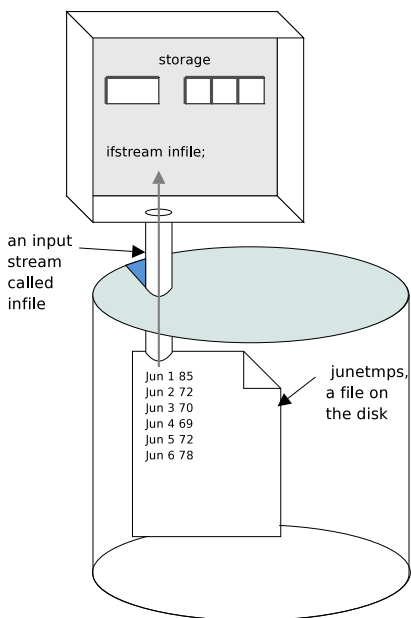### Getting Data from Files

Computers store data in files on disks. How can a program get data from a file or put data into a file?

The Big Idea

Working with files is easy. Here is an example. Say we have a file of daily high temperatures for the month of June. Say the name of the file is "junetmps". Reading data from that file involves a few basic steps:

STEP 0. Include the file stream stuff

```
#include <fstream>
```

STEP 1. Create an Input Stream Variable

```
ifstream infile;
```

STEP 2. Connect to File ("open the file")

```
string fname = "junetmps";   // Use -std=c++11 to use string
infile.open(fname);          // old style: char fname[] = ...
if (infile.fail()) {
    cerr << "Cannot open " << fname << endl;
    exit(1);
}
```

STEP 3. Read in and Process Lines of Data

```
string str;

while (getline(infile, str)) {    // read next line
    process(str);                 // do something
}
```

STEP 4. Close the Stream

```
infile.close();
```

storage

ifstream infile;

an input
stream
called
infile

Jun 1 85
Jun 2 72
Jun 3 70
Jun 4 69
Jun 5 72
Jun 6 78

junetmps,
a file on
the disk

Reading from Files into Other Data Types

What if you wanted to read data character by character? Or as a combination of strings, ints, and floats? How do I do *that*? Simple: replace the variable definition with the desired type, and replace the line of code marked "`// read next line`" with the corresponding input method:

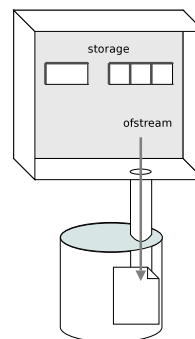|	LINES	CHAR-BY-CHAR	THINGS

vars	`string s;`	`char c;`	`string mon;`
			`int   day;`
			`float temp;`

input	`getline(infile, s);`	`infile.get(c);`	`infile >> mon >> day >> temp;`

Writing Data from Variables out to a File

Writing data from memory into a file is just as easy:

```
ofstream  outf;         // output stream
string    filename = "results";   // Use -std=c++11 or
                                  // else use char array
outf.open(filename);  // open file
if  (outf.fail())      // check for error
    exit(1);

outf << data;          // same as cout
                       // works for ALL types

outf.close();          // close stream (tell OS you're done)
```

storage

ofstream

Note: Use << to write lines, strings, floats, etc to a file just as you do with cout. Even individual characters.