

CS 132 – Data Structures in C++ - Spring 2005
Week 1 Lab Exercise

Week 1 Lab Exercise due: Wednesday, January 19th, END of lab

Purpose: Practice using the GNU C++ compiler on cs-server.

1. Use ssh (secure shell) to connect to **cs-server**. If you have never had an account in NHW 244 before, let me know now.

2. Create a directory for this lab, make it so that only you can read it, and change to that directory:

```
mkdir 132lab01
chmod 700 132lab01
cd 132lab01
```

3. Use pico, vi, or emacs (if you do not know what those are, use pico!) to open a file that will contain the source code for a C++ function, **nameInLights**. We'll put this source code in a file named **nameInLights.cpp**:

```
pico nameInLights.cpp # or vi or emacs...
```

Now you can paste or type in the code available from the course web page, under “Homeworks and Handouts” in the same section in which you found this handout.

Save your code, and exit. IF you are using pico for the first time, you can save by typing ctrl-o (typing the ctrl key and o key together) and then enter to confirm the file name to save, and you can exit by typing ctrl-x.

4. There is no main function here, so you cannot execute this. You can compile it, however, as so (from the UNIX prompt!):

```
g++ -c nameInLights.cpp
```

The result of this should be a file named `nameInLights.o` --- it is an object file, ready to be linked to a main function.

5. Since this non-main function is all by itself, it needs a header file. Using pico/emacs/vi, copy/type this (from the example given on the course web page) into a file **nameInLights.h**, save, and exit.

6. Use pico/emacs/vi to type or paste in **test_nameInLights.cpp**, whose purpose is to test this function. This code, too, is available from the course web page. Save and exit.

7. This, of course, uses function **nameInLights**, so it must be linked in during compilation. The following (UNIX-level) command for compiling **test_nameInLights.cpp** will also do this linking, and result in an executable file **test_nameInLights**:

```
g++ -o test_nameInLights test_nameInLights.cpp nameInLights.o
```

8. Now you can run **test_nameInLights** by simply typing `test_nameInLights` at the UNIX prompt. To run this program and have its output be written to a file instead of to the screen, one can redirect the output to a file by using `>` and a file name. For example, to write its output to a file named `test_nameInLights_out`:

```
test_nameInLights > test_nameInLights_output
```

9. When you are satisfied with your running program, put your name on the **Next**: list on the board. When it is your turn, you will run your program for me --- when I have verified that it runs, then you have completed the requirements for this lab exercise.

To receive credit for this lab exercise, the above must be completed by the end of the lab period.