C# StreamReader is used to read characters to a stream in a specified encoding. StreamReader.Read method reads the next character or next set of characters from the input stream. StreamReader is inherited from TextReader that provides methods to read a character, block, line, or all content.

StreamReader is defined in the System.IO namespace. StreamReader provides the following methods:

1. Peak – Returns if there is a character or not.
2. Read - Reads the next character or next set of characters from the input stream.
3. ReadAsync - Reads a specified maximum number of characters from the current stream asynchronously and writes the data to a buffer, beginning at the specified index.
4. ReadBlock - Reads a specified maximum number of characters from the current stream and writes the data to a buffer, beginning at the specified index.
5. ReadBlockAsync - Reads a specified maximum number of characters from the current stream asynchronously and writes the data to a buffer, beginning at the specified index.
6. ReadLine - Reads a line of characters from the current stream and returns the data as a string.
7. ReadLineAsync - Reads a line of characters asynchronously from the current stream and returns the data as a string.
8. ReadToEnd - Reads all characters from the current position to the end of the stream.
9. ReadToEndAsync - Reads all characters from the current position to the end of the stream asynchronously and returns them as one string.

using System;

using System.IO;

using System.Text;

class Program

{

**static** void Main(string[] args)

{

// File **name**

string fileName = @"C:\Temp\Demos.txt";

try

{

// **Create** a StreamReader

using (StreamReader reader = new StreamReader(fileName))

{

string line;

// **Read** line **by** line

while ((line = reader.ReadLine()) != null)

{

Console.WriteLine(line);

}

}

}

catch (Exception exp)

{

Console.WriteLine(exp.Message);

}

Console.ReadKey();

}

}

1. FileInfo fi = **new** FileInfo(fileName);
2. StreamReader sr = fi.OpenText();
3. **string** s = "";
4. **while** ((s = sr.ReadLine()) != **null**) {
5. Console.WriteLine(s);
6. }
7. // Create a StreamReader from a FileStream
8. **using** (StreamReader reader = **new** StreamReader(**new** FileStream(fileName, FileMode.Open)))
9. {
10. **string** line;
11. // Read line by line
12. **while** ((line = reader.ReadLine()) != **null**)
13. {
14. Console.WriteLine(line);
15. }
16. }
17. **using** (StreamReader reader = **new** StreamReader(**new** FileStream(fileName, FileMode.Open)))
18. {
19. **while** (reader.Peek() >= 0)
20. {
21. // Read here
22. }
23. }

StreamWriter class in C# writes characters to a stream in a specified encoding. StreamWriter.Write() method is responsible for writing text to a stream. StreamWriter class is inherited from TextWriter class that provides methods to write an object to a string, write strings to a file, or to serialize XML.

StreamWriter is defined in the System.IO namespace. StreamWriter provides the following Write methods,

1. Write – Writes data to the stream.
2. WriteAsync - Writes data to the stream asynchronously.
3. WriteLine – Writes a line terminator to the text string or stream.
4. WriteLineAsync - Writes a line terminator to the text string or stream asynchronously.
5. Creating a StreamWriter using a Filename

StreamWriter constructor takes a file name or a FileStream object with Encoding, and buffer size as optional parameters.

The following code snippet creates a StreamWriter from a filename with default encoding and buffer size.

1. // File name
2. **string** fileName = @"C:\Temp\Mahesh.txt";
3. StreamWriter writer = **new** StreamWriter(fileName);

The following code snippet creates a StreamWriter and adds some text to the writer using StreamWriter.Write method. If the file does not exist, the writer will create a new file. If the file already exists, the writer will override its content.

1. **string** fileName = @"C:\Temp\CSharpAuthors.txt";
2. **try**
3. {
4. **using** (StreamWriter writer = **new** StreamWriter(fileName))
5. {
6. writer.Write("This file contains C# Corner Authors.");
7. }
8. }
9. **catch**(Exception exp)
10. {
11. Console.Write(exp.Message);
12. }
13. **using** System;
14. **using** System.IO;
15. **using** System.Text;
16. **class** Program
17. {
18. **static** **void** Main(**string**[] args)
19. {
20. // File name
21. **string** fileName = @"C:\Temp\CSharpAuthors.txt";
22. FileStream stream = **null**;
23. **try**
24. {
25. // Create a FileStream with mode CreateNew
26. stream = **new** FileStream(fileName, FileMode.OpenOrCreate);
27. // Create a StreamWriter from FileStream
28. **using** (StreamWriter writer = **new** StreamWriter(stream, Encoding.UTF8 ))
29. {
30. writer.WriteLine("C# Corner Authors");
31. writer.WriteLine("==================");
32. writer.WriteLine("Monica Rathbun");
33. writer.WriteLine("Vidya Agarwal");
34. writer.WriteLine("Mahesh Chand");
35. writer.WriteLine("Vijay Anand");
36. writer.WriteLine("Jignesh Trivedi");
37. }
38. }
39. **finally**
40. {
41. **if** (stream != **null**)
42. stream.Dispose();
43. }
44. // Read a file
45. **string** readText = File.ReadAllText(fileName);
46. Console.WriteLine(readText);
47. Console.ReadKey();
48. }
49. }

<https://www.knowledgehut.com/tutorials/csharp/csharp-file-handling>