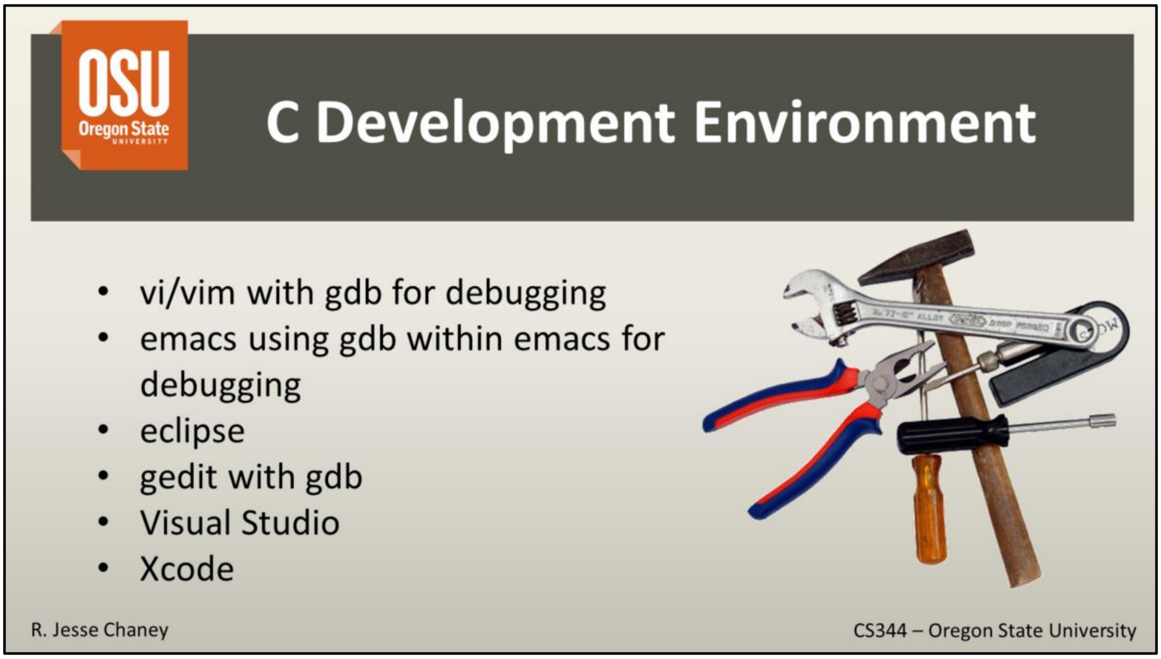
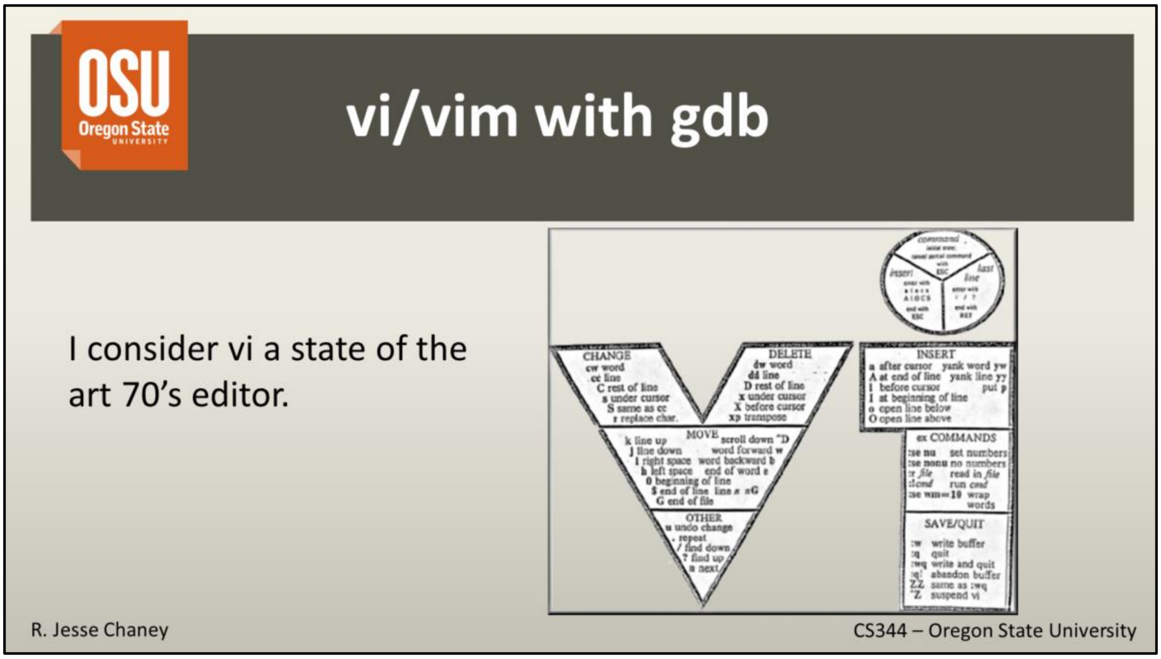


