

CS 211, Winter 2004  
Lab Assignment L4: Handin Instructions  
Assigned: Friday, Feb. 27, 11:00 AM, Due: Friday, Mar. 5, 11:00  
AM

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

## Overall

Tom will be the coordinator for lab 4. 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-lab4-yournetid.zip**.

**Did I mention that you should clean your workspace!!** The resulting zip archive shouldn't be more than a few 100 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-lab4-yournetid.tar.gz cheezo-dir
```

Then e-mail your archive as an attachment named **cs211-lab4-yournetid.tar.gz**

Also, in your e-mail include your name.

For this assignment there are no additional questions being asked outside of the requested classes.

Please follow the naming conventions for the cheezo functions specified in the homework and in this hand-out. Both for the actual code and for the archive name. It helps us make sure you get the right grade.

The same collaboration guidelines as the previous two labs applies.

For this assignment, I have uploaded a small archive of cheezo code to the class website under the labs directory: `cheezo-tests.zip` or `cheezo-tests.tar.gz`. This code should exercise all of the requested features. The code also includes some error expressions although these are commented out. Your interpreter should catch all of these errors and not crash. It's okay to print an error message and exit though.

## **README.txt**

You should include with your code a `README.txt` file that explains how you've gone about changing the cheezo interpreter. This is a plain old English text file.

In your `README.txt` make sure to discuss the following things:

- Your name
- The operating system/tools you built your interpreter under
- A list of files that you added to the interpreter
- A list of handed out files that you modified
- A list of classes that you added to the interpreter and a brief explanation of what they're supposed to do
- The underlying implementation of your list datatype
- The underlying implementation of your vector datatype
- How you dealt with variable binding for lets
- What, of the requested code, you think is working
- What, of the requested code, you took a stab at but you know is broken

This doesn't have to be a tome, but it should be enough that a motivated reader could find their way around your code without stumbling around a lot.

In a program of this size, it's challenging (and a bit unfair to you) to write an automated test suite to grade against. We'll just use a few small tests, but your `README.txt` also helps us know what you know. Thus, the better this is fleshed out, the better off we'll all be.