

## A Programming Problem: Baby Names

ch08/babynames.cpp

```
/**
Reads name information, prints the name if total >= 0, and
adjusts the total.
@param in_file the input stream
@param total the total percentage that should still be
processed
*/
void process_name(istream& in_file, double& total)
{
    string name;
    int count;
    double percent;
    in_file >> name >> count >> percent;
    // Check for failure after each input
    if (in_file.fail()) { return; }
    if (total > 0) { cout << name << " "; }
    total = total - percent;
}
```

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## A Programming Problem: Baby Names

ch08/babynames.cpp

```
int main()
{
    ifstream in_file;
    in_file.open("babynames.txt");
    // Check for failure after opening
    if (in_file.fail()) { return 0; }
    double boy_total = 50;
    double girl_total = 50;
    while (boy_total > 0 || girl_total > 0)
    {
        int rank;
        in_file >> rank;
        if (in_file.fail()) { return 0; }
        cout << rank << " ";
        process_name(in_file, boy_total);
        process_name(in_file, girl_total);
        cout << endl;
    }
    return 0;
}
```

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Reading Text Input (8.2)

There are more ways to read from stream variables than you have seen before, and there are some details you need to understand.

These follow...

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Reading strings

What really happens when reading a string?

```
string word;
in_file >> word;
```

1. If any reading can be done at all, all whitespace is skipped (whitespace is this set: '\t' '\n' ' ').
2. The first character that is not white space is added to the string word. More characters are added until either another white space character occurs, or the end of the file has been reached.

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Reading Characters

It is possible to read a single character – including whitespace characters.

```
char ch;
in_file.get(ch);
```

*cin.get(ch);  
reads from keyboard*

The get method also returns the "not failed" condition so:

```
while (in_file.get(ch))
{
    // Process the character ch
}
```

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Reading the Whole File Character by Character

The get method makes it possible to process a whole file one character at a time.

```
char ch;
while (in_file.get(ch))
{
    // Process the character ch
}
```

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Not Wanting What You .get ()

That which was got  
can be ungot!

```
in_file.unget();
```



Thank you.

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## One-Character Lookahead

You can look at a character after reading it  
and then put it back if you so decide.

However you can only put back  
the very last character that was read.

This is called one-character lookahead.

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Reading a Number Only If It Is a Number

A typical situation for lookahead is to look for numbers  
before reading them so that a failed read won't happen:

```
char ch;  
in_file.get(ch);  
if (isdigit(ch)) // Is this a number?  
{  
    // Put the digit back so that it will  
    // be part of the number we read  
    in_file.unget();  
  
    // Read integer starting with ch  
    int n;  
    data >> n;  
}
```

↑  
ifstream variable declared prior

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Reading a Number Only If It Is a Number

```
isdigit???
```

```
if (isdigit(ch))
```

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Character Functions in <cctype>

The `isdigit` function is one of several  
functions that deal with characters.

`#include <cctype>` is required.

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Character Functions in <cctype>

10/31/12

Table 1 Character Predicate Functions in <cctype>

Function	Accepted Characters
isdigit	0 ... 9
isalpha	a ... z, A ... Z
islower	a ... z
isupper	A ... Z
isalnum	a ... z, A ... Z, 0 ... 9
isspace	White space (space, tab, newline, and the rarely used carriage return, form feed, and vertical tab)

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Reading A Whole Line: `getline`

The function (it's not a method):

`getline()`

is used to read a whole line  
up to the next newline character.

*This is a line of text.*  
*the string*

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Reading A Whole Line: `getline`

stream to read from  
string to read into  
`getline(in_file, line);`  
stops at '\n',  
takes it from the stream  
and throws it away  
– does not store it in string

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Reading A Whole Line: `getline`

Until the next newline character,  
all whitespace and all other characters  
are taken and stored into the string  
– except the newline character – which is just “thrown away”

The *only* type that can be read into is the string type.

```
string line;  
getline(in_file, line); // stops at '\n'
```

↑  
*must be of  
"string" type*

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Reading A Whole Line: `getline`

The `getline` function, like the others we've  
seen, returns the “not failed” condition.

To process a whole file line by line:

```
string line;  
while( getline(in_file, line))  
{  
    // Process line  
}
```

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Processing a File Line by Line

Reading one line and then processing that line,  
taking it apart to get the individual pieces of data from it,  
is a typical way of working with files.

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Processing a File Line by Line

Here is a top secret online CIA file:

<http://www.cia.gov/library/publications/the-world-factbook/>

Don't tell anyone, but it looks like this:

```
China_1330044605  
India_1147995898  
United_States_303824646  
...
```

Each line has: country name, its population, a newline character.  
(And there is some whitespace in there too.)

C++ for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved.

## Processing a File Line by Line

After having copied the secret contents from the website into a text file we will read this file line by line:

```
// we've opened the file
// but we can't tell you it's name or...

string line;
while( getline(in_file, line))
{
    // Process line
}
```

© for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Processing a File Line by Line

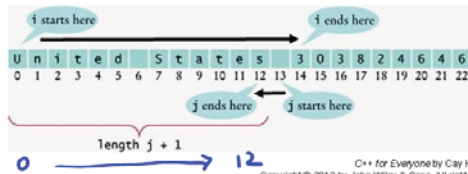
To extract the data from the line variable, you need to find out where the name ends and the number starts.

© for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Processing a File Line by Line

```
// Locate the first digit of population
int i = 0;
while (!isdigit(line[i])) { i++; }
```

```
// Find end of country name
int j = i - 1;
while (isspace(line[j])) { j--; }
```



© for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Processing a File Line by Line

All clear – make the extraction at 09:01 hours.

(that's CIA-speak for: use string methods to "extract" the country name and population values)

(we guess "at 09:01" hours means now)

```
string country_name = line.substr(0, j + 1);
string population = line.substr(i);
```

Handwritten notes: "start index" with an arrow pointing to 0, "length" with an arrow pointing to j+1, and "omit length so understood to go to end of line" with an arrow pointing to i.

© for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Processing a File Line by Line

Has the CIA made a blunder?  
Aren't these being stored as strings?

(Is that a security risk?)

```
string country_name = line.substr(0, j + 1);
string population = line.substr(i);
```

© for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved

## Processing a File Line by Line

No, the CIA is quite capable.

Later (09:01 hours),  
we will see their "nice" (secret) method  
for "extracting" information from  
a (captive) undercover numeric operative  
using `std::string` as an alias:

the **stringstream**  
CONTINGENCY

TO BE  
CONTINUED...

© for Everyone by Cay Horstmann  
Copyright © 2012 by John Wiley & Sons. All rights reserved