C Development Environment

- vi/vim with gdb for debugging
- emacs using gdb within emacs for debugging
- eclipse
- gedit with gdb
- Visual Studio
- Xcode

An important decision you rapidally have approaching is the choice of development environment. You've got quite a few from which to choose. My recommendation is that you pick one for the term and stick with it through the term. You really don't want to decide on Sunday night that switching to Xcode would be really cool only to find that it will take you a couple of weeks to become proficient enough with it to complete the assignment due on Monday.

Some of the choices are:



OSU Oregon State University

vi/vim with gdb

I consider vi a state of the art 70's editor.

R. Jesse Chaney

CS344 – Oregon State University

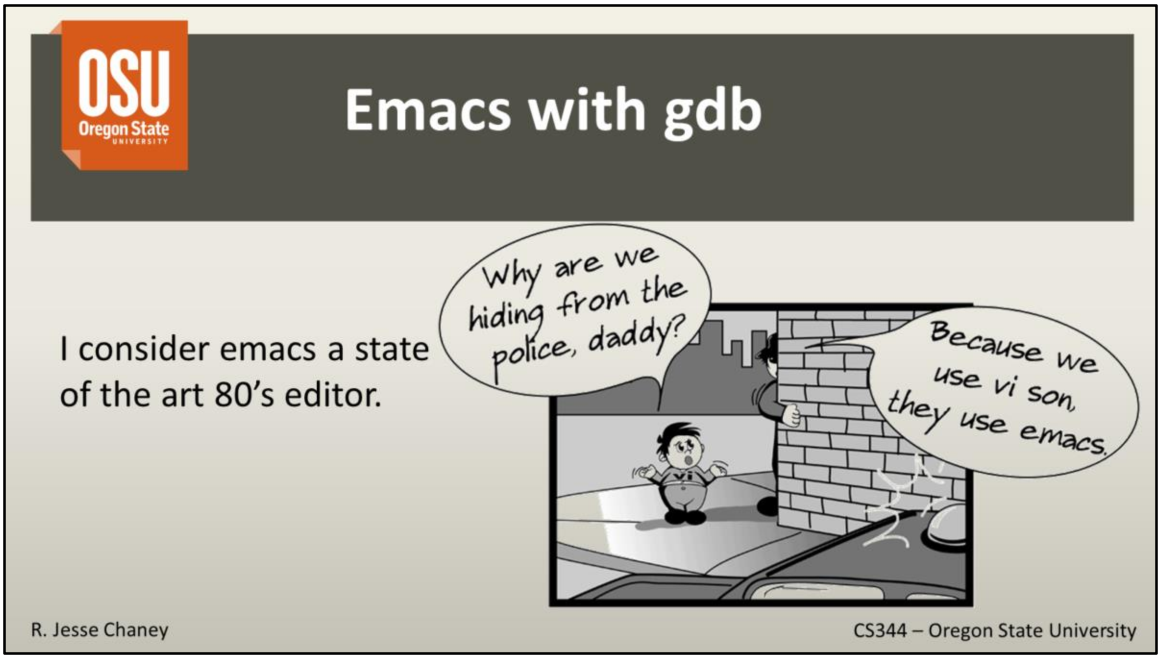
The vi/vim editors are probably available on every Unix/Linux system you'll ever touch. That is a good thing. I know just enough vi commands that I can edit config files or Makefiles when necessary. An important vi command is ":q!". I've used that one quite a few times to bail out of a file I've mistakenly modified.

