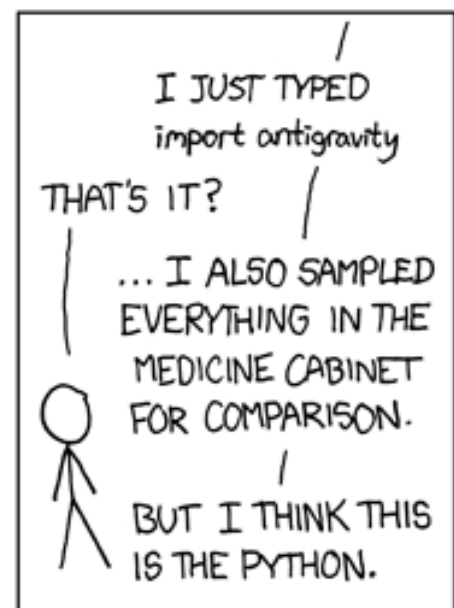
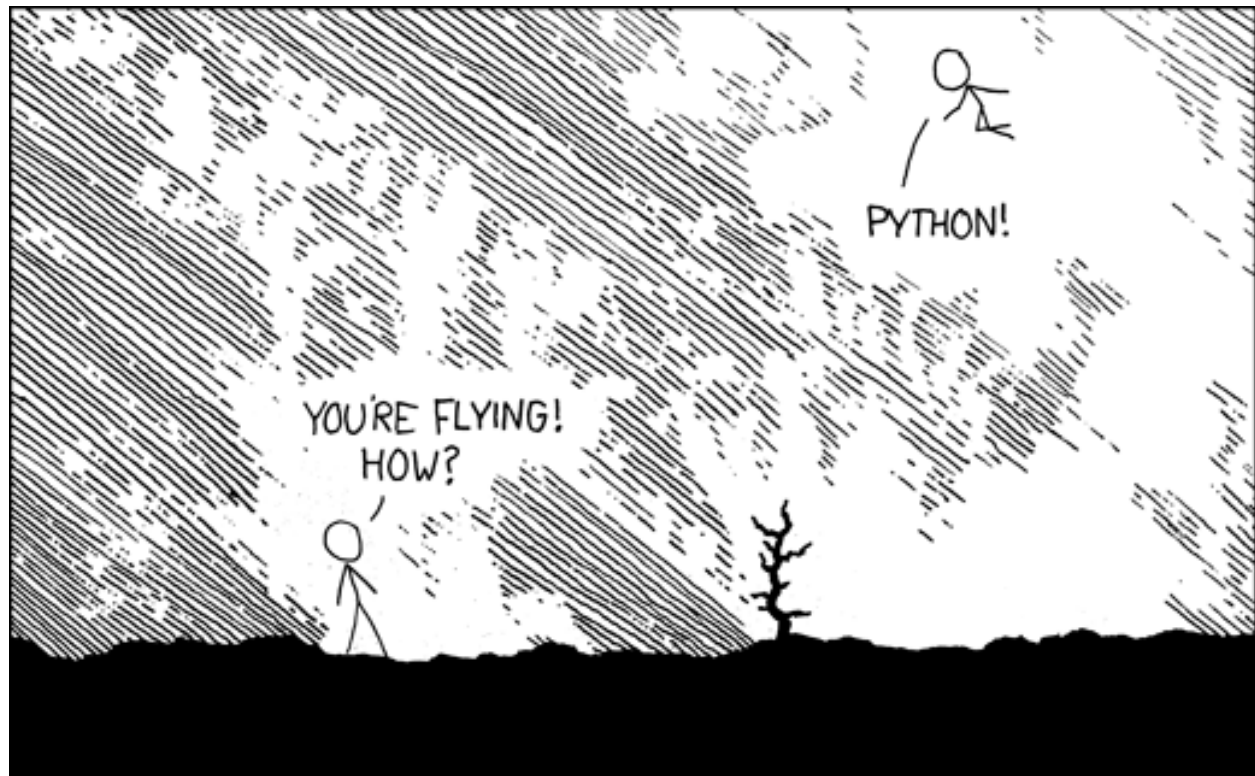


Syllabus for Computer Science 200

Introduction to Information Systems, Spring 2018



MWF 10:30-11:20, HLH 17, room 05

Stephen Slade

014 AKW, 432-1246 stephen.slade@yale.edu

Office hours: Wednesday 2 to 4 pm, and by appointment.

Teaching Assistants: Please see [Instructor and TA contact information](#).

Course Description

Information technology is ubiquitous. This course focuses on the real world artifacts and implementations that comprise the vital computational organisms that populate our world. Topics include hardware (computer architecture), software (programming languages, databases, networking), and related issues such as security, and software engineering. Examples stress practical applications of technology. Homework assignments will generally involve programming in Python.

