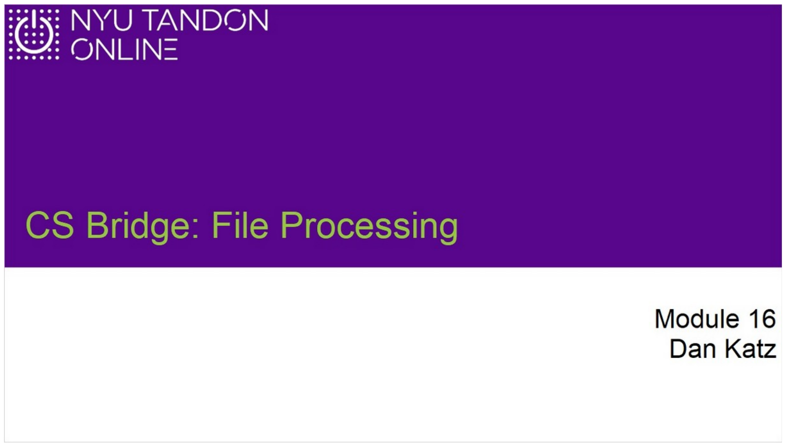
# CS Bridge Module 16 File Processing

## 1. File Processing

### *1.1 CS Bridge: File Processing*



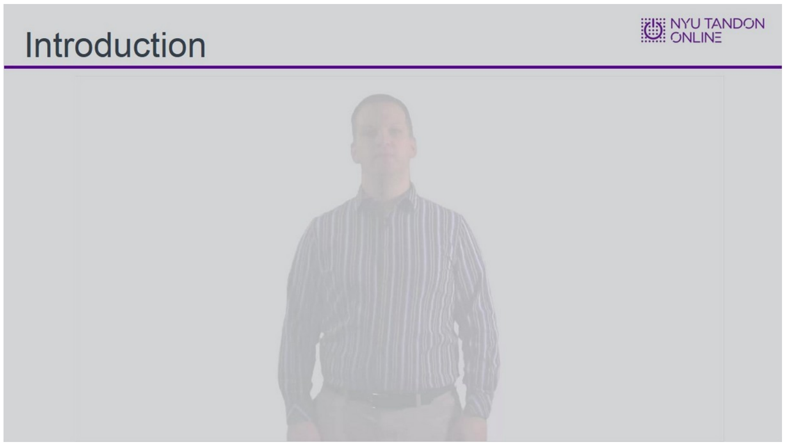
NYU TANDON ONLINE

CS Bridge: File Processing

Module 16
Dan Katz

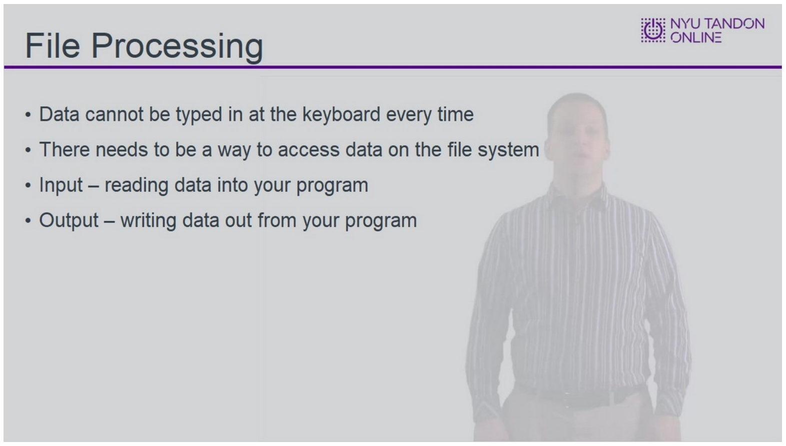
**Notes:**

### *1.2 Introduction*



Introduction

## *1.3 File Processing*

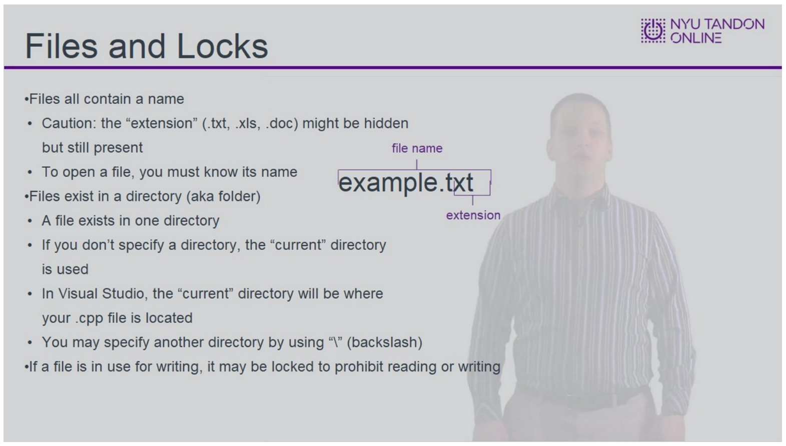


NYU TANDON ONLINE

### File Processing

- Data cannot be typed in at the keyboard every time
- There needs to be a way to access data on the file system
- Input – reading data into your program
- Output – writing data out from your program

## *1.4 Files and locks*

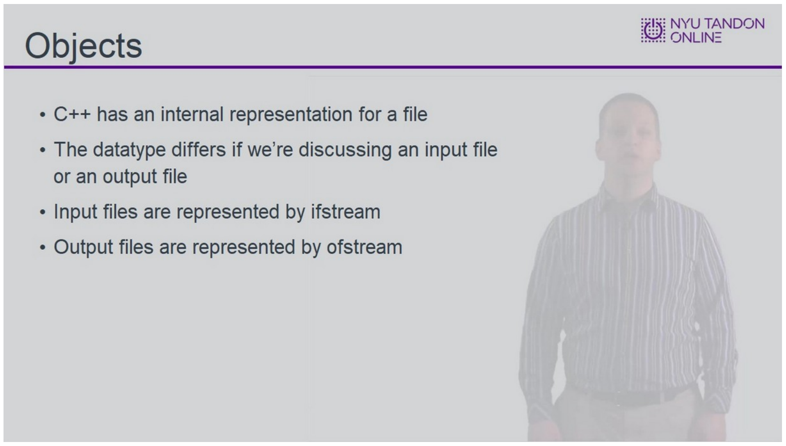


NYU TANDON ONLINE

### Files and Locks

- Files all contain a name
  - Caution: the "extension" (.txt, .xls, .doc) might be hidden but still present
  - To open a file, you must know its name
- Files exist in a directory (aka folder)
  - A file exists in one directory
  - If you don't specify a directory, the "current" directory is used
  - In Visual Studio, the "current" directory will be where your .cpp file is located
  - You may specify another directory by using "\" (backslash)
- If a file is in use for writing, it may be locked to prohibit reading or writing

