Announcements:

- Course surveys TODAY: Bonus points for filling out the survey (HKN is here to help). Get your code from the sheets that we will circulate to put on your final for credit.
- Scheme Art Judging next week (watch the website). Entries will be posted after 1 May (Monday).
- If you have regrade requests (or other grade issues), please get them to us by next Wednesday.
- Topic review sessions next week. See website for schedule.
- Guerilla section on Scheme, tail calls, interpreters, and SQL Saturday 4/29, 12-3PM in 247 Cory.
- Otherwise, no standard office hours next week, except mine (which
  may get rescheduled, however).
- Otherwise, no standard office hours next week.
- Section practice sessions on Tuesday, all labs, interpreters, and SQL Saturday.
- Topic review sessions next week. See website for schedule.
- If you have regrade requests (or other grade issues), please get
  them to us by next Wednesday.
- Submit your grading next week: submit all the work (papers)
  that will be graded. It is best to do this on your own, but if
  you are having trouble or other issues, please get
  them to us by next Wednesday.
- Course surveys TODAY: Bonus points for filling out the survey (HKN is here to help). Get your code from the sheets that we will circulate to put on your final for credit.
Software Engineering

- Biggest ideas: Abstraction, separation of concerns
- Specification of a program vs. its implementation
  - Syntactic spec (header) vs. semantic spec (comment)
- Example of multiple implementations for the same abstract behavior

Testing: for every program, there is a test.
- In "Extreme Programming" there is a test for every module.

Software engineering implicit in all our software courses, explicit in CS169.

Analysis

- What we can measure when we measure speed:
  - Raw time.
  - Counts of selected representative operations.
  - Symbolic expressions of running time.
- Looking at worst cases simplifies the problem (and is useful).
- Application of asymptotic notation ($\Theta$, etc.) to summarizing symbolic time measurements concisely.

Important Side Excursions

- Cryptography:
  - protecting integrity, privacy, and authenticity of data.
  - Symmetric (DES, Enigma) and asymmetric (public-key) methods.
- Computability [CS172]: Some functions cannot be computed. Problems that are "near" such functions cannot be computed quickly.

What's Next (Course-Wise)?

- CS61B: (conventional) data structures and languages
- CS61C: computing hardware as programmers see it.
- CSC100: Data Science
- CS170, CS172, CS174: "Theory"—analysis and construction of algorithms, theoretical models of computation, use of probabilistic algorithms and analysis.
- CS161: Security
- CS162: Operating systems
- CS164: Implementation of programming languages
- CS168: Introduction to the Internet
- CS160, CS169: User interfaces, software engineering
- CS176: Computational Biology
- CS188, CS189: Artificial intelligence, Machine Learning
- CS184: Graphics
- CS199 and research projects.

What's Next (Course-Wise) (II)

- CS186: Databases
- CS195: Social Implications of Computing
- CS C149: Embedded Systems.
- CS 150: Digital Systems Design
- CS194: Special topics. (E.g.) computational photography and image processing, crypography, cyberwar.
- Plus graduate courses on these subjects and more.

There's Also Electrical Engineering

- EE105: Microelectronic Devices and Circuits.
- EE118, EE134: Optical Engineering, Photovoltaic Devices.
- EE120: Signals and Systems.
- EE126: Probability and Random Processes.
- EE130: Integrated Circuit Devices.
- EE137A: Power Circuits.
- EE140: Linear Integrated Circuits (analog circuits, amplifiers).
- EE143: Microfabrication Technology.
- EE147: Micromechanical Systems (MEMS).
- EE192: Mechatronic Design.

Software Engineering

- biggest ideas: Abstraction, separation of concepts
- specification of a program vs. its implementation
  - example of multiple implementations for the same abstract behavior
- testing: for every program, there is a test.
- in “extreme programming,” there is a test for every module.
- examples of multiple implementations for the same abstract behavior
  - symbolic expressions of running time
  - counts of selected representative operations
  - raw time
- what we can measure when we measure speed
What's Next (Otherwise)?

• Programming contests.
• Still more paradigms and languages: the web.
• The open-source world: Go out and build something.
• And above all: Have Fun!