File Input and Output

File I/O

- https://cplusplus.com/doc/tutorial/files/
- File Input/Output (I/O) is reading from and writing to files.
- There are 3 basic classes to handle files:
  - **ifstream**: reading files
  - **ofstream**: writing files
  - **fstream**: reading/writing files

File I/O

- Use `#include <fstream>`
- Pass the path to the file to the constructors to open the file.
- EXAMPLES:
  - `std::ifstream inputStream("inputFile.txt");`
  - `std::ofstream outputStream("outputFile.txt");`

File I/O

- There are several ways to open a file: input, output, binary, append, truncate, at the end of the file.
- The default open modes:
  - **ifstream**: ios::in (input)
  - **ofstream**: ios::out (output)
  - **fstream**: ios::in | ios::out (both input and output)

3 Steps for working with files

1. Open the file
   - Either pass the file path to the **constructor** or call the **open** method.
2. Read/write the file
3. Close the file
   - Happens automatically when the variable goes out of scope OR call the **close** method.

File Input and Output

File Paths

- The **full path** to the file.
- Example:

$$\underbrace{\text{C:}\ \backslash}_{\text{Drive}}\ \underbrace{\text{Directory1}\ \backslash\ \text{Directory2}\ \backslash}_{\text{Directory Path}}\ \underbrace{\text{filename.extension}}_{\text{File name}}$$

File Paths

- Full Path
  C: \ temp \ 2109 \ sample.txt
- Relative Path
  relative to the working directory
  ..\..\Files\Config\sample.txt
- Current Directory
  the current directory of the application
  sample.txt

File Input and Output

CSV Data

CSV Data

- **CSV**: **C**omma-**S**eparated **V**alues
- CSV is a way to store data by separating the data inside the file with a char called a delimiter. It does NOT have to be a comma. It can be any character you choose.
  - “Thor,Captain America,Iron Man” (the delimiter is the , character)
  - “Thor|Captain America|Iron Man” (the delimiter is the | character)

File Input and Output

Writing CSV Data

Writing CSV Data

1. Open the file
   - `std::ofstream file("myData.csv");`
2. Write to the file
   - `char delimiter = '|';`
   - `file << "Batman!" << delimiter << 5 << delimiter << 13.7 << "\n";`
3. Close the file
   - `file.close();`

File Input and Output

Reading CSV Data

Reading CSV Data

- You can read a line from the file using std::getline.
- Add `#include <string>`

Reading CSV Data

1. Open the file
   - `std::ifstream inFile("myData.csv");`
2. Read the file
   - `std::string line;`
   - `std::getline(inFile, line); //reads 1 line from the file`
3. Close the file
   - `inFile.close();`

File Input and Output

Parsing CSV Data

Parsing CSV Data

- std::getline will read 1 line from the file and store it in a std::string.
- If the line is csv data, then you need to parse the string to get each piece of data separately.
- We can also use **std::getline** to get each piece of data from the line itself.

Parsing CSV Data

- Add `#include <sstream>`
- Use std::stringstream and std::getline.

```
std::stringstream strStream(csvString);
std::string data;
while (std::getline(strStream, data, '|'))
{
    std::cout << data << "\n";
}
```