Welcome to SI182
Building Information Environments
Data Analysis in Python

Charles Severance

Textbook: Python Programming: An Introduction to Computer Science, John Zelle (www.si182.com)
What is the Course About?

- A first programming course - but different
- Focus on skills that you will use regardless of your major
- Prepare you for further programming courses if you find programming interesting
- Part of understanding how to make programming more of a basic and useful skill across all majors
Informatics Curriculum

- New undergraduate LSA major *under development*
- Mathematics
- Statistics
- School of Information
- Electrical Engineering and Computer Science (EECS)
Informatics Major Structure

- Core courses - All students
- Informatics Tracks - Advanced Courses
  - Information Analysis
  - Social Computing
  - Computational Informatics
  - Life Sciences Informatics
Syllabus
drchuck
Preparing for SI182. Almost done.
less than 10 seconds ago from web

Played hockey. Felt good. Got a nice assist. about 10 hours ago from text

This so cool – human Tetris http://tinyurl.com/y5573isq about 13 hours ago from web

I love the IBM Innovation and Ideation commercials. Watching football. about 15 hours ago from web

Breakfast at Golden Harvest. about 22 hours ago from text

Watching video about standard deviation http://tinyurl.com/ynpy89 07:26 PM January 05, 2008 from web

Dr. Chuck Goes Vintage Motorsports Racing

Dr. Charles Severance
Clinical Assistant Professor
Office Hours: by appointment
Email: csev@umich.edu

www.dr-chuck.com
www.dr-chuck.com/csev-blog
twitter.com/drchuck/
www.dr-chuck.com/media.php
www.dr-chuck.com/images/
• My previous job: Sakai / CTools Architect

• My research topics: Software For Teaching and Learning, Web Lecture technologies, and High Performance Computing.

• I also work in developing standards for learning software interoperability

• Hobbies: Hockey, Off-Road Motorcycle Riding
Course Site

- Two sites
  - Semi-public - auditors and helpers
  - Private - Primarily grading
- Mailing list
  - si182@ctools.umich.edu
- Please use it like a conversation
Textbook

- PYTHON Programming: An Introduction to Computer Science
- John Zelle
- Franklin, Beedle, and Associates, www.fbeeldle.com
- Part of a movement to use Python as the first programming language.
Why Python?

• Many languages to choose from: FORTRAN, C++, Ruby, C, Java, Perl,
• Balance three things
  • Easy to use and learn - Not ugly and clunky
  • Powerful - You don’t have to write a lot of code to do what you want
  • Popular - You will meet others who can help you
Why Python?

- Python is one of the most popular languages for part-time programmers
- Python also scales to Enterprise levels such as Google
Why Python in SI182?

- Quick to learn
- Powerful
- Data Oriented
- Popular for writing small programs to manipulate data
History of Python

• In early 1980s Guido van Rossum was working on a language called ABC aimed at teaching programming to non-programmer -

• Guido developed Python starting in 1986 by adding things which were missing in ABC to solve real-world problems

“Personal computers had all this wonderful packaged software that dealt in files. There was a spreadsheet file, a word processor file, a graphics editor file. The ABC users wanted to write little ABC programs that took something from their word processor file and pushed it back into the spreadsheet, or the other way around, but they couldn't because of the limitation on IO.”
A key element of Python is the ability to extend it in any language. This has led to a rich set of add-on modules for Python which have been developed over the past 20 years.

“I came up with a flexible extensibility model for Python. I said: "We'll provide a bunch of built-in object types, such as dictionaries, lists, the various kinds of numbers, and strings, to the language. But we'll also make it easy for third-party programmers to add their own object types to the system."
What’s in a Name?

- Python was named after “Monty Python’ Flying Circus” - although many of the books and web sites have images of Python snakes.

http://www.youtube.com/watch?v=teMlv3ripSM
History of Python

- So ultimately, Python is a very extensible language which has its beginnings in - being easy to learn, easy to teach, focused on data, and yet very powerful and easy to extend.

- References:
  - http://www.python.org/~guido/
Python Software

- Mac OS/X 10.5 - Already installed - you can upgrade from pythonmac
- Mac OS/X 10.4 - http://www.pythonmac.org/packages/
- Windows Lab Computer - USB Stick
  - http://www.portablepython.com/site/download/
Do you have a Laptop?