I guess you could drop pico into this with vi/vim, except that pico is EVEN more basic. But, you won't be spending hours and hours learning the commands.

I consider vi a state of the art 70's editor. So, you are into retro or like classic editors, look no further.

Vi was actually written by Bill Joy (one of the co-founders of Sun Micro Systems, when it existed).

Vi is capable of doing some syntax coloring while editing.



Emacs with gdb

I consider emacs a state of the art 80's editor.

OSU
Oregon State
UNIVERSITY
Why are we hiding from the police, daddy?
Because we use vi son, they use emacs.
vi

R. Jesse Chaney

CS344 – Oregon State University

The emacs editor kind of grew out of a need to have something with more flexability/power that vi.

The name actually stands for Editing MACroS, but quite a few alternative meanings have been suggested:
Escape-Meta-Alt-Control-Shift and Even a Master of Arts Comes Simpler
http://www.gnu.org/fun/jokes/gnuemacs.acro.exp.html.

I used to be really good with emacs and using gdb within emacs. Emacs is an amazingly flexible editor. It is almost like its own operating system. You can even read email within emacs.

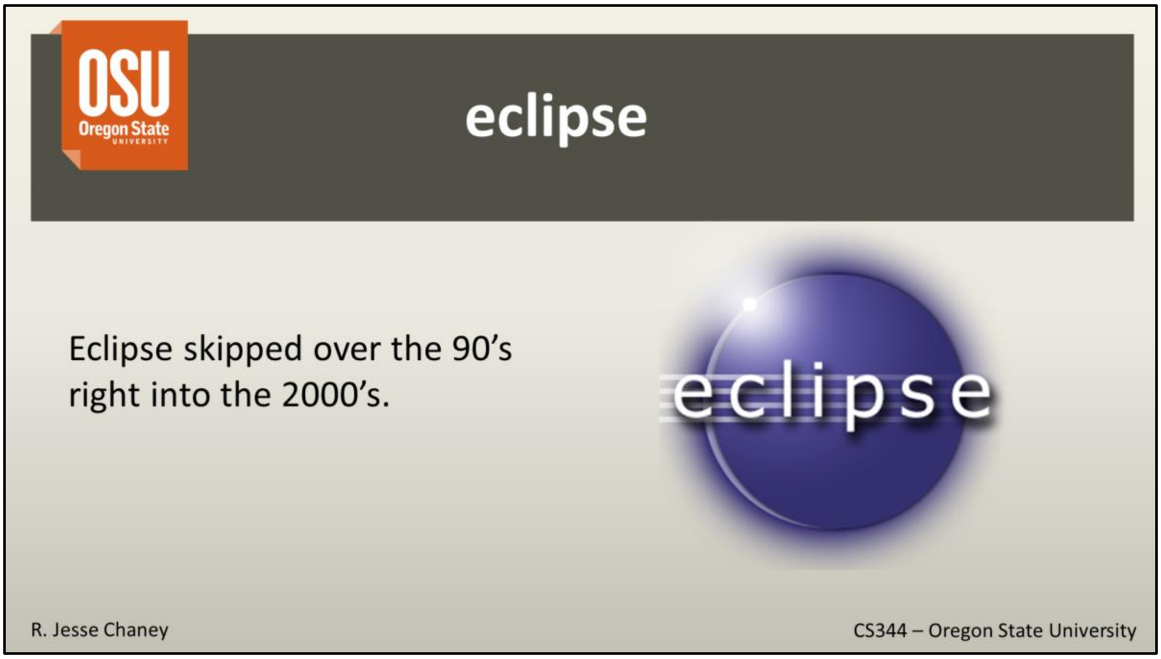
There is a story that (many years ago), MIT students wrote an emacs macro that would call the elevator to a floor in one of their labs. If you really get into emacs, you can program some pretty amazing macros for it. I've done a few, but none such as a reviewed on this web page:
http://www.ibm.com/developerworks/ibm/library/lol/pinhead.html.