## *1.5 Objects*

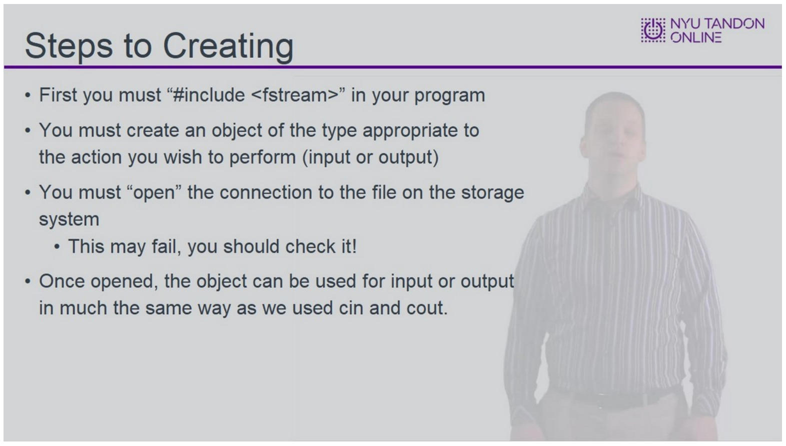


NYU TANDON ONLINE

### Objects

- C++ has an internal representation for a file
- The datatype differs if we're discussing an input file or an output file
- Input files are represented by ifstream
- Output files are represented by ofstream

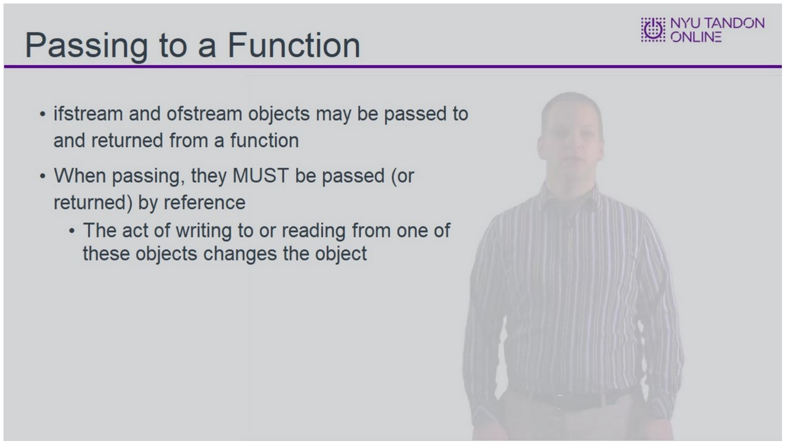
## *1.6 Steps to creating*



### Steps to Creating

- First you must "#include <fstream>" in your program
- You must create an object of the type appropriate to the action you wish to perform (input or output)
- You must "open" the connection to the file on the storage system
  - This may fail, you should check it!
- Once opened, the object can be used for input or output in much the same way as we used cin and cout.

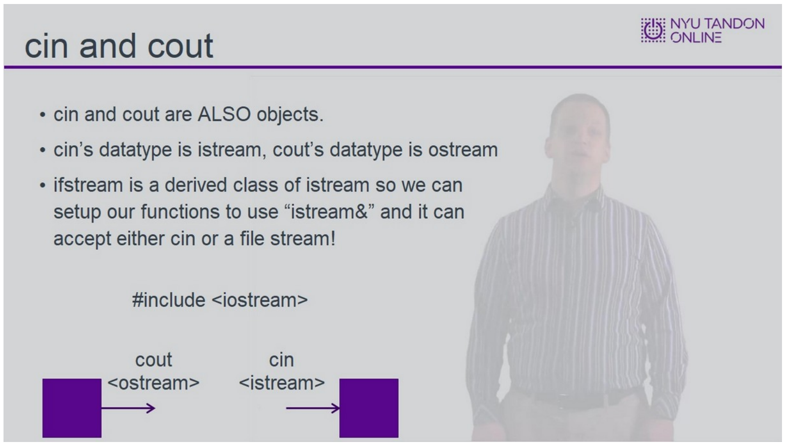
## *1.7 Passing to a function*



### Passing to a Function

- ifstream and ofstream objects may be passed to and returned from a function
- When passing, they MUST be passed (or returned) by reference
  - The act of writing to or reading from one of these objects changes the object

## *1.8 cin and cout*



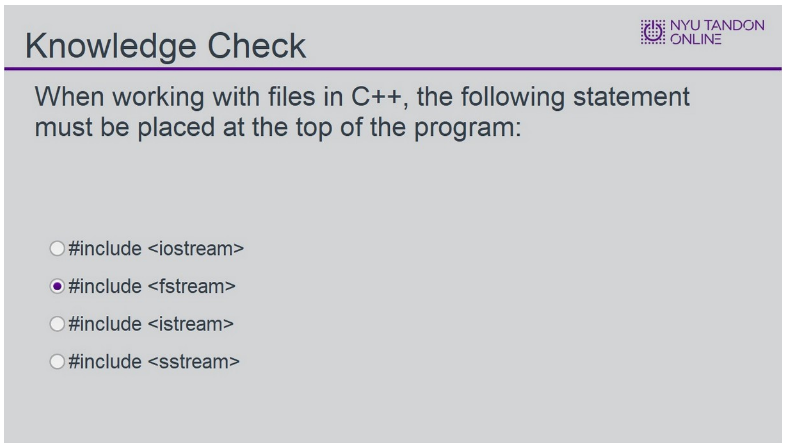
### cin and cout

- cin and cout are ALSO objects.
- cin's datatype is istream, cout's datatype is ostream
- ifstream is a derived class of istream so we can setup our functions to use "istream&" and it can accept either cin or a file stream!

