HOW TO RUN g++
Principles of Programming Languages 198:314
Spring 1999

We will be running GNU C++ or g++ on the undergraduate SUN servers: remus and romulus. You can run from one of the undergraduate SUNs or from your personal computer, by logging in on one of these servers.

The preferred way to run C++ is from within GnuEmacs (emacs), as then your output is in an emacs window and can be manipulated as text after you exit the interpreter. We prefer to use a two window (i.e., buffer) emacs environment, which encourages an interactive style of editing and running C++ programs. The upper window is used primarily for editing your program. The lower window (which is running the shell) is used for executing programs; it keeps a running record of your execution. Thus, you can save the results of a C++ run by saving this lower window using the usual emacs file saving commands (e.g., Control-x Controls). Because this lower window is saving your execution, if you recompile your program, you should delete the contents of the lower window before re-starting execution, or the old execution record will be saved as well as the new.

TO START: When you have logged on, the command emacs will get you into GnuEmacs. Then, the following command sequence should be executed (where Control-a stands for depressing the control key while hitting the keyboard key a and Esc stands for hitting the escape key by itself). Note, these commands are all GnuEmacs commands to setup your two windows for maximal ease of execution.

Control-x 2 creates two windows (buffers) on your screen, one above the other.

Control-x o puts the cursor in the lower buffer; this is a toggle instruction which can switch you from window to window and back again.

Esc x shell this turns your lower window into an UNIX shell.

After you execute this command, you type into the emacs buffer exactly as you would at the system level. After you end the C++ session, you can exit the shell by typing exit at the prompt. You will remain in the emacs buffer, but it will be transformed into a regular emacs text buffer.

TO RUN: To run your C++ programs, we will be using the GNU C++ compiler called g++. You invoke this compiler thusly: g++ <filename>.

Either you will see error messages from the compiler here or the next system prompt for a successful compile. As with C, the executable is put in file a.out by default which you can run by typing that filename a.out at the system prompt. The results of your cout statements should send output to your screen, unless you have redirected standard output using > in the usual manner. As explained above, you can capture the output of your program easily by running g++ within a shell window in emacs and then cut/paste/save those lines you want.

To help with debugging your syntax/type errors, you can use the emacs Control-u command which lets you repeat an arbitrary emacs command a specified number of times. If you place the cursor at the beginning of your program using Control-x [ and then type Control-u 10 Control-n, the cursor will be placed at the beginning of the 10th line in your buffer. (Control-n skips to the next line in a buffer.) This helps when g++ tells you, for example, that syntax errors occurred on line 10 of your program.

To find execution errors, you can take two approaches. The simpler approach is to trace the behavior of your program by including lots of calls to cout to print things out, so you can find out what’s going on. Remember it’s not wise to mix printfs with couts; use only one type of output in your program.

Alternatively, you can use the debugger. If you wish to use the g++ debugger, you must compile your program with the -g option. Then by typing:

gdb foo.exe

you will run your program under the control of the debugger. Typing help will give you access to the debugger’s extensive help facility.