CSCI 1300
Intro to Computing

Gabe Johnson

Lecture 2   Jan 16, 2013

Computing Environment
and
Basic Python

Lecture Goals

1. Virtual Box
2. Shell Interaction
3. Text and Code Editors
4. Writing and Running Programs
5. Demo of SIMI

Upcoming Homework Assignment

Hello World

The upcoming HW assignment will be released on Friday. It will be pretty easy: it gives you a chance to learn how to make very simple programs, run them, and turn them in to RetroGrade.

I'll introduce RetroGrade next week.

Virtual Box

Need help installing? Go to an installation help session in the CSEL Wireless Cafe (ECCS 128).
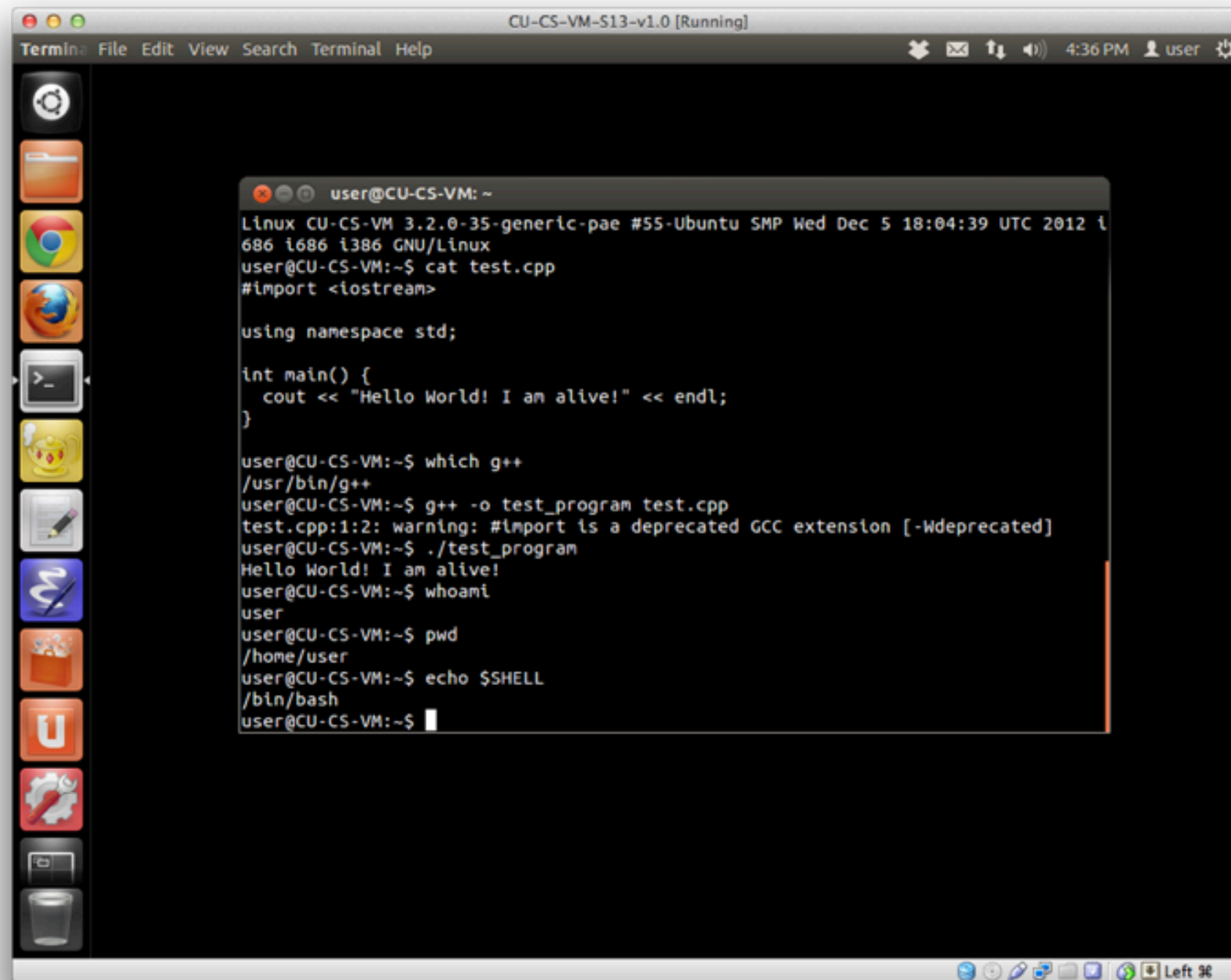
Scheduled Sessions are 1 hour each:
Wed Jan 16 2pm
Fri Jan 18 4pm
Wed Jan 23 12pm
Wed Jan 23 2pm

