

Welcome to CS61B!

- Discussion sections start this week. *Get* an account (if needed) and try to register electronically using the class website.
- Go to any sections, labs where you fit. *DON'T* worry about where TeleBEARS thinks you are.
- We'll take care of those on the waiting lists because of full sections soon.
- Class website and newsgroup set up: read them regularly!
- See *General Course Information* on web page for info on grading, lateness, cheating policy, etc.

Texts

- There are two readers at Vick Copy (*not* Copy Central), corner of Hearst and Euclid.
- These readers are also on-line.
- You can do without printed versions, *except* that we don't allow computers in tests (but do allow printed stuff)
- No other required textbooks (they're too expensive, anyway).
- For those of you wanting extra (less referency) stuff on Java, last semester's textbook was *Head First Java*.

Course Organization

- You read; we illustrate.
- Labs are important: practical dirty details go there.
- Homework is important, but really not graded: use it as you see fit and *turn it in!* You get points for just doing that.
- Individual projects are *really* important! Expect to learn a lot.
- Use of tools *is* part of the course. Programming takes place in a *programming environment*:
 - Handles editing, debugging, compilation, archiving versions.
 - We'll see Eclipse in lab.
 - Or there are coordinated suites of tools (e.g., Emacs + gjdb + make + svn).
- 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 NT, or...)
- Programming principles span many languages
 - Look for connections.
 - Syntax ($x+y$ vs. $(+ x y)$) is superficial.
 - E.g., Java 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 read Chapter 1 of Reader #1 (Java Reference).
- This is an overview of most of Java's features.
- We'll start looking at examples on Friday.
- 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.

Advertisement

- The Berkeley Programming Contest is approaching!
- 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.