I consider emacs a state of the art 80's editor. The truth is that development on emacs began around the same time as development on vi.

The vi vs emacs flame wars have been going on for a long time and will continue to go for a long time.

The original emacs was written by Richard Stallman (of FSF fame) and Guy Steele (of Common Lisp fame).

If you do get to using emacs, make sure you look at the Towers of Hanoi demo and the psychoanalyze-pinhead function.



OSU Oregon State University

eclipse

Eclipse skipped over the 90’s right into the 2000’s.

R. Jesse Chaney

CS344 – Oregon State University

Eclipse , like emacs, has become an entire system in itself.

I think of eclipse as the successor to emacs. As **BIG** as emacs has become, I think eclipse has surpassed it for consuming even more resources. Eclipse is actually much more than JUST an editor. It has all sorts of stuff that have been built for it.

Before you get to use eclipse, you have to load a JVM within which is must run. If you are going to install a JVM for eclipse, I strongly encourage the actual Oracle/Sun JVM. The OpenJDK has never quite done it for me.

When possible, I tend to use eclipse for my development of both C and Python. The integrated debugger is quite nice and the syntax highlighting is excellent. However, to use eclipse, I really need to be logged into the console on the system I’m using. This is fine for my personal Linux workstation or when using a VirtualBox VM, but not so great when working through a ssh connection into OSU on os-class.

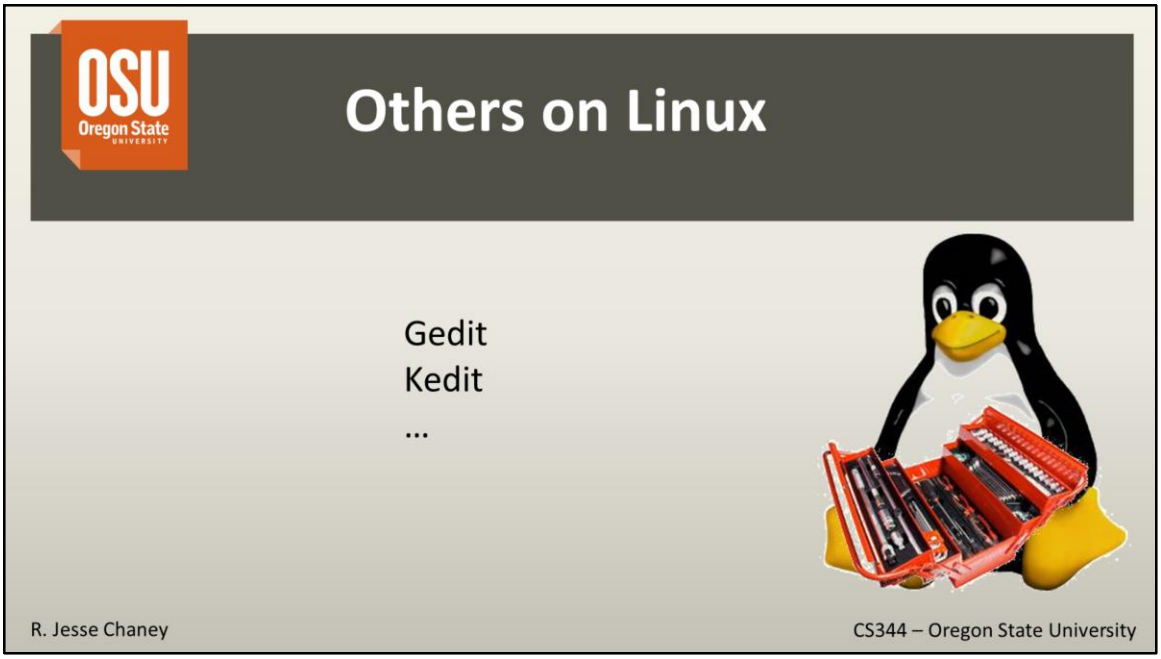
The short answer here is while eclipse may be great, possibly even the best, it also will probably not be the only one you’ll need to be able to use.