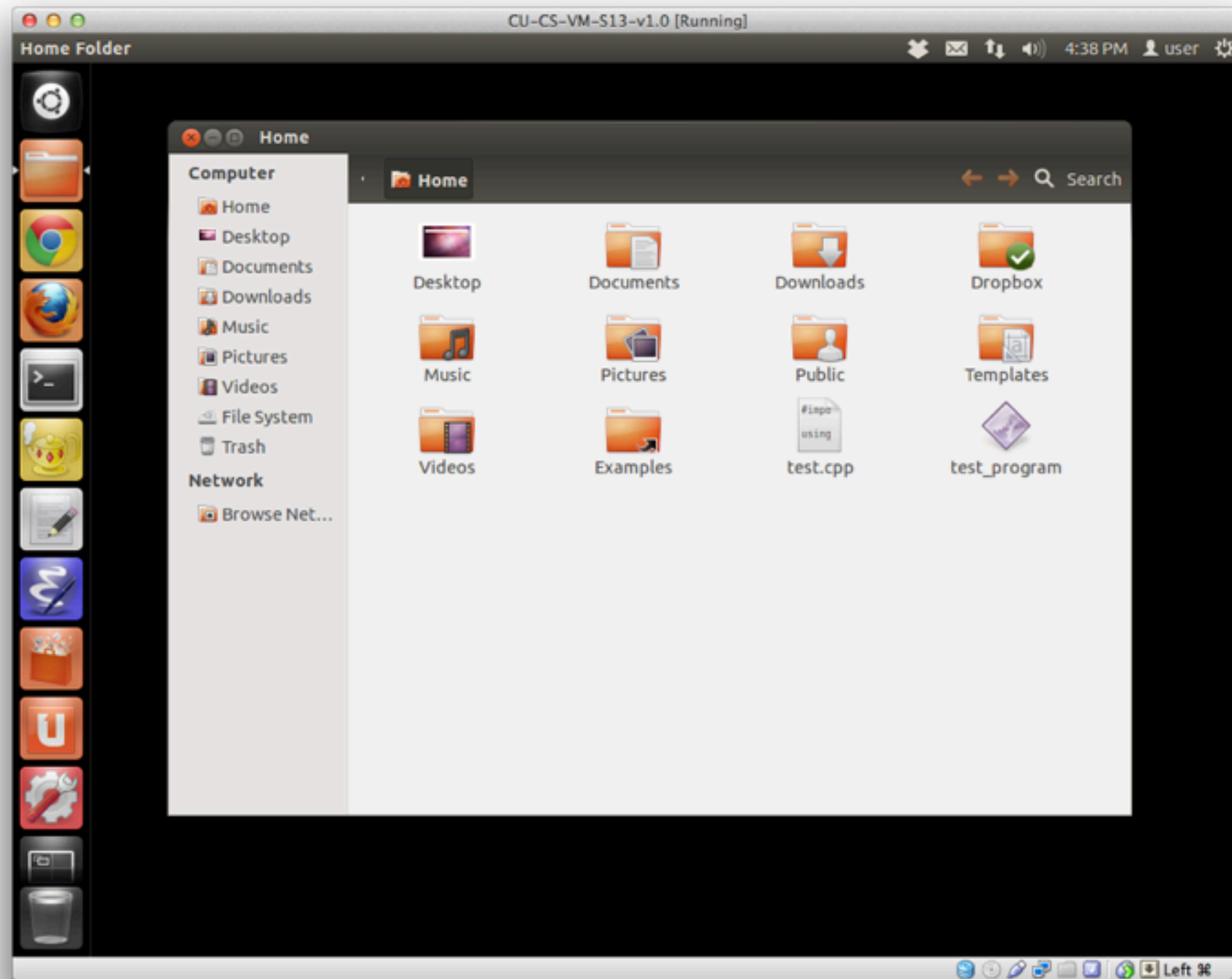Additional sessions by may be scheduled by appointment. Contact