#include <iostream>
cout <ostream>
cin <istream>

## *1.9 Knowledge Check*

*(Multiple Choice, 10 points, 4 attempts permitted)*



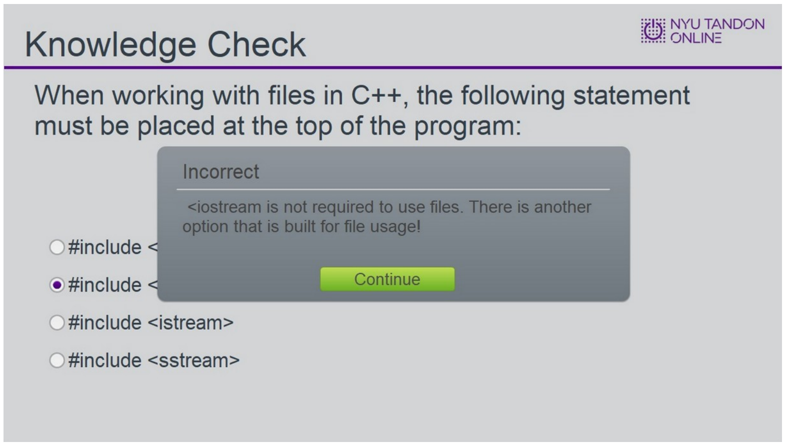
Knowledge Check

When working with files in C++, the following statement must be placed at the top of the program:

- #include <iostream>
- [✓ #include <fstream>]
- #include <istream>
- #include <sstream>

| Correct | Choice | Feedback |
|---|---|---|
| | #include <iostream> | <iostream is not required to use files. There is another option that is built for file usage! |
| X | #include <fstream> | Correct! |
| | #include <istream> | istream is part of the iostream object. Therefore, <istream> is not a standalone item |
| | #include <sstream> | <sstream> is meant for usage with strings. For files, there is another object with a similar name |

**Incorrect (Slide Layer)**



Knowledge Check

NYU TANDON ONLINE

When working with files in C++, the following statement must be placed at the top of the program:

Incorrect

<iostream is not required to use files. There is another option that is built for file usage!

Continue

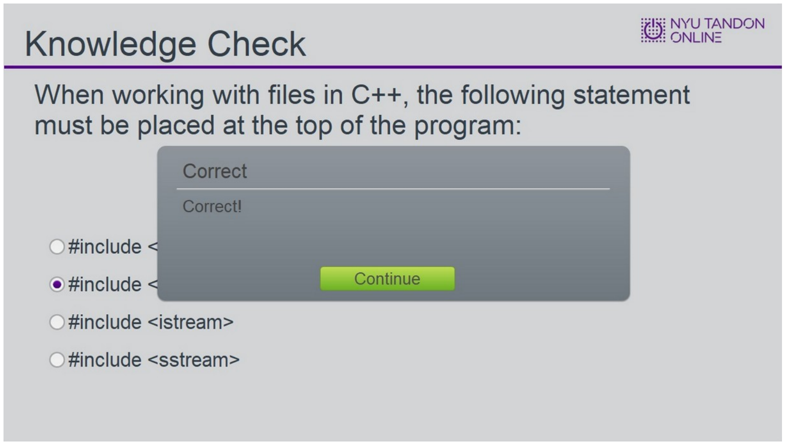
#include <

[✓ #include <]

#include <istream>

#include <sstream>

**Correct (Slide Layer)**



Knowledge Check

NYU TANDON ONLINE

When working with files in C++, the following statement must be placed at the top of the program:

Correct

Correct!

Continue

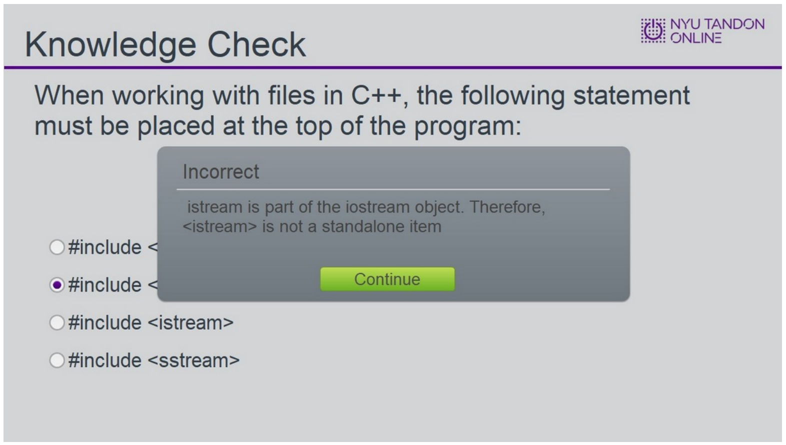
#include <

[✓ #include <]

#include <istream>

#include <sstream>

**Incorrect (Slide Layer)**



Knowledge Check

NYU TANDON ONLINE

When working with files in C++, the following statement must be placed at the top of the program:

Incorrect

istream is part of the iostream object. Therefore, <istream> is not a standalone item

Continue

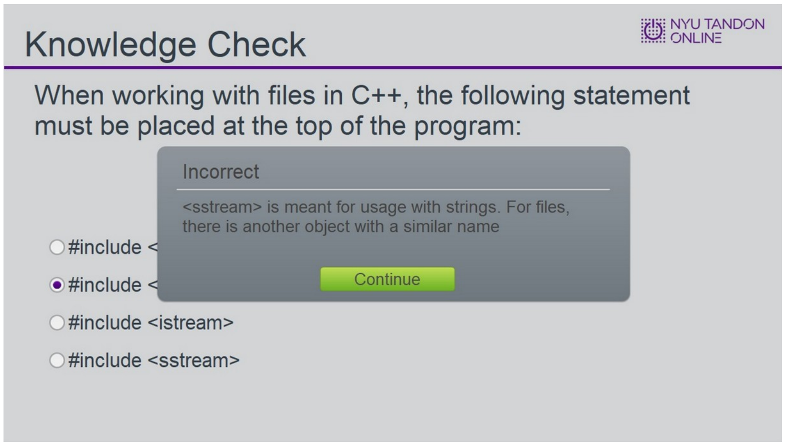
#include <

[✓ #include <]

