Welcome to CS61B!

- Sometime this weekend (we hope), you’ll be able to get an account form electronically using your Calnet ID. Watch the class web page:
  
  http://inst.eecs.berkeley.edu/~cs61b

  Get an account and register electronically before your first lab. This is unrelated to TeleBEARS registration.

- After registering, please fill out our background survey.

- If you plan to work from home, try logging in remotely to one of the instructional servers.

- Discussion sections and labs start next week. Go to any sections, labs where you fit. If you have a Monday discussion (which won’t meet next week), go to another discussion section next week, so as not to miss anything.

- We’re working on taking care of those on the waiting lists because of full sections, but this won’t be resolved before next week.
Welcome, Continued

• We’ll be using Piazzza for notices, on-line discussions, questions.

• See General Course Information on web page for info on grading, lateness, cheating policy, etc.

• Lectures will be be screencast.
Texts

• There are two readers currently on-line (see the website).

• I will have paper copies at Vick Copy (not Copy Central), corner of Hearst and Euclid, when I get a count of those who want one.

• You could do without printed versions, except that we don’t allow computers in tests (but do allow printed stuff).

• Textbook (for first part of the course only) is *Head First Java*. It’s kind of silly, but has the necessary material.
Course Organization

• You read; we illustrate.

• Labs are important: exercise of programming principles as well as practical dirty details go there. Generally we will give you homework points for doing them.

• Homework is important, but really not graded: use it as you see fit and *turn it in!* You get points for just putting some reasonable effort into it.

• Individual projects are *really* important! Expect to learn a lot. Projects are *not* team efforts (that’s for later courses).

• Use of tools *is* part of the course. Programming takes place in a *programming environment*:
  
  - Handles editing, debugging, compilation, archiving versions.
  
  - Here, we keep it simple: Emacs + gjdb + make + svn, (documented in one of the readers and on-line). Eclipse is OK, too.

• Tests are challenging: better to stay on top than to cram.

• Tests, 45%; Projects, 45%; HW, 10%

• Stressed? Tell us!
Programming, not Java

• Here, we learn *programming*, not Java (or Unix, or Windows, or...)

• Programming principles span many languages
  - Look for connections.
  - Syntax \((x+y)\) vs. \((+ x y)\) is superficial.
  - E.g., Java, Python, and Scheme have a lot in common.

• Whether you use GUIs, text interfaces, or embedded systems, important ideas are the same.
For next time

- Please do HW #0 (not turned in), which is due Tuesday night. Do it if you want to be prepared for class!

- Please read Chapter 1 of *Head First Java*, plus §1.1-1.9 of the on-line book *A Java Reference*, available on the class website and in the second part of the first reader.

- This is an overview of most of Java’s features.

- We’ll start looking at examples on Wednesday.

- Always remember the questions that come up when you read something we assign:
  - Who knows? We might have made a mistake.
  - Feel free to ask at the start of lectures, or by email.
Acronyms of Wisdom

DBC

RTFM
public class Hello {

    public static void main(String... args) {
        System.out.println("Hello, world!");
    }
}

Advertisement

- The Berkeley Programming Contest is approaching (late September).
- We use it as a qualifying trial for the ACM regional contest in November.
- So, if you know any real hotshots (or are one yourself) tell them about this opportunity to show that they have what it takes.