andrew.sayler@colorado.edu for more information.
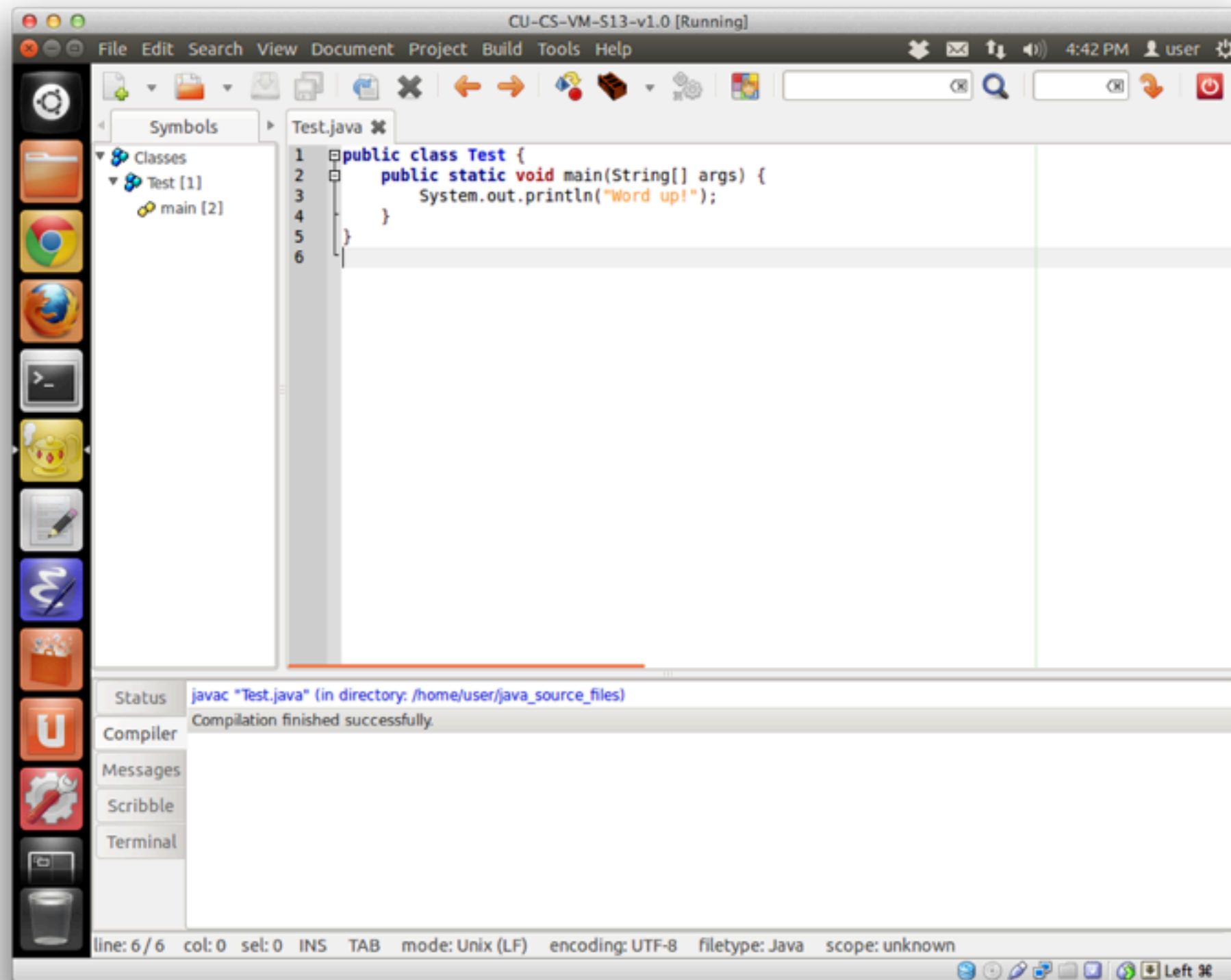
Shell Interaction