#include <istream>

#include <sstream>

**Incorrect (Slide Layer)**



## Knowledge Check

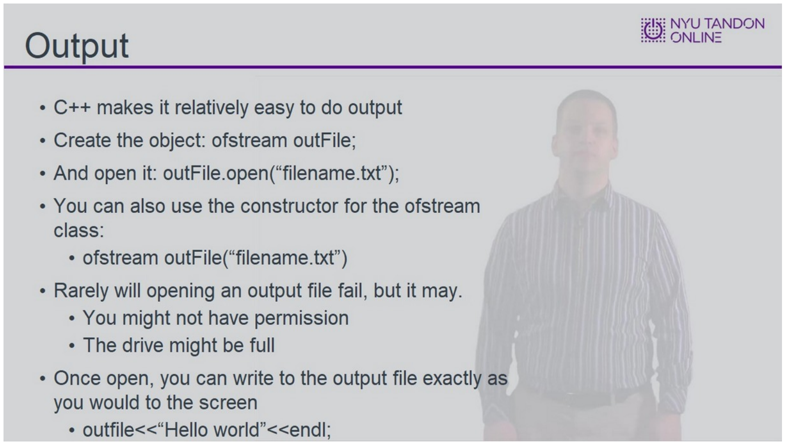
When working with files in C++, the following statement must be placed at the top of the program:

Incorrect

<sstream> is meant for usage with strings. For files, there is another object with a similar name

- ○ #include <
- ● #include <
- ○ #include <istream>
- ○ #include <sstream>

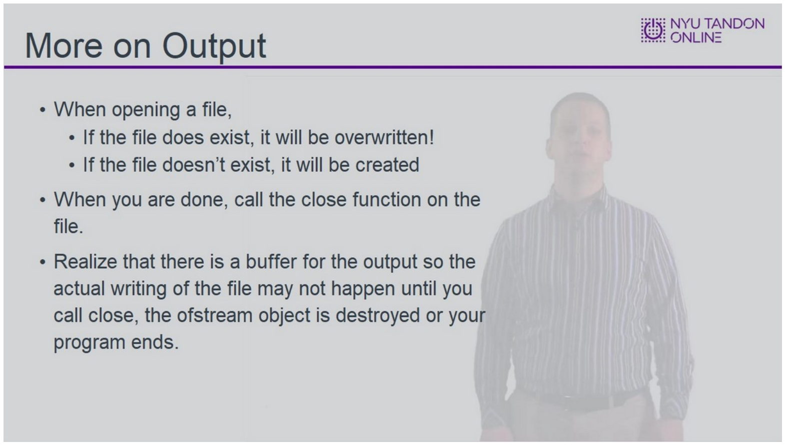
### *1.10 Output*



## Output

- C++ makes it relatively easy to do output
- Create the object: ofstream outFile;
- And open it: outFile.open(“filename.txt”);
- You can also use the constructor for the ofstream class:
  - ofstream outFile(“filename.txt”)
- Rarely will opening an output file fail, but it may.
  - You might not have permission
  - The drive might be full
- Once open, you can write to the output file exactly as you would to the screen
  - outfile<<“Hello world”<<endl;

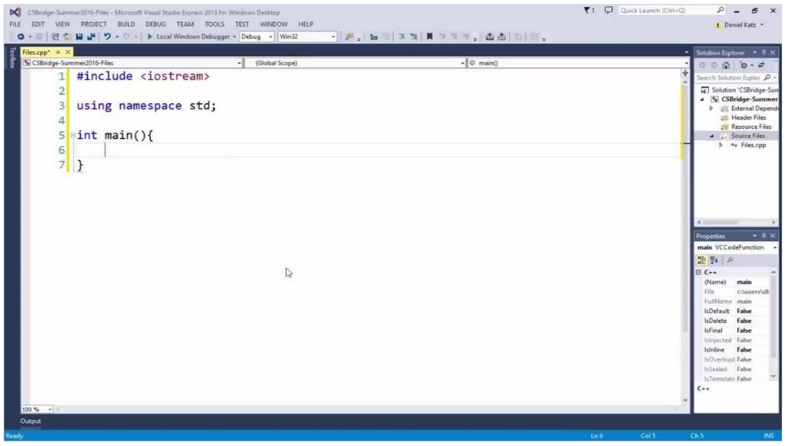
### *1.11 More on output*



## More on Output

- When opening a file,
  - If the file does exist, it will be overwritten!
  - If the file doesn’t exist, it will be created
- When you are done, call the close function on the file.
- Realize that there is a buffer for the output so the actual writing of the file may not happen until you call close, the ofstream object is destroyed or your program ends.

## 1.12 Example: Creating a File and Outputting Information


CSBridge-Summer2016-Files - Microsoft Visual Studio Express 2013 for Windows Desktop
Files.cpp*
Solution Explorer
Properties

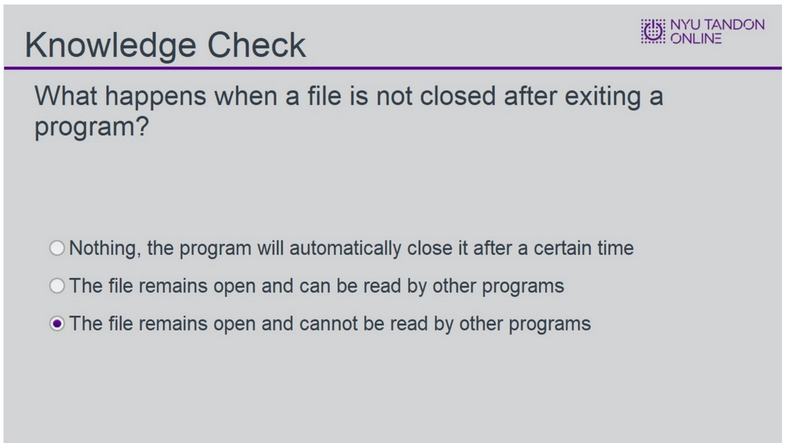
```
#include <iostream>

using namespace std;

int main(){
    
}
```

## 1.13 Knowledge Checks

*(Multiple Choice, 10 points, 1 attempt permitted)*



Knowledge Check

What happens when a file is not closed after exiting a program?