Textbooks

- [Learning Python, 5th Edition](#). By Mark Lutz. Publisher: O'Reilly Media. June 2013. Pages: 1540 [Online copy at Yale Library](#) [Errata and source code from book](#)

Complete with quizzes, exercises, and helpful illustrations, this easy-to-follow, self-paced tutorial gets you started with both Python 2.7 and 3.3 — the latest releases in the 3.X and 2.X lines — plus all other releases in common use today. You'll also learn some advanced language features that recently have become more common in Python code.

- [Python Cookbook, 3rd Edition](#). Recipes for Mastering Python 3. By David Beazley, Brian K. Jones. Publisher: O'Reilly Media. Final Release Date: May 2013. Pages: 706 [Online copy at Yale Library](#) [Github source code from book](#)

If you need help writing programs in Python 3, or want to update older Python 2 code, this book is just the ticket. Packed with practical recipes written and tested with Python 3.3, this unique cookbook is for experienced Python programmers who want to focus on modern tools and idioms

- [Effective Python](#). Brett Slatkin. Paperback: 256 pages. Publisher: Addison-Wesley Professional; 1 edition (March 8, 2015). [Online copy at Yale Library](#)

It's easy to start writing code with Python: that's why the language is so immensely popular. However, Python has unique strengths, charms, and expressivity that can be hard to grasp at first -- as well as hidden pitfalls that can easily trip you up if you aren't aware of them. Effective Python will help

you harness the full power of Python to write exceptionally robust, efficient, maintainable, and well-performing code. Utilizing the concise, scenario-driven style pioneered in Scott Meyers's best-selling Effective C++, Brett Slatkin brings together 59 Python best practices, tips, shortcuts, and realistic code examples from expert programmers.

Other Resources

Web page

The course web page is at <http://zoo.cs.yale.edu/classes/cs200>.

Piazza

Students will be enrolled in a piazza site for the course. which permits and interactive exchange of questions and information: www.piazza.com Note: students are not allowed to post code to piazza.

Zoo accounts

The Zoo is a collection of computers located on the 3rd floor of AKW at the front of the building. You will need a *course account* on the Zoo to submit homework. Sign up for a CS 200 course account on the Zoo using the on-line signup procedure at <http://zoo.cs.yale.edu/accounts.html>. There will be *help sessions* on using the Zoo early in the term. A Zoo tutorial is available on-line from the course web page.

Course directory

The course directory, `/c/cs200` is accessible from your Zoo course account. It contains copies of handouts.

Course Requirements

The course requirements consist of class attendance, (more-or-less) weekly programming assignments in Python and occasional written homework, and two EVENING MIDTERMS and a FINAL EXAM. Plan on spending between 6-8 hours per week on the course outside of class. The programming assignments are an integral part of the course.

Please try not to leave the homework to the last minute. You will be more efficient, learn more, have more chance to get help, and generally be calmer and happier if you do the associated reading first and start the programming or other problems early.

Grading

The final grade in the course will be based on class participation, your performance on the programming assignments and other homework, and the exams. The weighting of these components will be approximately 40% on homework and 60% on exams.

Late Policy

Late work without a Dean's excuse will be assessed a penalty of 5 points per day, based on the day and time recorded by the Zoo electronic submit program. At the end of term, up to 25 points will be deducted from the total lateness penalties your homework has accrued.

Policy on Working Together

Unless otherwise specified, the homework assignments are your individual responsibility. Plagiarism is a violation of University rules and will not be tolerated. You must neither copy work from others nor allow your own work to be copied. You are definitely on the wrong side of the boundary if you give or receive a printed or electronic copy of your or anyone else's work for the course from this term or previous terms. Working in groups to solve homework problems is not permitted.

Course Outline

Week	Date	Topic
1	Jan 17, 19	Introduction. Python: strings, lists, UNIX, recursion.
2	Jan 22, 24, 26	Python: sorting, dicts, files, testing
3	Jan 29, 31, Feb 2	Python: regular expressions, utilities, UNIX shell scripts
4	Feb 5, 7, 9	Python: object oriented programming
5	Feb 12, 14, 16	Python: databases (oop)
6	Feb 19, 21, 23	Python: SQL
7	Feb 26, 28, Mar 2	Review for midterm I. Computer architecture
	Mar 1	Evening midterm 7-8:30pm DL 220
8	Mar 5, 7, 9	Computer Architecture. Python Virtual Machine.
9	Mar 26, 28, 30	Cryptography and Information Security
10	Apr 2, 4, 6	Cryptography and Information Security
11	Apr 9, 11, 13	Review for midterm II. security, encryption
	Apr 12	Evening midterm 7-8:30pm RTBA
12	Apr 16, 18, 20	Machine learning
13	Apr 23, 25, 27	Machine learning
	May 4, 2pm	Final Exam, RTBA

Last modified: 03/08/2018 11:51:20