Welcome to CS61B!

- Get an account form and then register electronically with o
  ware using the "Account Administration" link on the class we

  `http://inst.eecs.berkeley.edu/~cs61b`

  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 on
  instructional servers.
- Discussion sections and labs start next week. Go to any s
  labs where you fit.
- We're working on taking care of those on the waiting lists
  of full sections, but this won't be resolved before next weel
- We'll be using Piazzza for notices, on-line discussions, quest
- See *General Course Information* on web page for info on
  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), co
  Hearst and Euclid, when I get a count of those who want one
- You could do without printed versions, *except* that we don
  computers in tests (but do allow printed stuff).
- Textbook (for first part of the course only) is *Head First Ja*
  kind of silly, but has the necessary material.

Course Organization

- You read; we illustrate.
- Labs are important: exercise of programming principles as practical dirty details go there. Generally we will give you ho points for doing them.
- Homework is important, but really not graded: use it as you and *turn it in!* You get points for just putting some reasonable into it.
- Individual projects are *really* important! Expect to learn a lo are *not* team efforts (that's for later courses).
- Use of tools *is* part of the course. Programming takes pl *programming environment:*
  - Handles editing, debugging, compilation, archiving version
  - Here, we keep it simple: Emacs + gjdb + make + svn, (docu 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
- 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 syste portant ideas are the same.

For next time

- Please read Chapter 1 of *Head First Java*, plus §1.1–1.9 of the book *A Java Reference*, available on the class website and 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 rea thing 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

Advertisement

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