- Nothing, the program will automatically close it after a certain time
- The file remains open and can be read by other programs
- [✓ The file remains open and cannot be read by other programs]

| Correct | Choice |
|---|---|
| | Nothing, the program will automatically close it after a certain time |
| | The file remains open and can be read by other programs |
| X | The file remains open and cannot be read by other programs |

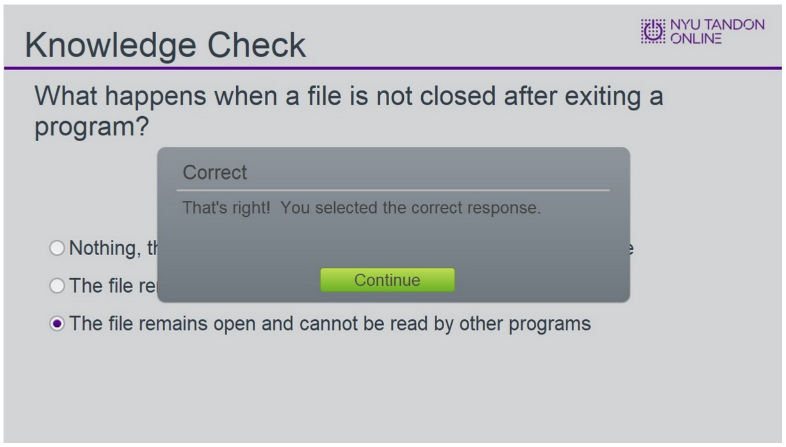
**Feedback when correct:**

That's right!  You selected the correct response.

**Feedback when incorrect:**

You did not select the correct response.

## Correct (Slide Layer)



### Knowledge Check

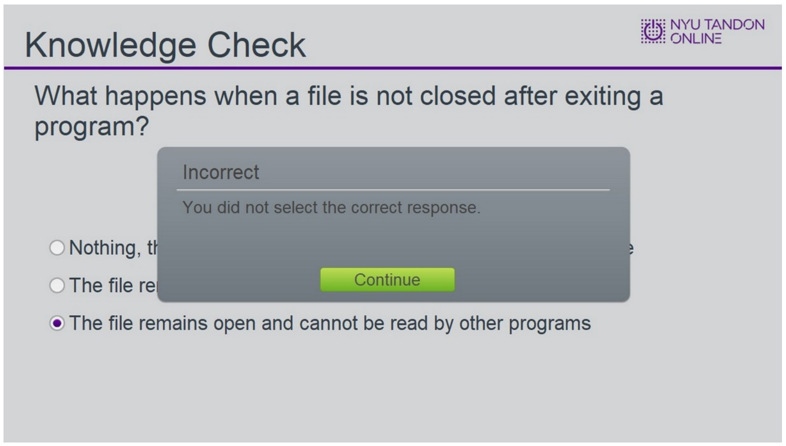
What happens when a file is not closed after exiting a program?

Correct

That's right!  You selected the correct response.

- ○ Nothing, th
- ○ The file re
- ● The file remains open and cannot be read by other programs

## Incorrect (Slide Layer)



### Knowledge Check

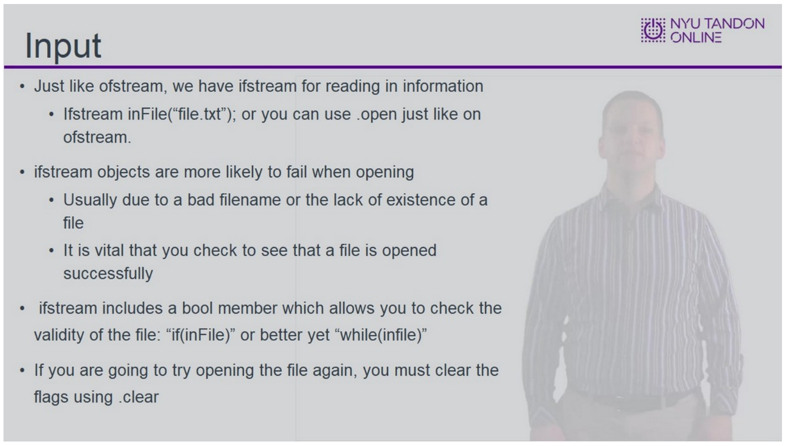
What happens when a file is not closed after exiting a program?

Incorrect

You did not select the correct response.

- ○ Nothing, th
- ○ The file re
- ● The file remains open and cannot be read by other programs

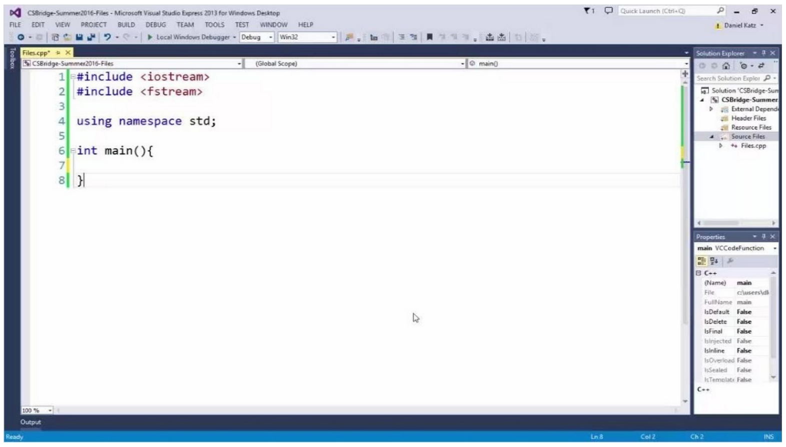
### *1.14 Input*



## Input

- Just like ofstream, we have ifstream for reading in information
  - Ifstream inFile("file.txt"); or you can use .open just like on ofstream.
- ifstream objects are more likely to fail when opening
  - Usually due to a bad filename or the lack of existence of a file
  - It is vital that you check to see that a file is opened successfully
- ifstream includes a bool member which allows you to check the validity of the file: "if(inFile)" or better yet "while(infile)"
- If you are going to try opening the file again, you must clear the flags using .clear

