THE UNIVERSITY OF VERMONT
COLLEGE OF ENGINEERING &
MATHEMATICAL SCIENCES

Reading Input From a File

in C++

CS 124 / Department of Computer Science

Three Examples

- Reading entire lines from a text file,
- reading fields from a CSV file
- reading fields from CSV file that contains a quoted field.

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

The Basics

Open a file, read its lines, close the file

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../wasteland.txt");  // your filename here
    std::string line;
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            std::getline(ifs, line);  // get line
            std::cout << line << std::endl;   // print the result
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

Comma-delimited data

A simple example

```
California,39512223,CA
Texas,28995881,TX
Florida,21477737,FL
New York,19453561,NY
Pennsylvania,12801989,PA
Illinois,12671821,IL
Ohio,11689100,OH
Georgia,10617423,GA
North Carolina,10488084,NC
Michigan,9986857,MI
New Jersey,8882190,NJ
Virginia,8535519,VA
Washington,7614893,WA
...
```

Comma-delimited data

A simple example

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../states.csv");  // your filename here
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            ... // more stuff here
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

A simple example

```
char comma = ',';
std::string sname;
int population;
std::string abbr;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, sname, comma);  // read name; comma is delimiter
    ifs >> population;  // read next field into pop
    ifs >> comma;  // get that comma
    std::getline(ifs, abbr);  // finish line; newline is delimiter

    std::cout << sname << " | " << population << " | "
              << abbr << std::endl;
}
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
Albert Ayler,Spiritual Unity,1965
Alexander von Schlippenbach,Pakistani Pomade,1972
Alexander Hawkins,"All There, Ever Out",2012
Alfred Schnittke,Cello Sonatas,1998
Allison Miller,Boom Tic Boom,2010
Amalgam,Prayer for Peace,1969
AMM,"Live In Allentown, USA",1996
Andrew Hill,Black Fire,1964
Angelica Sanchez,A Little House,2011
Animal Collective,Sung Tongs,2004
Anna Webber,Percussive Mechanics,2013
Anneli Drecker,Revelation for Personal Use,2017
Anthony Braxton,For Alto,1969
Anton Webern,"Complete Works (Juilliard, LSO, Boulez, et al.)",1978
Aretha Franklin,Aretha's Gold,1969
Arrington de Dionyso,"Trance Punk Manifesto, Vol IV",2012
Art Blakey,"At the Cafe Bohemia, Vol 1",1955
...
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
Albert Ayler,Spiritual Unity,1965
Alexander von Schlippenbach,Pakistani Pomade,1972
Alexander Hawkins,"All There, Ever Out",2012
Alfred Schnittke,Cello Sonatas,1998
Allison Miller,Boom Tic Boom,2010
Amalgam,Prayer for Peace,1969
AMM,"Live In Allentown, USA",1996
Andrew Hill,Black Fire,1964
Angelica Sanchez,A Little House,2011
Animal Collective,Sung Tongs,2004
Anna Webber,Percussive Mechanics,2013
Anneli Drecker,Revelation for Personal Use,2017
Anthony Braxton,For Alto,1969
Anton Webern,"Complete Works (Juilliard, LSO, Boulez, et al.)",1978
Aretha Franklin,Aretha's Gold,1969
Arrington de Dionyso,"Trance Punk Manifesto, Vol IV",2012
Art Blakey,"At the Cafe Bohemia, Vol 1",1955
...
```

Comma-delimited data

A simple example

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

int main() {
    std::ifstream ifs;  // input file stream
    ifs.open ("../albums.csv");  // your filename here
    if (ifs) {
        while(ifs && ifs.peek() != EOF){  // while all is good...
            ... // more stuff here
        }
    } else {
        std::cout << "Unable to read from file." << std::endl;
    }
    ifs.close();
    return 0;
}
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Comma-delimited data

Handling quoted strings (which contain commas)

```cpp
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Comma-delimited data

Handling quoted strings (which contain commas)

```
char comma = ',', doubleQuote = '"';
std::string artist, title, yearStr;
int year;
while(ifs && ifs.peek() != EOF) {
    std::getline(ifs, artist, comma); // get artist (up to first comma)
    if (ifs.peek() == doubleQuote) {  // is the next char " ?
        ifs >> doubleQuote;  // if so, consume it
        std::getline(ifs, title, doubleQuote);  // consume to the next "
        ifs >> comma;  // consume the following comma
    } else {
        std::getline(ifs, title, comma);  // otherwise consume to next ,
    }
    std::getline(ifs, yearStr);  // consume to newline
    year = std::stoi(yearStr);  // optionally convert string to int
```

Summary

- Use `fstream` to read from file
- Create an `ifstream` object, open your file, and be sure to close file when done
- Check to see if your `ifstream` object evaluates to true — did you open file OK?
- Use a `while` loop to iterate through your file
- Use `std::getline` and set delimiter if appropriate
- `std::getline` without delimiter set will read to end of line
- Use extract operator (`<<`) to extract chunks as needed
- Annotated source code posted on Blackboard