Eclipse is also a common choice of Java developers and very common in industry. For this reason, knowing how to use eclipse is a good idea.

There are some tools that allow you to edit remote projects using eclipse, but may be difficult for you to get going for just a 1 term class.

Eclipse actually started out as a Smart Canada project. I have no idea what a Smart Canada project is, but we got eclipse out of it.

How many of you have built a virtual system using something like VMware workstation or VirtualBox? If you want to use eclipse on your own Linux system, that may be the best way to go.



Others on Linux

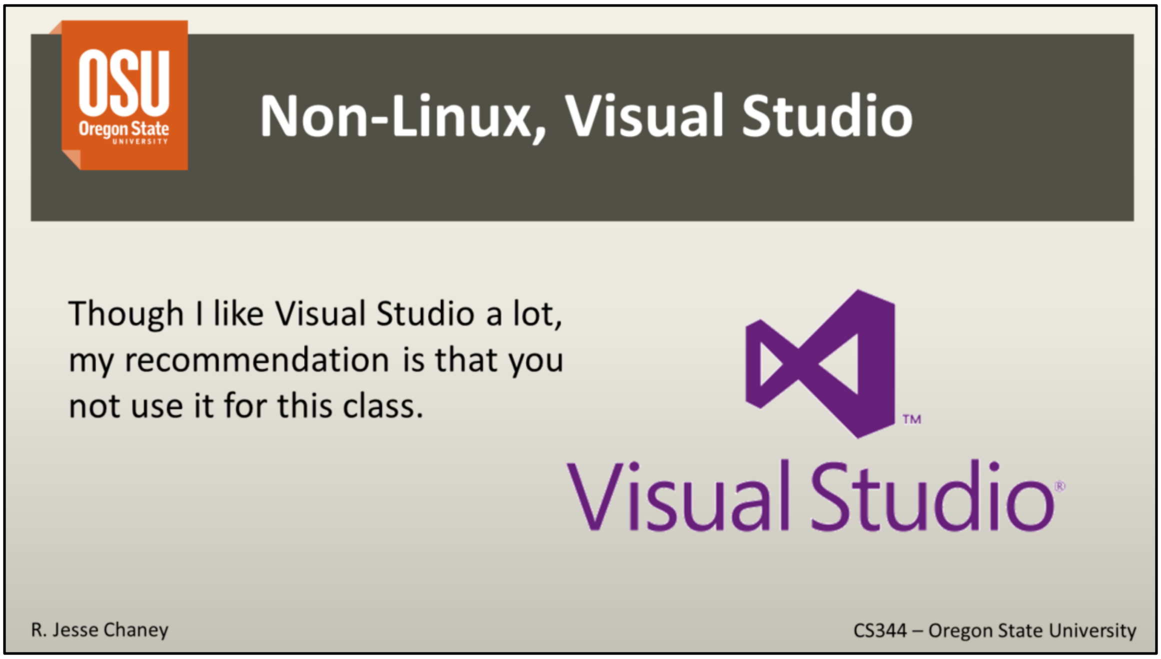
Gedit
Kedit
…

R. Jesse Chaney

CS344 – Oregon State University

Gedit is a nice simple editor to use when on the **console** of a Linux system. It does not give you very much, but does not ask a lot of you either. You'll be getting comfy with gdb if using gedit.

