



UC Berkeley  
Computer Science  
Lecturer SOE  
Dan Garcia

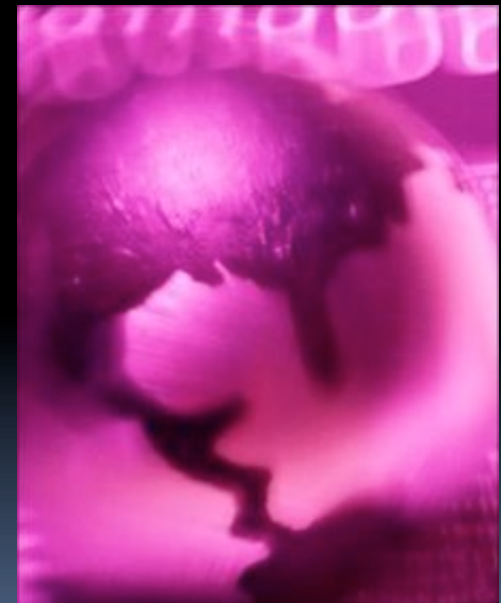
# CS10 The Beauty and Joy of Computing

## Lecture #1 Welcome; Abstraction

2011-8-29

### ALGORITHMS CONTROL THE WORLD?

The algorithm is rapidly taking over vital functions of businesses, from the next Pandora song, to suggesting what to buy on Amazon, to evaluating whether a movie was going to be a hit, to making trades, etc. They're running a Chi-NY dedicated fiber so an algorithm can close deals 3ms faster!



[www.bbc.co.uk/news/technology-14306146](http://www.bbc.co.uk/news/technology-14306146)

# CS 61AS (Structure & Interpretation of Computer Programs) Self-paced

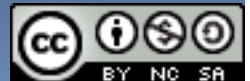
- **Same material as CS61A (but in Scheme not Python)**
- **Self-paced in that**
  - The course is meant to adapt to whatever level you are!!
  - If you have NO programming experience, we work with you
    - You'll go through CS3S material to get you ramped up
  - Lab-centric learning at your own pace
  - You may watch CS61A videos on your own pace
  - You take mini-quizzes to assess your progress
  - Depending on how many book chapters you've finished, you'll take one of several different final exams
  - Option to do additional work for Honors designation
- **If you don't finish all 4 units in one semester, you can continue on to CS61B[L] and finish them concurrently**
- **If you're EECS or L&S CS, consider this option...**



# Design constraints of CS10

---

- **CS61A expects program. experience, recursion**
  - CS10 hits that in week 5, just about the same time as CS3
- **What should ugrads know about computing?**
  - Computational Thinking
  - History, CS+X, Industry guests
  - apps that changed the world, hot research
  - “How stuff works” ... demystifying computing
- **Passion, Beauty, Joy & Awe**
  - Take every step to make fun for non-traditional students
- **Make all resources free, available (Berkeley way)**
  - Videos, notes, exercises, book!



# Non-majors: Out with CS3, In with CS10

## ■ CS3S & CS3L

- Programming, programming, programming
  - Prog Ideas: Recursion, Functions-as-data
- Scheme
  - + Same as CS61A
  - some take CS3L for wrong reason
  - Never remix code
  - Maybe graphical, interactive by week 15
- 1 big Final project

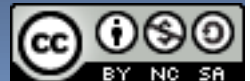
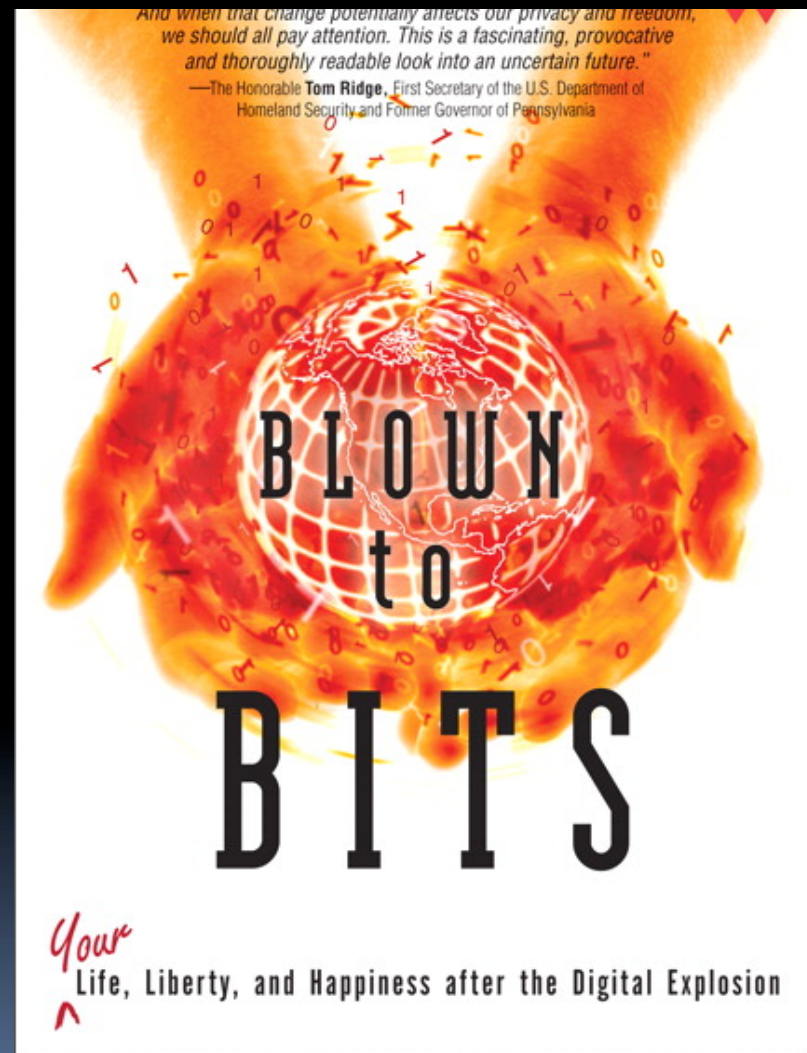
## ■ CS10

