
Reading from and Writing to Files

10/25/06

CS150 Introduction to Computer Science 1

1

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

10/25/06

CS150 Introduction to Computer Science 1

2

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

10/25/06

CS150 Introduction to Computer Science 1

3

1. Libraries

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

10/25/06

CS150 Introduction to Computer Science 1

4

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

10/25/06

CS150 Introduction to Computer Science 1

5

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");
```

10/25/06

CS150 Introduction to Computer Science 1

6

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);
}
```

10/25/06

CS150 Introduction to Computer Science 1

7

== 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

10/25/06

CS150 Introduction to Computer Science 1

8

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;`

10/25/06

CS150 Introduction to Computer Science 1

9

20.1 Example: Writing to a File

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

10/25/06

CS150 Introduction to Computer Science 1

10

Example

```
#include<fstream>
#include<iostream>
using namespace std;
int main()
{
    ofstream outputFile;
    int num;
    outputFile.open("out.dat");
    if (!outputFile)
    {
        cout << "*** Error opening file" << endl;
        exit (1);
    }
    cout << "Enter a number (9999 to quit): ";
    cin >> num;
    while (num != 9999)
    {
        outputFile << num << " ";
        cin >> num;
    }
    return 0;
}
```

10/25/06

CS150 Introduction to Computer Science 1

11

20.2 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)

10/25/06

CS150 Introduction to Computer Science 1

12

20.3 Reading Until the EOF

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

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

10/25/06

CS150 Introduction to Computer Science 1

13

20.4 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.
```

10/25/06

CS150 Introduction to Computer Science 1

14

Solution

```
ifstream inputFile;
char letter;

inputFile.open("in.dat");
if (!inputFile.is_open())
{
    cout << "*** Error opening file" << endl;
    exit (1);
}

while (inputFile >> letter)
{
    cout << letter;
}
cout << endl;
```

10/25/06

CS150 Introduction to Computer Science 1

15

The Output

- HelloEveryone!I'mafilethatcontainsstext.
- 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

10/25/06

CS150 Introduction to Computer Science 1

16

Solution

```
ifstream inputFile;
char letter;

inputFile.open("in.dat");
if (!inputFile.is_open())
{
    cout << "*** Error opening file" << endl;
    exit (1);
}

while (inputFile.get( letter ))
{
    cout << letter;
}
cout << endl;
```

10/25/06

CS150 Introduction to Computer Science 1

17

20.5 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;
```

10/25/06

CS150 Introduction to Computer Science 1

18

---

---

---

---

---

---

---

---