```
StreamReader theTextFile = new
StreamReader("someData.txt");
```

The parameter *someData.txt* is the path to the file. If no path is given, then the file is assumed to be in the same directory as the program that you are executing. If the path that you use as a parameter contains a backslash, you must use two backslashes, for example

```
StreamReader theTextFile = new
StreamReader("c:\\someData.txt");
```

Reading from a File

Once the file has been opened, you read from the file just as you would from the console. All of the same rules apply. For example,

```
     StreamReader theTextFile = new
StreamReader("someData.txt");
     int number;
     double size;
     number = int.Parse(theTextFile.ReadLine( ) );
     size = double.Parse(theTextFile.ReadLine( ) );
```