### *1.15 Example: Opening a File Stream Object*



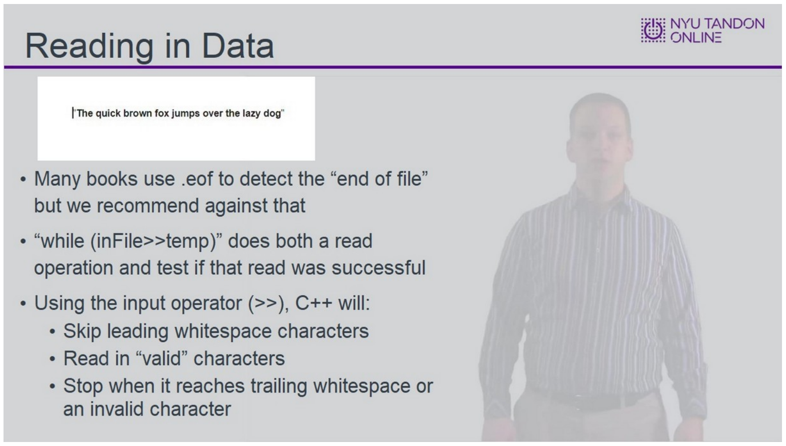
```
#include <iostream>
#include <fstream>

using namespace std;

int main(){

}
```

### *1.16 Reading in data*

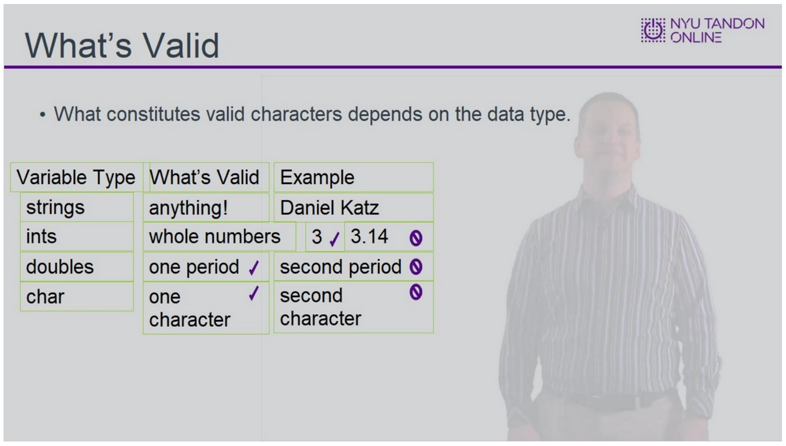


## Reading in Data

"The quick brown fox jumps over the lazy dog"

- Many books use .eof to detect the "end of file" but we recommend against that
- "while (inFile>>temp)" does both a read operation and test if that read was successful
- Using the input operator (>>), C++ will:
  - Skip leading whitespace characters
  - Read in "valid" characters
  - Stop when it reaches trailing whitespace or an invalid character

## 1.17 *What's valid*



### What's Valid

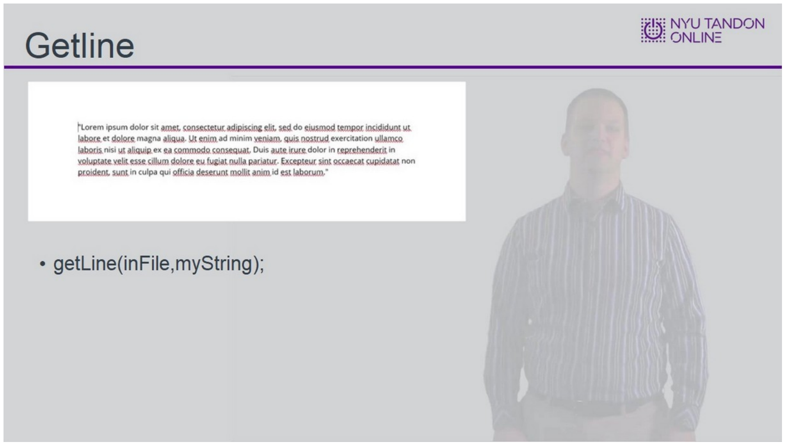
NYU TANDON ONLINE

- What constitutes valid characters depends on the data type.

| Variable Type | What's Valid | Example |
|---|---|---|
| strings | anything! | Daniel Katz |
| ints | whole numbers | 3 ✓ 3.14 ⊘ |
| doubles | one period ✓ | second period ⊘ |
| char | one character ✓ | second character ⊘ |

**Notes:**

## 1.18 *Getline*



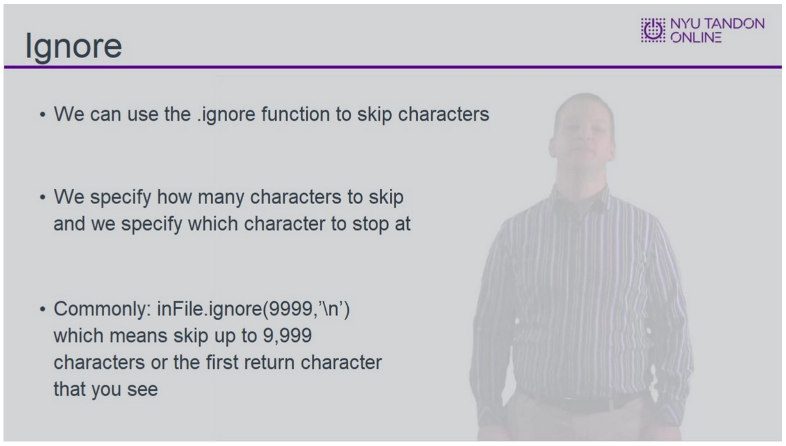
### Getline

NYU TANDON ONLINE

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

- getLine(inFile,myString);