There is a plethora of other editors available on Linux.



OSU
Oregon State
UNIVERSITY

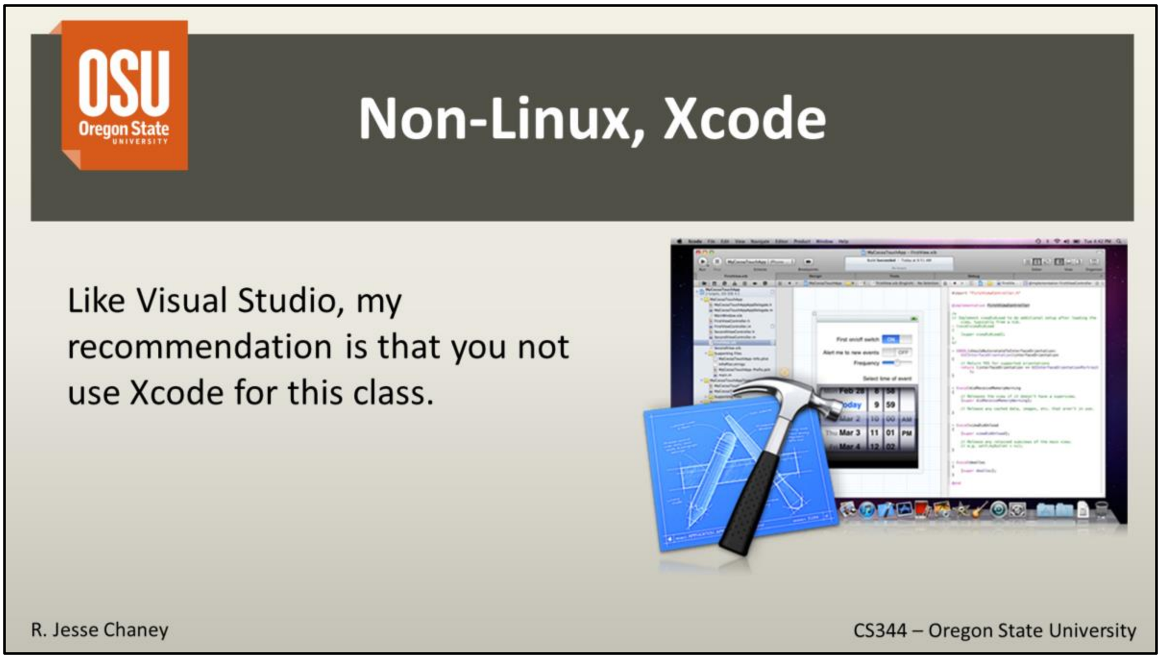
Non-Linux, Visual Studio

Though I like Visual Studio a lot, my recommendation is that you not use it for this class.

Visual Studio

An environment that I discourage you using is Visual Studio. I don’t discourage it because I don’t like it, but because I think it will make it difficult for you to do your development in it and feel confident that it will run on os-class.

I like Visual Studio quite a bit and use it a lot, but not for Linux development. If you like it enough to put the time and energy into making it work for you through a double secret handshake ftp tunnel, that's great.



