Reading from and Writing to Files

Section 3.12 & 13.1 & 13.5

Files (3.12)

- Data stored in variables is temporary
- Files are used to permanently store large amounts of data
- We will learn how to write programs that can
  - Create files
  - Write to files
  - Read from files
- This is similar to how we read from the keyboard and wrote to the screen

Steps to Using Files

- There are five steps that must be taken in order to use files in C++
  1. Include header files
  2. Define a file stream object
  3. Open the file
  4. Use the file
  5. Close the file

1. Libraries

- To access files you will need to include
  - <iostream>
  - <fstream>

2. File Stream Objects

```
ifstream inputFile;
ofstream outputFile;
fstream inAndOut;
```

- File stream objects are the ways that you refer to the files you are using
  - Can specify which input/output file to use
  - May input from more than one file
  - May output to more than one file

3. Opening Files

```
inputFile.open("filename")
```

- Same syntax for both input and output files
- Filename is a string literal
- Example:

```
ifstream inputFile;
inputFile.open("input.dat");
```

Check File Opened Correctly

- Before we start using the file for reading or writing, we should make sure that it opened correctly

```
if(!inputInfo == true)
{
   cout << "Error opening input file ";
   exit(1);
}
```


== true

- These two statements are equivalent
  - `if(!inputInfo == true)`
  - `if(!inputInfo)`
- Even if you don't have == true in your loop, C++ will put it there by default
- This applies to all conditional statements in repetition and selection structures

4. Using File Streams

- Use input file variable wherever you use cin
- Examples:
  - `inputFile >> num;`
- Output output file variable wherever you use cout
- Examples:
  - `outputFile << num;`

Example: Writing to a File

- The following program asks the user to input numbers and writes these numbers to a file

Reading from a File

- Write a program that will read in a sequence of numbers (double) from a file and calculate the sum. Assume that the last number is the trailer (-9999)

Reading Until the EOF (p 811)

- It is possible to read from a file until the end is reached

```
while (inputFile >> num)
{
    cout << num << " ";
    sum += num;
}
```

Reading Characters

- Write a program that reads in some text from a file and outputs that text to the screen
- The file contains:

```
Hello Everyone!
I'm a file that
contains some text.
```


The Output (p 828)

- `HelloEveryone!I'mafilethatcontainssometext.`

- What's happened?!
- All spaces, tabs, and new lines have been ignored.
- This is because >> only reads visible characters
- How can we read all characters so that the output looks exactly like the input

Problem

- Consider the data file below, where - indicate spaces:

```
--12--33.4
-d--12.3
-2--5
```

- What values would be assigned to the variables for each of the statements below where `inputFile` is the file variable?

```
int i,j;
double x,y;
char ch;
```

  - `inputFile >> i >> x >> y;`
  - `inputFile >> i >> j;`
  - `inputFile >> ch >> i;`
  - `inputFile >> x >> y >> ch >> x;`