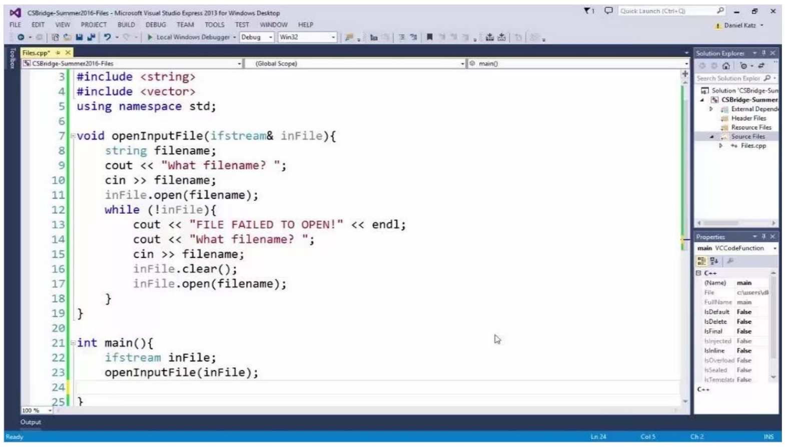
### *1.19 ignore*



Ignore

- We can use the .ignore function to skip characters
- We specify how many characters to skip and we specify which character to stop at
- Commonly: inFile.ignore(9999,'\n') which means skip up to 9,999 characters or the first return character that you see

### *1.20 Example: Opening a File Stream Object*



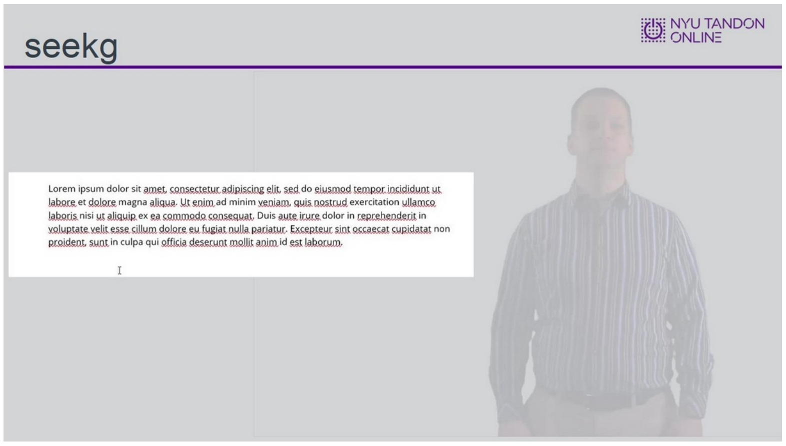
```
#include <string>
#include <vector>
using namespace std;

void openInputFile(ifstream& inFile){
    string filename;
    cout << "What filename? ";
    cin >> filename;
    inFile.open(filename);
    while (!inFile){
        cout << "FILE FAILED TO OPEN!" << endl;
        cout << "What filename? ";
        cin >> filename;
        inFile.clear();
        inFile.open(filename);
    }
}

int main(){
    ifstream inFile;
    openInputFile(inFile);

}
```

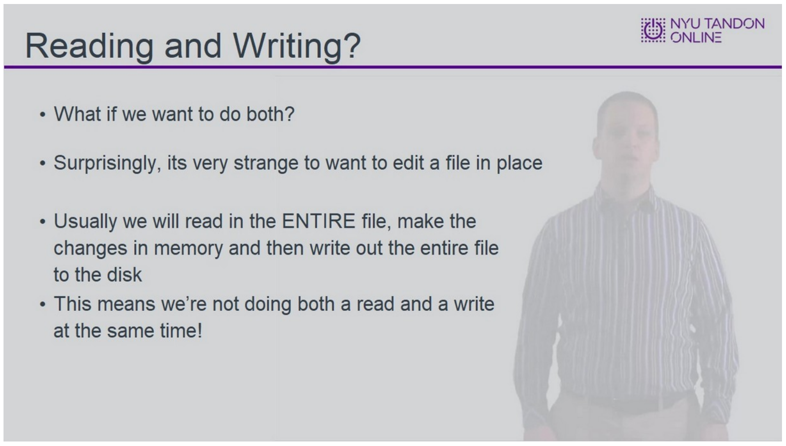
### *1.21 seekg*



seekg

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

### *1.22 Reading and writing?*



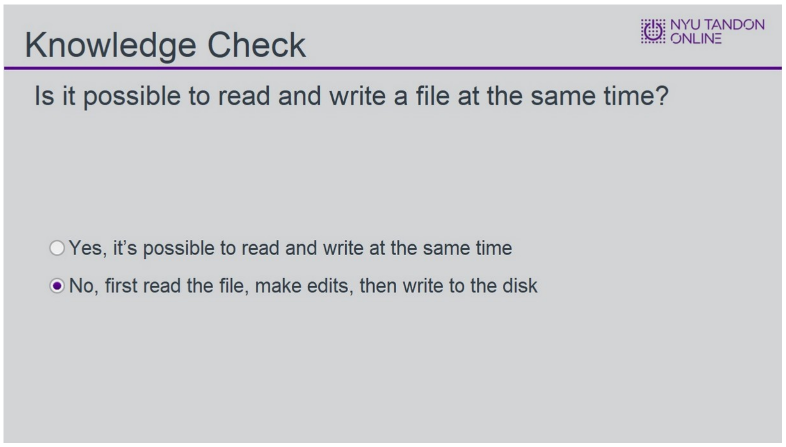
## Reading and Writing?

NYU TANDON ONLINE

- What if we want to do both?
- Surprisingly, its very strange to want to edit a file in place
- Usually we will read in the ENTIRE file, make the changes in memory and then write out the entire file to the disk
- This means we're not doing both a read and a write at the same time!

### *1.23 Knowledge Check*

*(Multiple Choice, 10 points, 1 attempt permitted)*



## Knowledge Check

NYU TANDON ONLINE

Is it possible to read and write a file at the same time?

○ Yes, it's possible to read and write at the same time

● No, first read the file, make edits, then write to the disk

| Correct | Choice |
|---|---|
| | Yes, it's possible to read and write at the same time |
| X | No, first read the file, make edits, then write to the disk |

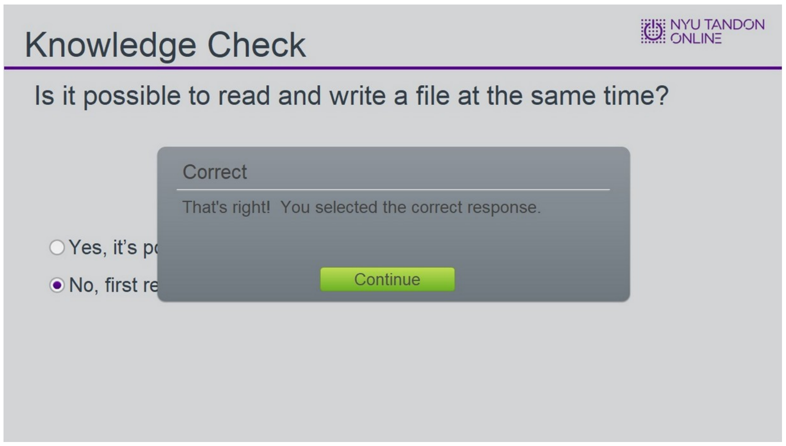
**Feedback when correct:**

That's right!  You selected the correct response.

