Text File IO

Class 17

Programs That Read from Files

- the open function places the read marker at the first byte of the file
- as data are extracted from the file, the read marker is advanced toward the end of the file
- cannot back up

Programs That Read from Files

- the open function places the read marker at the first byte of the file
- as data are extracted from the file, the read marker is advanced toward the end of the file
- cannot back up

- if data are numeric, use stream extraction

  `input file >> value;`
- if data are strings, use getline to read an entire line

  `getline (input file, a string);`
- see program 5-19, which uses getline because it's reading strings

End of File

- a file may contain a little data
- or a lot
- typically we don't know how many lines of data a file contains
- the stream extraction operator $\gg$ returns a value
- it returns true if the read was successful
- it returns false if the read was not successful
- this allows us to control a while loop with the extraction operator

- see program 5-22

File Open Errors

- there are various conditions that may cause an open function to fail
  - attempt to read a file that does not exist
  - attempt to write (create) a file without OS permissions in that directory
  - attempt to write (create) a file on a disk that's full

File Open Errors

- there are various conditions that may cause an open function to fail
  - attempt to read a file that does not exist
  - attempt to write (create) a file without OS permissions in that directory
  - attempt to write (create) a file on a disk that's full

- in general, always need to check whether the open function succeeded

- When the file open causes error, the fail() function returns true

```
input_file.open(filename);
if (input_file.fail()== false)
{
   ...
```

A More Realistic Example

- the examples shown so far have been relatively simplistic
- as a more realistic example, consider a file structured as shown below
- for use as input to a program to generate phone plan invoices

```
Rachel Carson
12.3
Aldo Leopold
7.25
Farley Mowat
1.23
```

Omitted

- we are using C++11
- therefore the top half of page 288 is irrelevant
- expunge those old C-strings whenever you can!

getline and Stream Extraction

- you are familiar with stream extraction $\gg$ that skips whitespace and stops at either
  - whitespace
  - a character not valid for the variable being read (e.g., a decimal point for an integer)

getline and Stream Extraction

- you are familiar with stream extraction that skips whitespace and stops at either
  - whitespace
  - a character not valid for the variable being read (e.g., a decimal point for an integer)

- you are familiar with getline that reads a string from the current position to the newline, skips the newline, and positions the read marker at the beginning of the next line

getline and Stream Extraction

- you are familiar with stream extraction that skips whitespace and stops at either
  - whitespace
  - a character not valid for the variable being read (e.g., a decimal point for an integer)

- you are familiar with getline that reads a string from the current position to the newline, skips the newline, and positions the read marker at the beginning of the next line

- there is no problem with a getline followed by a stream extraction, but
- stream extraction followed by a getline cause major problems

Data Files

- consider a data file that has alternating lines of data
- the first line consists of name
- the second line has numbers

B	o	b	\n	
4	7	2	9	\n
A	n	n	e	\n
1	2	3	\n	

Getline and Stream Extraction

- we use getline for the first name
- no problem, the read marker reads B-o-b
- skips *newline*
- points at 4 in the next line

B	o	b	\n	
4	7	2	9	\n
A	n	n	e	\n
1	2	3	\n	

Getline and Stream Extraction

- we now use stream extraction $\gg$ for the number
- no problem, we read 4-7-2-9 and the read marker ends on the *newline*

B	o	b	\n	
4	7	2	9	\n
A	n	n	e	\n
1	2	3	\n	

Getline and Stream Extraction

- now we want to use getline again to read Anne
- but the read marker is on the newline at the end of 4729
- getline reads until the next newline but it's already on a newline
- so it reads 0 characters, skips the newline, and puts the read marker on the A of Anne

B	o	b	\n	
4	7	2	9	\n
A	n	n	e	\n
1	2	3	\n	

Getline and Stream Extraction

- now everything is in disarray
- we're ready to read an integer
- but the read marker is on a character

B	o	b	\n	
4	7	2	9	\n
A	n	n	e	\n
1	2	3	\n	

Getline and Stream Extraction

- solution: after extraction $\gg$, we have to skip over the newline
- to read up to and skip over the newline, stream.ignore()

```
string name;
int number;

getline(file_stream, name);
file_stream >> number;
stream.ignore(); // to skip the newline
getline(file_stream, name);
file_stream >> number;
```