



CS 2530

Files / Streams



Files and Streams

- **File:**
ordered and named sequence of bytes having persistent storage. Ends with end-of-file marker
- **Stream:**
provide a way to write and read bytes to and from a backing store



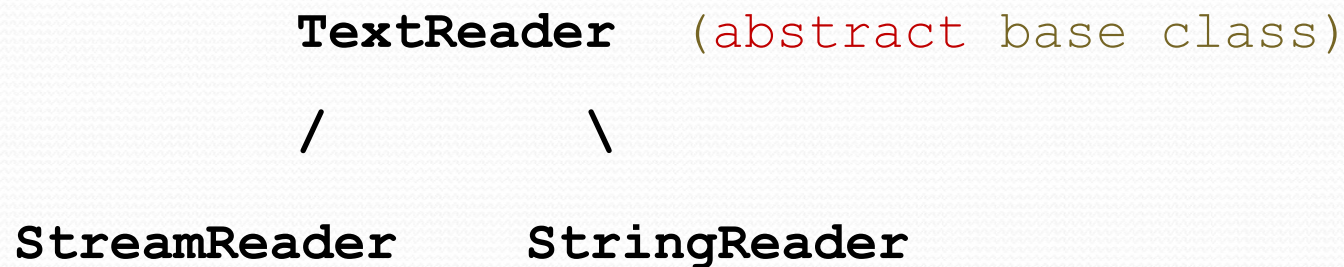
Files and Streams

- When a Console Application runs the runtime environment creates 3 streams:
 - **Console.In**
 - **Console.Out**
 - **Console.Error**

text input / output

Namespace: System.IO

- **StreamReader** .. Text input from file
- **StreamWriter** .. Text output to a file
- **StringReader / StringWriter** .. Text input from / to string





Example StreamReader

```
string line;
```

```
using (StreamReader reader = new StreamReader(filePath))  
{  
    while ((line = reader.ReadLine()) != null)  
    {  
        System.WriteLine(line);  
    }  
}
```

Example StreamReader

```
string line;
```



Using statement

```
using (StreamReader reader = new StreamReader(filePath))  
{  
    while ((line = reader.ReadLine()) != null)  
    {  
        System.WriteLine(line);  
    }  
}
```


Example StreamReader

```
string line;
```

Using statement

```
using (StreamReader reader = new StreamReader(filePath))  
{  
    System.WriteLine(reader.ReadToEnd());  
}
```



System.IO

- **File**

provides static methods for the creation, copying, deletion, moving, and opening of files

- **FileInfo**

provides instance methods for the creation, copying, deletion, moving, and opening of files



System.IO

- **Directory**

provides static methods for creating, moving, and enumerating through directories and subdirectories

- **DirectoryInfo**

provides instance methods for creating, moving, and enumerating through directories and subdirectories



System.IO

- **Path**

performs operations on string instances that contain file or directory path information

- **GetFileName**
- **GetDirectoryName**
- **ChangeExtension**



Dialog to select files / directories:

Namespace: `Microsoft.Win32`

- **OpenFileDialog:**

Prompts the user to open a file with a dialog box

- **FolderBrowserDialog**

Prompts the user to select a folder



Example OpenFileDialog

```
OpenFileDialog dialog = new OpenFileDialog();

if (dialog.ShowDialog() == true)    // user clicked OK button
{
    string filePath = dialog.FileName;

    using (StreamReader reader = new StreamReader(filePath))
    {
        fileContentTb.Text = reader.ReadToEnd();
    }
}
```


Example OpenFileDialog

```
OpenFileDialog dialog = new OpenFileDialog();
```

```
if (dialog.ShowDialog() == true)    // user clicked OK button  
{
```

```
    string filePath = dialog.FileName;
```

Returns file
name incl. path

```
    using (StreamReader reader = new StreamReader(filePath))
```

```
    {
```

```
        fileContentTb.Text = reader.ReadToEnd();
```

```
    }
```

```
}
```



Using Statement

- Convenient syntax that ensures correct use of **IDisposable** objects.
- All types that encapsulate unmanaged resources (e.g. files, fonts) must implement IDisposable

```
using (Font font1 = new Font("Arial", 10.0f))  
{  
    byte charset = font1.GdiCharSet;  
}
```




Using Statement

- The **using** statement :
 - **calls** the **Dispose** method (even if exception occurs)
 - **causes** the **object to go out of scope** as soon as **Dispose** is called.
- Within the **using** block, the object is **read-only** and cannot be modified or reassigned.