**Feedback when incorrect:**

You did not select the correct response.

## Correct (Slide Layer)



### Knowledge Check

Is it possible to read and write a file at the same time?

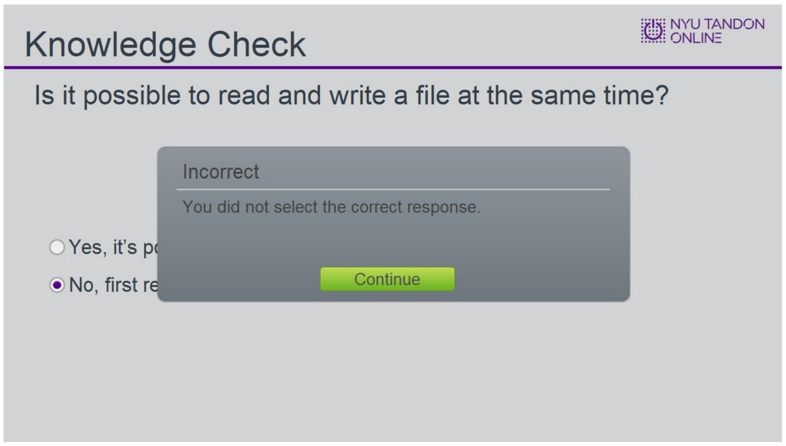
Correct

That's right!  You selected the correct response.

○ Yes, it's p

● No, first re

## Incorrect (Slide Layer)



### Knowledge Check

Is it possible to read and write a file at the same time?

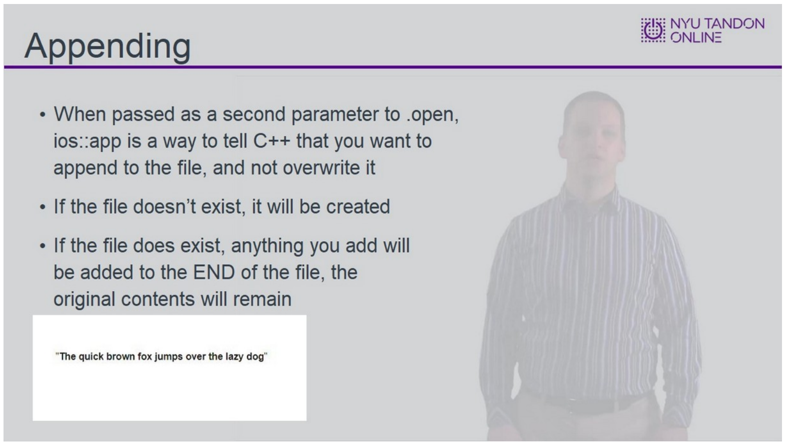
Incorrect

You did not select the correct response.

○ Yes, it's p

● No, first re

## *1.24 Appending*

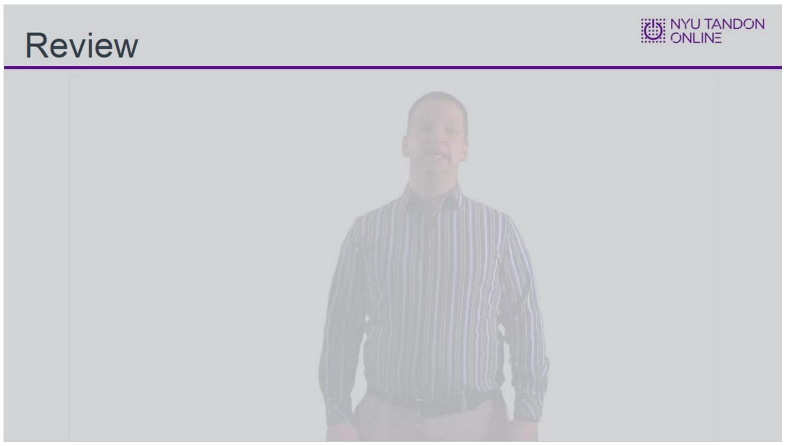


Appending

- When passed as a second parameter to .open, ios::app is a way to tell C++ that you want to append to the file, and not overwrite it
- If the file doesn't exist, it will be created
- If the file does exist, anything you add will be added to the END of the file, the original contents will remain

"The quick brown fox jumps over the lazy dog"

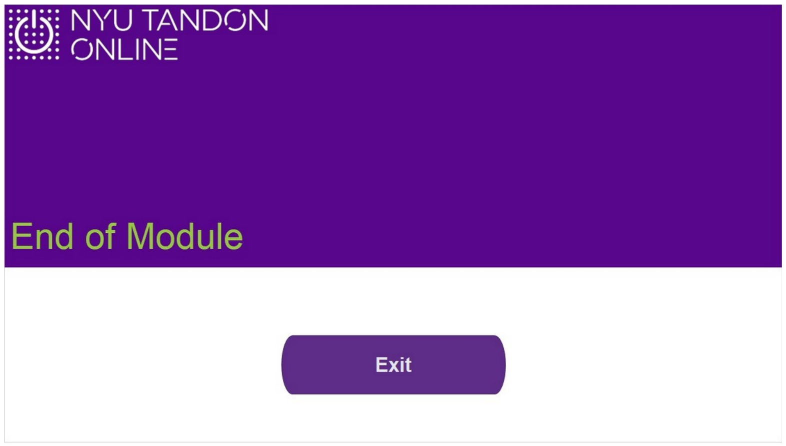
## *1.25 Review*



Review

**Notes:**

### *1.26 End of Module*



End of Module