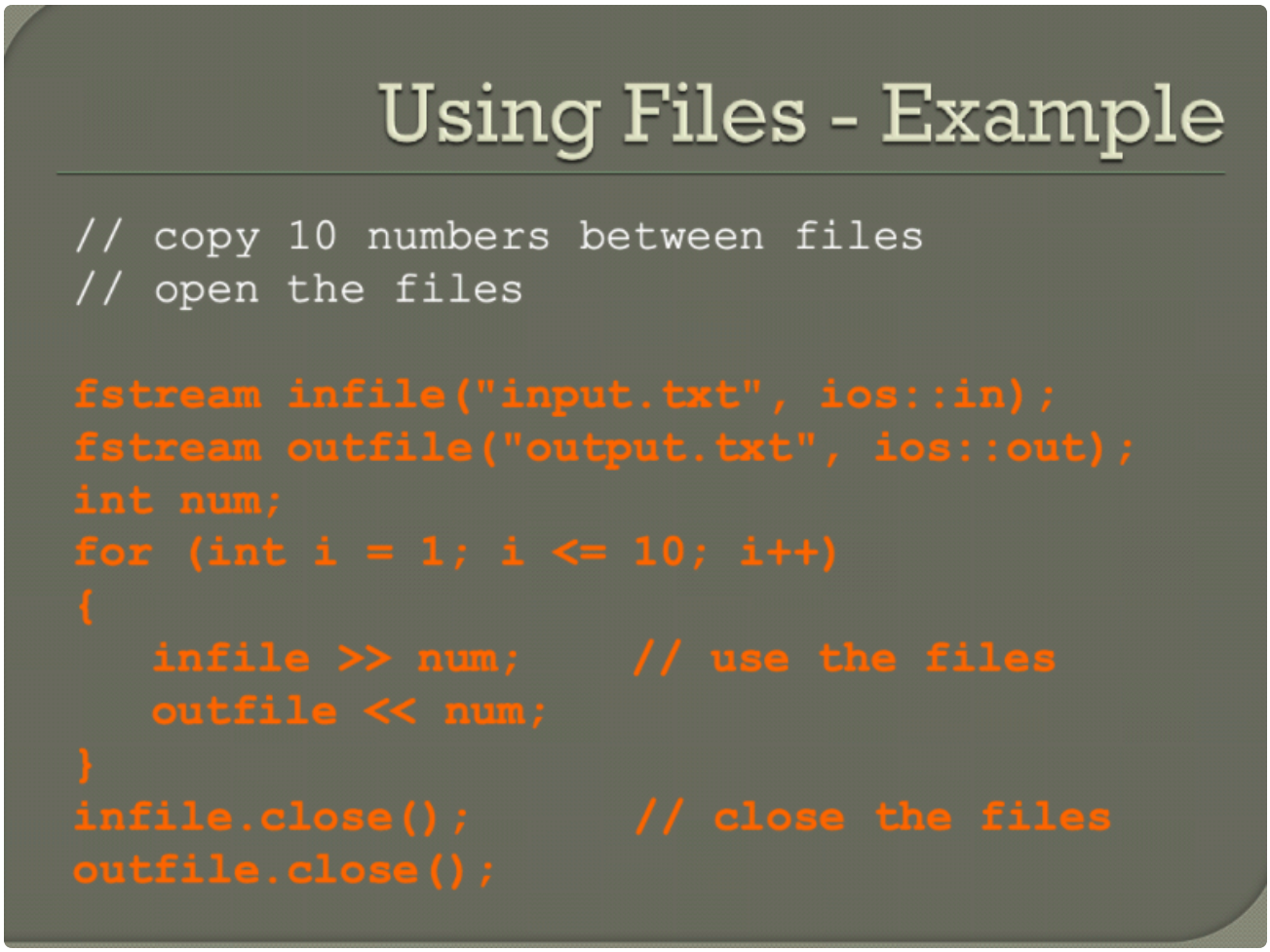
Overview

- File Operations
- File Output Formatting
- Passing File Stream Objects to Functions
- Error Testing
- Member Functions For Reading and Writing Files
- Working With Multiple Files
- Binary Files
- Opening a File for Input and Output

File Operations

- **File**: a set of data stored on a computer, often on a disk drive
- Programs can read/write files
  - word processing, databases, spreadsheets, compliers
- Systems require `fstream` header files
  - `ifstream` data type for input files
  - `ofstream` data type for output files
  - `fstream` from both
    - `ios::in` is used to take in inputs
    - `ios::out` is used to write
    - `dFile.open("file.txt", ios::in, ios::out)` can be used for all
  - All can use `>>` or `<<` to read and write to a file
  - Can use `eof` member function to test for end of input file
- Many access flags exists:
  - `ios::app` can be used to *append* to a file, if the file contents already exists.
    - Adds to the end by default
    - If file does not exist, it will create
  - `ios::ate` can be used to go to the end of an existing file. Output can be written anywhere in file
  - `ios::binary` causes data to be written or read in pure binary format. (default is text)
  - `ios::in` is input mode. Data will be read from file. If file DNE then it fails
  - `ios::out` data will be written to file. If file DNE then it will be created
  - `ios::trunc` essentially wipes all the content of the file



- `infile` in this context is essentially the variable name

- `ifstream` opens file for input only, file can not be written, and `open` fils if the file DNE
- `ofstream` open for output only
  - File cannot be read from, file is created if DNE, file contents is erased if file exists.

Error Testing

- You can check if file stream has failed by doing `if (!gradelist)...` or `if (gradeList.fail())...`
- `ios::eofbit` set when the end of file
- `ios::failbit` set when operations fail
- `ios::hardfail` set when error occurred and no recovery
- `ios::badbit` set when invalid operation is attempted
- `ios::goodbit` set when no other bits are set
- `eof()` returns true if eofbit set
- `fail()` returns true if `failbit` or `hardbit`
- `bad()` returns true if `badbit`
- `good()` returns true if `goodbit`
- `clear()` clears all flags

```
68  void showState(fstream &file)
69  {
70     cout << "File Status:\n";
71     cout << "  eof bit: " << file.eof() << endl;
72     cout << "  fail bit: " << file.fail() << endl;
73     cout << "  bad bit: " << file.bad() << endl;
74     cout << "  good bit: " << file.good() << endl;
75     file.clear();  // Clear any bad bits
76  }
```

Reading Member Functions for Reading and Writing Files

- `geline()` reads inputs including whitespace
  - Has 3 arguments: `Char[]` to hold characters, number of characters to read, stop if certain char is found.
    - By default `'\n' is default for the third operation`
- `get` reads a single character
- `put` writes a single character