

CS 211, Winter 2004

Lab 2 Handin Instructions

Brian M.Dennis, Instructor
Bin Lin, Tom Lechner, Rachel Goldsborough Teaching Assistants

How to Turn In Lab 2

Tom will be the coordinator for lab 2. You will mail your submission to him at: t-lechner@northwestern.edu.

If you completed your assignment in Visual C++, you should **clean your workspace**. This means removing built executables and object files. Then you should create a zip archive of the workspace. Utilities like WinZip

<http://www.winzip.com/>
and FreeZip

<http://members.ozemail.com.au/~nulifetv/freezip/>

make this easy to do. Then e-mail your archive as an attachment named **cs211-lab1-yournetid.zip**.

Did I mention that you should clean your workspace!! The resulting zip archive shouldn't be more than a few KB. Definitely not over 1 MB.

If you completed your assignment on UNIX or Cygwin, try to get all of your source files into one directory. You should provide a **make** file that can build each of the various exercises. Delete any built executables and object files. You should then create a compressed tar file of the directory like so:

```
tar zcvf cs211-lab1-yournetid.tar.gz lab-dir
```

Then e-mail your archive as an attachment named **cs211-lab1-yournetid.tar.gz**

Also, in your e-mail include your name.

Updates from Lab 1

- **Put your name in your e-mail.**
- Be sure to check that the source files needed to build your project are actually in your archive. For Lab 1, a few folks actually forgot to ship those.
- Please follow the naming conventions specified in the homework and in this handout. Both for the actual code and for the archive name. It helps us make sure you get the right grade.
- Actually read the assignment and answer any additional questions over and above the programming. They're not rhetorical in nature. Answers to any additional questions can be placed in a text file named `README.txt` and added to your project.

- We really only want archives in `zip` or `tar.gz` format. We know there are other nice archive formats out there, but currently submissions in these formats just slow us down. If you insist on doing this, we'll insist on grading your assignment last, when we're really grouchy.

Input checking. In Lab 1, we noted situations where your code could crash or loop indefinitely for some input. These were relatively obvious situations such as `fact(-1)` that can be easily handled. However, we did not take points off in these cases.

You may be docked on lab 2 if we see this in your code. Your programs don't have to deal with completely moronic input or operating conditions, but it should deal reasonably with obvious cases. For example, in the last part of the assignment, you are asked to read some values from the command line arguments. Your code should fail gracefully, (e.g. print an error message and exit) if it doesn't see the types or number of arguments that it's expecting.

Collaboration

You should do the assignment on your own. This means that all of the code you hand in should have been written by you. You are free to discuss the homework with your peers and this is actively encouraged. However, after you turn in the homework, if we locked you in a room and gave you the same set of questions, you should be able to solve them completely by yourself and write out the code yourself.

Here's the guidelines from Ian Horswill's version of CS 211, which carries the spirit of what we'd like:

We do still want to encourage you to work together. In particular the following are acceptable and encouraged:

- Groups of friends having "hacking parties" where they sit down at separate machines and program in parallel. They can discuss the problem set as much as they want, as long as they don't actually edit one another's code, or copy one another's code.
- Asking questions about the assignment, or about your program, of friends in the class
- Asking questions on the CS 211 Homework forum on Blackboard
- Asking questions of friends who aren't in the class
- Asking questions of the lab managers in the CS lab
- Getting help with debugging from friends, lab managers, etc.

So basically, you can get lots of help from friends, but you have to (at a minimum) do all the typing.