File Browser (familiar?)

Text and Code Editors: Geany

Text and Code Editors: gedit
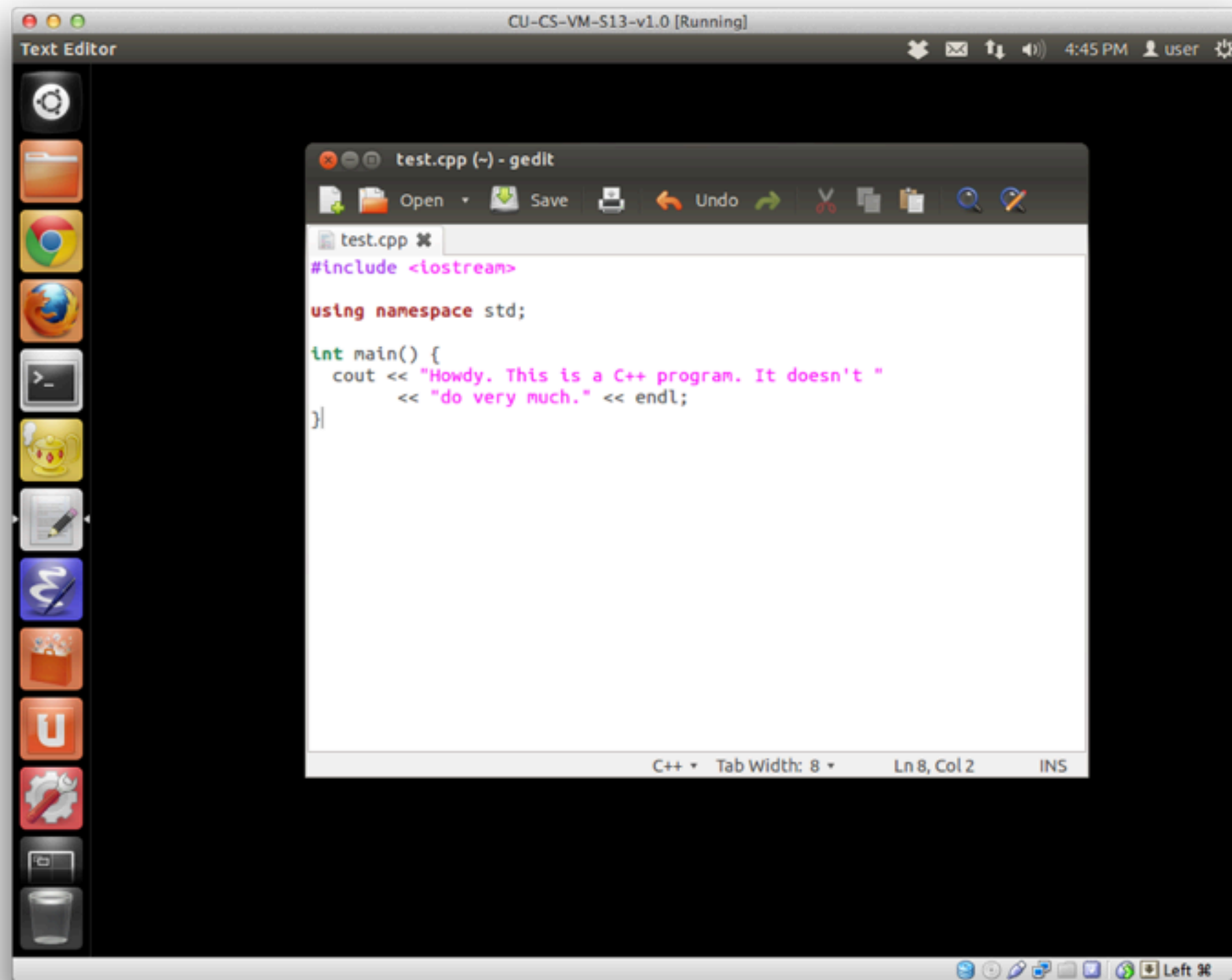
Text and Code Editors: emacs
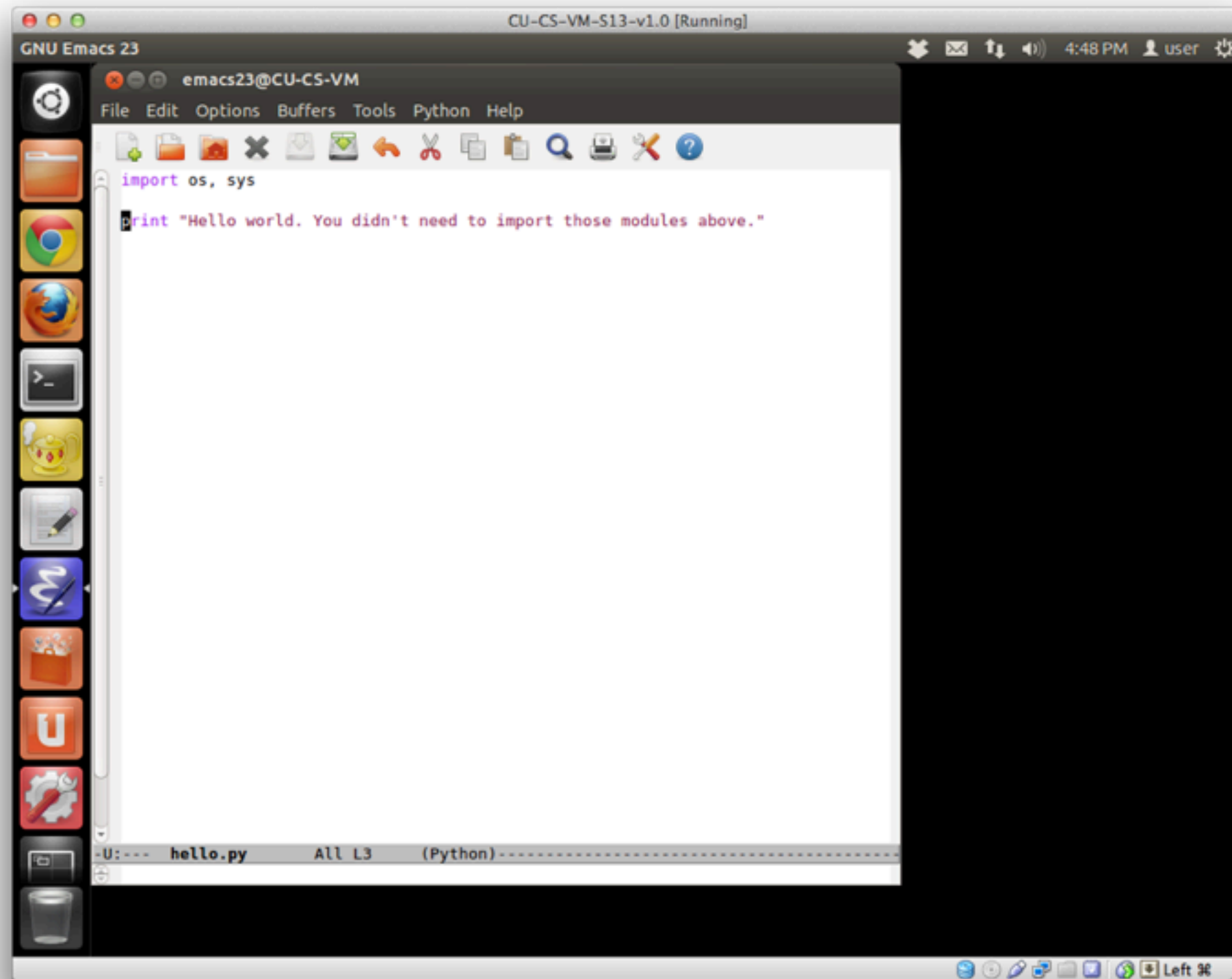
Live Demo

