I/O

- All langages have different libraries to handle I/O

- Two kinds of I/O
  - Raw bytes
  - Interpreted data

I/O - Interpreted

- Stream classes
  - StreamReader
  - StreamWriter

- StreamReader/Writer

  Implements a TextReader that reads/writes characters from/to a byte stream in a particular encoding

```
try
{
    // Create an instance of StreamReader to read from a file.
    // The using statement also closes the StreamReader.
    using (StreamReader sr = new StreamReader("TestFile.txt"))
    {
        string line;
        // Read and display lines from the file until the end of
        // the file is reached.
        while ((line = sr.ReadLine()) != null)
        {
            Console.WriteLine(line);
        }
    }
}
catch (Exception e)
{
    // Let the user know what went wrong.
    Console.WriteLine("The file could not be read:");
    Console.WriteLine(e.Message);
}
```

```
// Get the directories currently on the C drive.
DirectoryInfo[] cDirs = new DirectoryInfo(@"c:\").GetDirectories();

// Write each directory name to a file.
using (StreamWriter sw = new StreamWriter("CDriveDirs.txt"))
{
    foreach (DirectoryInfo dir in cDirs)
    {
        sw.WriteLine(dir.Name);

    }
}

// Read and show each line from the file.
string line = "";
using (StreamReader sr = new StreamReader("CDriveDirs.txt"))
{
    while ((line = sr.ReadLine()) != null)
    {
        Console.WriteLine(line);
    }
}
```

I/O – Raw

- BinaryReader and BinaryWriter
- Read/write data instead of characters
- Exposed methods can serialize data into bytes and deserialize bytes into data !
  - ReadInt32
  - ReadString
  - ReadBoolean
  - ...

```
const string fileName = "AppSettings.dat";

static void Main()
{
    WriteDefaultValues();
    DisplayValues();
}

public static void WriteDefaultValues()
{
    using (BinaryWriter writer = new BinaryWriter(File.Open(fileName, FileMode.Creat
    {
        writer.Write(1.250F);
        writer.Write(@"c:\Temp");
        writer.Write(10);
        writer.Write(true);
    }
}
```

```
public static void DisplayValues()
{
    float aspectRatio;
    string tempDirectory;
    int autoSaveTime;
    bool showStatusBar;

    if (File.Exists(fileName))
    {
        using (BinaryReader reader = new BinaryReader(File.Open(fileName, FileMode.C
        {
            aspectRatio = reader.ReadSingle();
            tempDirectory = reader.ReadString();
            autoSaveTime = reader.ReadInt32();
            showStatusBar = reader.ReadBoolean();
        }

        Console.WriteLine("Aspect ratio set to: " + aspectRatio);
        Console.WriteLine("Temp directory is: " + tempDirectory);
        Console.WriteLine("Auto save time set to: " + autoSaveTime);
        Console.WriteLine("Show status bar: " + showStatusBar);
    }
}
```