

CSCI-1200 Computer Science II — Fall 2006

Lab 2 — Strings, Files, Command Line Arguments

Overview

This lab explores the use of command-line arguments, file input and output, and strings. It is an important lab because it introduces tools that will be used in your labs and homework assignments throughout the semester. Create a folder for this lab in your CSII labs directory and download these files:

```
http://www.cs.rpi.edu/academics/courses/fall06/cs2/labs/02_strings_args/simple_stats.cpp
http://www.cs.rpi.edu/academics/courses/fall06/cs2/labs/02_strings_args/in_numbers.txt
http://www.cs.rpi.edu/academics/courses/fall06/cs2/labs/02_strings_args/in_strings.txt
```

Once you have downloaded these files, you need not access the network any further. Turn off all network connections.

Checkpoint 1

Start by reading the code and comments in `simple_stats.cpp`. It contains an explanation of what is needed in a program to handle command-line arguments and to open and read files. Then, just as you did in the last lab, compile/build this program in your favorite C++ development environment.

If you are using g++, run the program from your shell / command prompt by typing:

```
simple_stats.exe in_numbers.txt
```

If you are using Visual Studio, there are two ways to run the program, *try both*:

1. Open a command prompt window and `cd` into the directory that contains the executable for the program. It should be in the `Debug` directory under the project. Copy the downloaded file `in_numbers.txt` into this directory as well. Then you can run the program from the command prompt by typing

```
simple_stats.exe in_numbers.txt
```

If you created the program with a different project name than `simple_stats` you will need to provide a different executable name.

2. You can also run the program from within Visual Studio. This has the advantage of allowing the use of the debugger. Go to `Project->Properties`, select `Debugging` and in `Command arguments` specify the arguments you would like to pass into the program. In our case this is `in_numbers.txt`. (Note that you should not give the name of the program executable, even though it will appear as one of the actual arguments passed to the `main` function in your program.) The `Working directory` is by default the directory of your project files, so you don't need to change it unless you place the files you are using somewhere else. (You should make sure that `in_numbers.txt` is in the working directory.) When you run your program in Visual Studio, these arguments are used as if you entered them from the Command Prompt.

Complete this checkpoint by demonstrating to a TA that you have successfully been able to run the program as described above. (Visual Studio users must show **both** methods).

Checkpoint 2

Write a program that reads all of the strings from an input infile and copies all strings that are of length at least four to an output file, putting one string on each line. Your program will now have 3 arguments, one of which is the name of the program and the other two are the names of the input file and the output file (as `argv[1]` and `argv[2]`, respectively). You should open both the input file and the output file before you start to read from the stream attached to the input file.

Output files are opened using a very similar mechanism to input file opening. In particular,

```
std::ofstream out_str( argv[2] );
```

will open the output file and attach it to stream `out_str`. You can then check whether or not it was successfully opened using

```
if ( out_str ) ...
```

Once you have opened it and checked it, you are ready to use `out_str` in the same manner that you used `std::cout`.

Complete this checkpoint by showing the execution and output of your program on the data set `in_strings.txt`.