- Life is simplest if you have a laptop - it is best when you just install Python on your laptop and work wherever you like
- If you do not have a laptop - you need a USB stick - at least 1GB
- We will work this all out in Lab - Wednesday
Helping Others

• Please ask for and/or give help

• In the beginning this is very foggy - hard to find the big picture

• But remember that your purpose is to learn

• Ask the mailing list - post code bits - it is OK

Chuck’s Basic Rules

• Coming late or leaving early - OK
• Sleeping in class - OK
• Using a laptop - OK
• Eating or drinking - OK if the room permits it
• Stepping out to take a bio break - OK
• Asking questions at any time - OK
• Correcting me when I make a mistake - OK
• Skipping class - not very wise - but OK
• Doing things that distract other students or making difficult for us all to learn - Not OK
• Skipping class or sleeping in class and then expecting me to repeat entire lectures in office hours - Not OK
• Waiting to the last minute and asking me to review the whole semester in office hours - Not OK
Programming Assignments

• Probably the most important part of the course - our task is to teach you programming skill - so you can ultimately do this on your own

• Increasing difficulty over time

• I monitor areas of difficulty and adjust the course material, lectures, assignment difficulty, everything

• Sometimes it helps to do the assignments twice if you are struggling

• Must be handed in on time - so I can distribute sample solutions
Exams

• Two practical exams

• Come to Lab - I hand out a programming problem - must finish and hand in within 2 hours - open book, open notes, open laptop, can look at your old programming assignments, surf the web - just no help from other people.

• Two written exams - classic stuff on paper

• Short answer, multiple choice, read code and tell what it does, very little code writing - page of notes
Participation

• In class-exercises - handed out - collected - not formally graded
• In class questions / comments
• Helping other students
• Giving me feedback about the course
• ...

Grading

- Approximate percentages
- Assignments: 50% Exams: 40% Participation: 10%
- Straight scale from written syllabus
Course Outline

- Since this is the first time in the class - the outline is just a starting point - I will adjust as I think of better ways to do things

- Even though the outline is just five days old - I am already thinking about a new approach to the material - “Sprint and then Fill-in”
What we have to teach

- Techniques / Tactics - Python Skillz and Visualization Skillz
- Data Analysis Concepts / Strategy - Informatic
- Programming Concepts / Strategy - Computer Science

- Initially I will focus as much as I can on the techniques - we will skip high level stuff - we will come back and fill in the blanks later
Phase I: Get to the Good Stuff

- Programming: 1.5 - 1.7
- Simple programs: 2.2 - 2.8
- Numbers and Types: 3.1, 3.6
- String data and Files: 4.1 - 4.3, 4.4* 4.5.1, 4.6
- Functions: 6.2, 6.4, 6.5.1 through page 178

- Decision Structures: 7.1, 7.2
- Loop Structures: 8.1 - 8.3 not 8.3.4
- Lists: 11.2
- Dictionaries: 11.6.1
Handling Mail Data

- Our motivating application for the first few weeks will be an E-Mail data set.
- We will take this data set and take it apart piece by piece.
From stephen.marquard@uct.ac.za Sat Jan  5 09:14:16 2008
Return-Path: <postmaster@collab.sakaiproject.org>
Received: from murder (mail.umich.edu [141.211.14.90])
    by frankenstein.mail.umich.edu (Cyrus v2.3.8) with LMTPA;
    Sat, 05 Jan 2008 09:14:16 -0500
To: source@collab.sakaiproject.org
From: stephen.marquard@uct.ac.za
Subject: [sakai] svn commit: r39772 - content/branches/sakai_2-5-x/content-impl/impl/src/java/org/sakaiproject/content/impl

Details: http://source.sakaiproject.org/viewsvn/?view=rev&rev=39772
|
Author: stephen.marquard@uct.ac.za
Date: 2008-01-05 09:12:07 -0500 (Sat, 05 Jan 2008)
New Revision: 39772

Modified:
content/branches/sakai_2-5-x/content-impl/impl/src/java/org/sakaiproject/content/impl/ContentServiceSqlOracle.java
content/branches/sakai_2-5-x/content-impl/impl/src/java/org/sakaiproject/content/impl/DbContentService.java

Log:
SAK-12501 merge to 2-5-x: r39622, r39624:5, r39632:3 (resolve conflict from differing linebreaks for r39622)
Some Possible Assignments

- How many lines are in the file
- How many messages are in the file
- Which programmer did the most work on Sakai?
- Which school did the most work on Sakai?
- Which continent did the most work on Sakai?
- Draw a picture of the world and superimpose the data on that picture.
Success in The Course

• Don’t wait until the last minute each week

• If you get stuck on something - move around - review some material - read the book - then come back

• When you look back - you will see that this was all *really* easy

• When you feel stuck - communicate - use the list - ask a friend

• I need to get feedback - a lot
Beware of Overconfidence

- Students who have some prior experience may be at some disadvantage because the class may seem too easy and/or too slow.
- Start to skip lectures and labs - just do the assignments by themselves.
- Once the course starts to speed up - they get lost quickly and find themselves a few weeks behind.
- Solution: Come to class and lecture and catch up on E-Mail with one ear on the material. Also help beginning students to make sure *you* understand.
No Experience Required

• I am committed to teaching the course to students with no prior experience in programming.

• I will alter the pace and/or order of the material as I see a need based on how well students are doing.

• Make sure to let me know on the mailing list, or by private mail or talking to be in lecture or lab how you think we are doing - or if you missed something.
Welcome to the course... 

- Any questions?