

C++ Strings and Things

When we have names or a phrase or a sentence in C, we store it as a *char* type, or actually a pointer to an array of a fixed length. For example, the declaration:

```
char filename[30]; // declare a char array with room for 29 characters
```

In C++, we can use C-style strings and associated commands (such as `sprintf` or `strcmp`) or we can use C++ strings. I recommend C++ strings for several reasons:

- they can be any length, and adjust automatically (i.e., we do *not* declare the length);
- they work with `==` and `+` as you would hope and expect;
- they are overall safer.

Here's a quick guide to the use of C++ strings, with the goal of constructing filenames. The `filename.test.cpp` program illustrates the summary given here.

Action	How to do it
include file for strings	<code>#include <string></code>
declaring	<code>string my_name;</code> <code>string my_name = "Furnstahl";</code>
assigning	<code>my_name = "Smith";</code>
comparing	<code>if (my_name == "Furnstahl") { (do something) }</code>
joining	<code>string my_full_name;</code> <code>first_name = "Dick", last_name = "Furnstahl";</code> <code>my_full_name = first_name + last_name;</code>
printing	<code>cout << "My name is " << my_full_name << endl;</code>
converting to char	<code>my_full_name.c_str();</code>

Suppose we want to open a file with the name stored in the string `my_full_name`. Here's how:

```
ofstream my_file; // declare the output file stream
file.open (my_full_name.c_str()); // notice the use of c.str()
```

If we tried `file.open(my_full_name);` instead of `file.open(my_full_name.c_str());` we would get *many* errors. Now suppose we want to create a filename with the current value of the integer `i` as part of the name. We do this with an "output string stream" or `ostringstream`. Here's what it looks like (note: additional `include` files are needed, see `filename.test.cpp`)

```
int i = 5; // define our integer
ostringstream my_filename_stream; // declare the output string stream
filename_stream << "output_" << i; // just like using cout
string my_filename = filename_stream.str(); // convert the stream to a string
file.open (filename.c_str()); // open the file, using c.str()
```