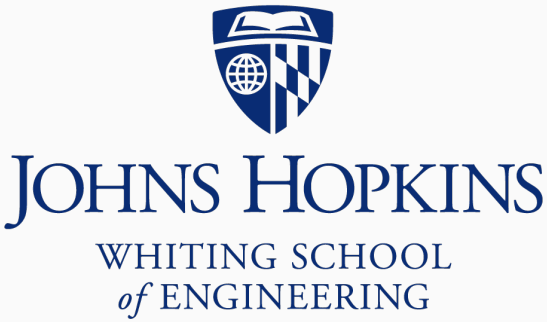
fstream

Ben Langmead

ben.langmead@gmail.com

www.langmead-lab.org



Source markdown available at github.com/BenLangmead/c-cpp-notes

fstream

In C, `printf` wrote `stdout` and `scanf` read from `stdin`

`fprintf` and `fscanf` were their counterparts for working with named files

In C++, we have `std::cout` and `std::cin`

`std::ofstream` and `std::ifstream` are their counterparts for working with named files

- `#include <fstream>`

ofstream / ifstream

```cpp
#include <iostream>
#include <fstream>

using std::ofstream;
using std::endl;

int main() {
    // open for writing, create if needed, possibly overwrite
    // just like fopen("hello.txt", "w")
    ofstream ofile("hello.txt");
    if(ofile) {
        ofile << "Hello, World!" << endl;
    } else {
        std::cout << "Error opening file" << endl;
        return 1;
    }
    return 0;
}
```

ofstream / ifstream

```
$ g++ -o ofs1 ofs1.cpp -std=c++11 -pedantic -Wall -Wextra
$ ./ofs1
$ cat hello.txt
Hello, World!
```

ofstream / ifstream

```cpp
#include <iostream>
#include <fstream>
#include <string>

using std::cout;
using std::endl;
using std::string;
using std::ifstream;

int main() {
    ifstream ifile("hello.txt"); // assuming we've written it
    if(ifile) {
        string word;
        while(ifile >> word) { cout << word << ' '; }
        cout << endl;
    } else {
        std::cout << "Error opening file" << endl;
        return 1;
    }
    return 0;
}
```

ofstream / ifstream

```
$ g++ -o ifs1 ifs1.cpp -std=c++11 -pedantic -Wall -Wextra
$ ./ifs1
Hello, World!
```

ofstream / ifstream

Could have explicitly closed the file:

```
    if(ifile) {
        string word;
        while(ifile >> word) { cout << word << ' '; }
        cout << endl;
        ifile.close(); // *** explicitly close ***
    }
}
```

Not necessary; if you don't close explicitly *the* `fstream` *destructor closes it for you*

ofstream / ifstream

Unlike *streams* (`std::ostream`, `std::istream`), you can move *forward and backward* through a file

```
#include <iostream>
#include <fstream>

using std::endl;      using std::cout;
using std::ifstream;  using std::ofstream;

int main() {
    char *buffer = NULL;
    int length = 0;
    {
        ifstream is("hello.txt");
        if(!is) {
            cout << "Error opening file" << endl;
            return 1;
        }
        is.seekg(0, is.end);  // move to *end* of file
        length = is.tellg();  // get my offset into file
        is.seekg(0, is.beg);  // move to *beginning* of file

        buffer = new char[length];
        is.read(buffer, length);     // read file contents
    } // is goes out of scope and is closed
```

ofstream / ifstream

```
// main() continued

    {
        ofstream os("hello2.txt");
        if(!os) {
            cout << "Error opening file" << endl;
            delete[] buffer;
            return 1;
        }
        os.write(buffer, length);
        delete[] buffer;
    } // os goes out of scope and is closed

    return 0;
}
```

ofstream / ifstream

```
$ g++ -o seek seek.cpp -std=c++11 -pedantic -Wall -Wextra
$ cat hello.txt
Hello, World!
$ ./seek
$ cat hello2.txt
Hello, World!
```

ofstream / ifstream

To summarize:

```
std::ifstream is("filename");  // open file for reading
if(is) {                       // check that open was successful
    is >> xyz;                 // read from file
    is.seekg(0, is.end);       // move to *end* of file
    length = is.tellg();       // get my offset into file
    is.seekg(0, is.beg);       // move to *beginning* of file
    is.read(buffer, length);   // read from file (binary, like fread)
    is.close();                // close (alternately, let destructor do it)
}

std::ofstream os("filename2"); // open file for writing
if(os) {                       // check that open was successful
    os << xyz;                 // write to file (text)
    os.write(buffer, length);  // write to file (binary, like fwrite)
}                              // destructor closes file
```

ofstream / ifstream