What is the Course About?

• A second programming course - but different

• Focus on design skills, assuming you already know some programming
Where it fits?

• A core course for Informatics students. Elective for CS students.

• The new course refactors the computing material so that most-needed skills are covered first:
  • Application of data structures
  • Object-oriented programming and testing
  • Applied databases
Informatics Curriculum

- New undergraduate LSA concentration
- Mathematics
- Statistics
- School of Information
- Electrical Engineering and Computer Science (EECS)
Informatics Structure

- Core courses - All students
- Informatics Tracks - Advanced Courses
  - Information Analysis
  - Social Computing
  - Computational Informatics
  - Life Sciences Informatics
Informatics information:
http://informatics.umich.edu
Teaching Staff

- Professor Atul Prakash, Computer Science (aprakash)
- TAs: Vijay Alexander (vjalex), Scott Reed (reedscot)
• My education: undergraduate at IIT Delhi, Ph.D. at University of California, Berkeley in Computer Science

• My research topics: Computer security, software systems.

• Some recent work in the news: bank’ web site security analysis

---

Security holes in online banking sites found to be widespread

By Mark Der Hovanesian

AMANDAL, Min. (Ams. Jan. 31) — Hackers have infected 91 per cent of the nation’s 3,900 banks and credit unions, according to a survey by the University of Michigan computer science professor who studied 214 sites. His 2006 survey, presented at the July 25 Symposium on Usable Privacy & Security, found 75% of sites vulnerable to hacking, with two big worrisome trends: log-in boxes placed on insecure pages on a bank’s domain and the use of third-party vendors that transfer customers to insecure outside pages (via, say, a “Contact Us” link). “Banks should try to keep the site on a single domain,” Prakash says, adding that users should look for a URL starting with “https” on pages asking for sensitive personal data. The “s” signals an extra security layer.
Course Communication Tools

- Primary Course Portal:
  - http://ctools.umich.edu

- Public page: http://www.eecs.umich.edu/~aprakash/eecs282
Download Java Software  
(will take a while)

• Login to ctools now and see the announcement for the precise links

• Java: [http://java.sun.com](http://java.sun.com): Download the latest SE version

• Eclipse: [http://www.eclipse.org](http://www.eclipse.org) (Download the latest Java Developer version)
Getting Help

- Use the discussion forum on 282 site in ctools. Use it like a conversation, ask questions, provide tips, and help others

- Email to eecs282@umich.edu (but forum is preferred)

- Office hours and labs
Textbooks

Free book available as download

Optional

Nice book on how to do testing
Why Java?

• One of the popular programming languages (others being C++, Python, PHP)

• Write-once, run-anywhere philosophy (unlike C++)

• Safer programming than in C++

• Faster than scripting languages (e.g., Python)

• Other languages use Java-like syntax: Javascript, C#
What can you do in Java?

• Do general programming: Many libraries for visualization, building graphical user interfaces, networking, multimedia, etc.

• The "brains" behind many web sites are written in Java (e.g., J2EE, Java Server Pages, and Tomcat)

• Java applets: used in online calculators and animations

• Google Android mobile phone applications

• See http://java.sun.com/nav/used
History of Java

• Developed in 1994 by Sun Microsystems.
  Key architect: James Gosling

• Hardware-independent language designed
to work on a variety of digital devices (set-top boxes, cellphones, Windows, Unix, …)

• Originally called "Oak" at Sun.
When a Program Runs...

- When a program runs it:
  - Takes some input data
  - Processes the data using a set of instructions (a program)
  - Produces some output
  - Think of it as “value add”

An example program takes a text file as its input
- counts the lines in the file
- prints out the number of lines in the file
Do you have a Laptop?

- Life is simplest if you have a laptop - it is best when you just install Java software on your laptop and work wherever you like.

- If you do not have a laptop - you need a USB stick. The hardware in the lab is in a state of flux.

- We will work this all out in labs.
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. Submitted work must be in your own style and done by you. See Engineering Honor Code Policy at course ctools site and read syllabus

• Forum - post code bits - it is OK

Atul’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 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 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

• Generally, doable in the lab + 4-8 hours of weekly self-study/work.

• Includes small exercises during the class

• 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
Exams

- Two exams
- Open book/open notes
Grading

- Assignments: 45%, Mini-exercises: 5%, Exams: 50%

- Easy to get points:
  - In-class mini-exercises (solutions provided. Effort is what counts)
  - Do key parts of your assignments in the lab when help is available
  - Straight scale from written syllabus
Bonus Points

• There will be opportunities for bonus points on the assignments. They just get added to your score.

• If you are borderline at a grade boundary, I may factor in:
  
  • How well you finish the course (e.g., final performance)
  
  • Communication during the semester.

• Clever insights, creative solutions, and anything else that I did not anticipate
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 phorum - ask a friend
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 other students to make sure *you* understand.