Here's a live demo of the computing environment in the virtual machine, and on my own laptop so you can see how I do it.

Things we will cover:

Use CLI to navigate your file system— change directories, see what files are there, make directories and files, find files, delete files/directories.

Writing Simple Programs

Program, verb: the act of writing text sentences that end up being one of the noun senses of the term.

Program, noun (1): the source code that specifies some sequence of operations that we demand the computer take.

Program, noun (2): the compiled, binary objects the computer directly uses.

Python Example

```
# simplest possible example:

print "Hello World"
```

Python is an interpreted language. This means it directly eats text (like this) and converts it into lower-level code, and runs that low-level code.

Analogies

Try to think of learning to program as analogous to learning a new spoken language. There's a *vocabulary*, some rules about *grammar*, maybe some special *idioms*.

But there's also shared *context* and *experience*.

Syntax and Semantics

Two big words, two fundamentally important concepts.

Syntax

"Syntax" is a fancy word that means how the computer expects your source code to be written.

In English, we call it Spelling (word is put together correctly and Grammar (sentence is put together correctly).

"The slippery dog burns with green dreams."

Valid grammar. Valid syntax.

Semantics

"Semantics" is another fancy word that refers to what your program should do.

In English, we call this "making sense", or "meaning".

"Wish you were here"
"Wish you were her"

Syntactically similar. Semantically, not so much.

Syntax Errors

```
>>> print "Hello World'
  File "<stdin>", line 1
    print "Hello World'
SyntaxError: EOL while scanning string literal
```

Syntax errors will be your first and greatest enemy throughout CS 1300.

What's wrong with the above?

(we are mixing double-quotes and single quotes, and python can't interpret this because it is syntactically incorrect.)

Syntax Errors

```
>>> print "Hello World'
  File "<stdin>", line 1
    print "Hello World'
SyntaxError: EOL while scanning string literal
```

Syntax error **messages** will be your second-greatest enemy throughout CS 1300.

Learn to read these syntax error messages and translate them from nerd-speak into human speak.

Basic Vocabulary Analogy

Spoken languages share concepts, even if they have different words to represent them.

*Dog. Le Chien. Das Hund. El Perro.*該犬*.*

Similarly, any programming language you might be interested in using all draw from the same set of basic concepts. They are just represented with different words.

Basic Programming Concepts

How do you record values?

How do you determine what to execute?

How do you get input from the user?

How do you provide output to the user?

What's the language grammar?

What are the words in the language's dictionary?

How do you give things names?

How do you refer to named things?

How do you have a sense of time?

For Next Time

Command Line + Python = Win!

:: Read up through Exercise 2 in **_Learn Python The Hard Way._**

:: Read as much of **_The Command Line Crash Course_** as you can stand.

(you should be able to do the entire thing in about 5 hours.)

Next Time

We will start programming in Python for real. We'll cover everything really quickly to get a superficial sense of most of the important stuff, then we will go back through and start understanding Python in greater detail.

Remember to read and play along with Zed Shaw's Python book through Exercise 2.