Non-Linux, Xcode

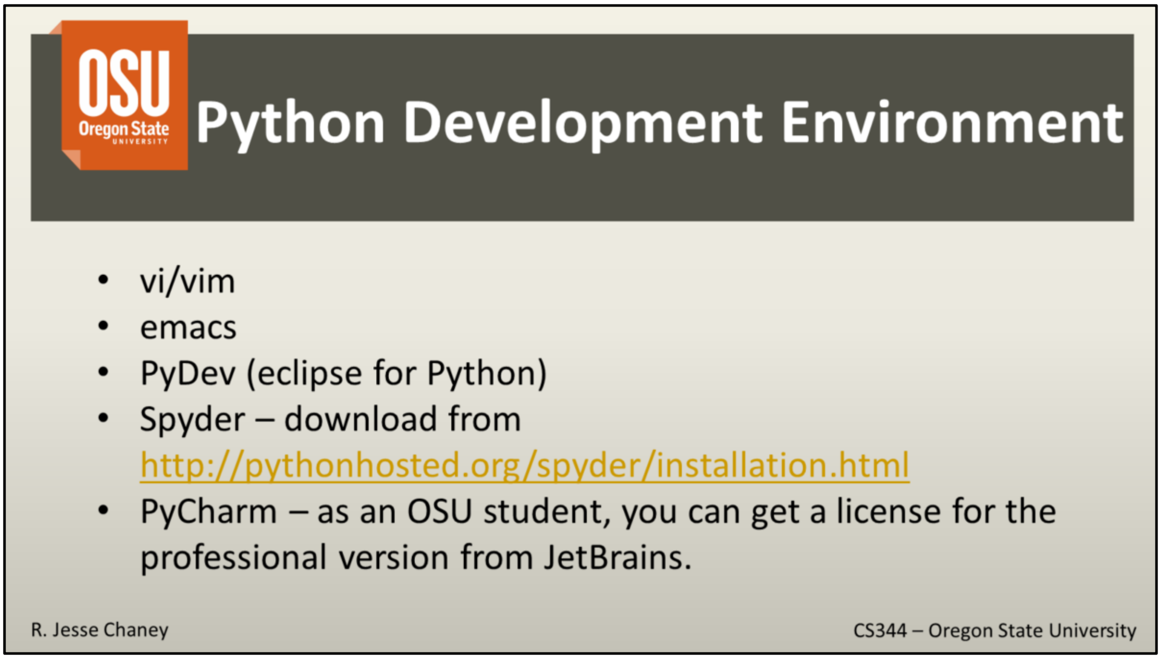
Like Visual Studio, my recommendation is that you not use Xcode for this class.

R. Jesse Chaney

CS344 – Oregon State University

My thoughts on Xcode are pretty much the same as Visual Studio, though I've not used Xcode very much.

I do know that though OS X (the Mac operating system) is a Unix-like operating system, DO NOT expect it to behave like a Unix/Linux/POSIX operating system. I spent quite a few hours debugging some simple code on a Mac one night (and early morning), thinking that surly the Mac would do the right thing. I won't be getting those hours of my life back because the code worked great when I eventually tested it on Linux. I also won't make that mistake again.



Python Development Environment

- vi/vim
- emacs
- PyDev (eclipse for Python)
- Spyder – download from http://pythonhosted.org/spyder/installation.html
- PyCharm – as an OSU student, you can get a license for the professional version from JetBrains.

R. Jesse Chaney

CS344 – Oregon State University

Any of these is a fine choice for Python development. What I really (REALLY) want to encourage you to use is an editor that is Python aware. Using Notepad is not something I recommend. With Python the difference between tabs and spaces is very important. Spaces are very important to Python.

For the quickest start, Spyder is quite nice. You can use it to install Python on your laptop and do your development on your laptop, with only limited testing necessary on os-class. It has an intergraded debugger and can install Python along with the Sypder installation.

R. Jesse Chaney

CS344 – Oregon State University

When t comes right down to it, I don’t care what you use to write your code, only that you write it (and debug it of course). And that it be done on time, and be readable, work correctly, and with integrity.

Whatever choice you make for a development environment is fine, just get the code done. That said, you also won’t be getting a lot of support if your .emacs file macros go wacko or if you mistakenly load a binary file into vi and don’t know how to exit and your terminal is now beeping 3 times a second. I think emacs is the most powerful environment that will work across all connections (both console logins and ssh logins), but getting good at emacs takes time and in your heart of hearts, you know that you can only can give it about 8 seconds. The vi/vim editors are not nearly as powerful, but will consume less of your time learning and will work across all types of connections. Eclipse may be the best, but does have its requirement for being on the console.

If you can get some special ftp double encrypted connection to work so you can use eclipse on your laptop and automatically update code on os-class, please write up a how-to and share it with us.