- Programming 1/2 story
  - Big ideas, HowStuffWorks, history, great applications, social implications too!
  - Prog Ideas: Recursion, Functions-as-data
- Scratch + BYOB
  - CS10,61[ABC] each in a different language
  - Graphical, interactive, musical by week 2
  - Share and upload code!
- Two projects + essay



# Format, Textbooks, Grading

- **Format**
  - Two 1-hr lectures / wk
  - Two 2-hr labs / wk
  - One 1-hr TA discussion/wk
- **Selected Reading**
  - Taken from recent books and papers
- **Grading**
  - Quest, Midterm, Final
  - One paper (or blog)
  - Midterm project
  - Final project
  - Weekly readings & HW
  - Effort, Participation, Altruism



# Peer Instruction

- Increase real-time learning in lecture, test understanding of concepts vs. details
- As complete a “segment” ask multiple choice question
  - 1-2 minutes to decide yourself
  - 2 minutes in pairs/triples to reach consensus. Teach others!
  - 2 minute discussion of answers, questions, clarifications



# Piazza for {ask,answer}ing questions

The screenshot displays the Piazza website interface. At the top, there is a search bar and navigation links for 'CS 10 Questions', 'Statistics', and 'Add Question/Note'. The user 'Dan Garcia' is logged in. Below the search bar, there are popular tags like '#instructor-question', '#admin', '#logistics', and '#welcome'. The main content area is divided into a 'QUESTION FEED' on the left and a detailed view of a question on the right. The question is 'When are TA / professor office hours?' and has been answered by 'instructors' with the text: 'We haven't established our office hours yet, but we'll make that information available as soon as possible. Check back here for an update by the second week of classes.' The interface includes buttons for 'Good Question!', 'Good Answer!', 'Ask a Followup', and 'Start off a Students' Response'. At the bottom, there are statistics for 'AVERAGE RESPONSE TIME' (N/A), 'SPECIAL MENTIONS' (Luke Segars answered in 1.1 hr), and 'USERS ONLINE THIS WEEK' (3).



# Abstraction

- **Detail removal**
  - “The act or process of leaving out of consideration one or more properties of a complex object so as to attend to others.”
- **Generalization**
  - “The process of formulating general concepts by abstracting common properties of instances”



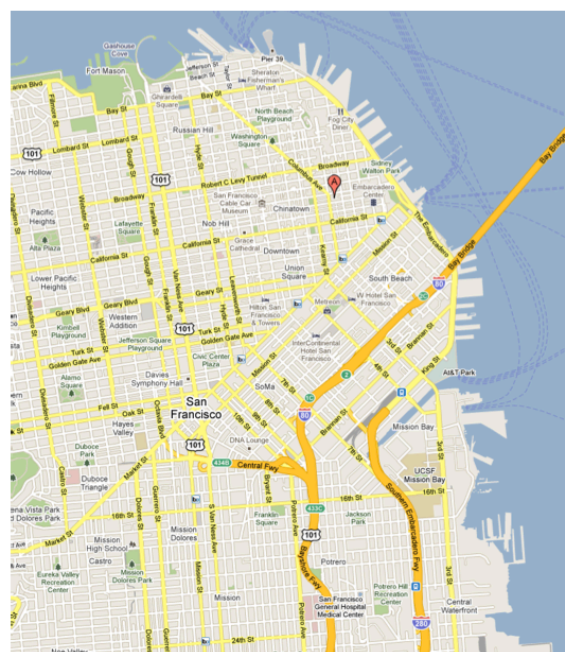
Henri Matisse “Naked Blue IV”

Garcia, Fall 2011





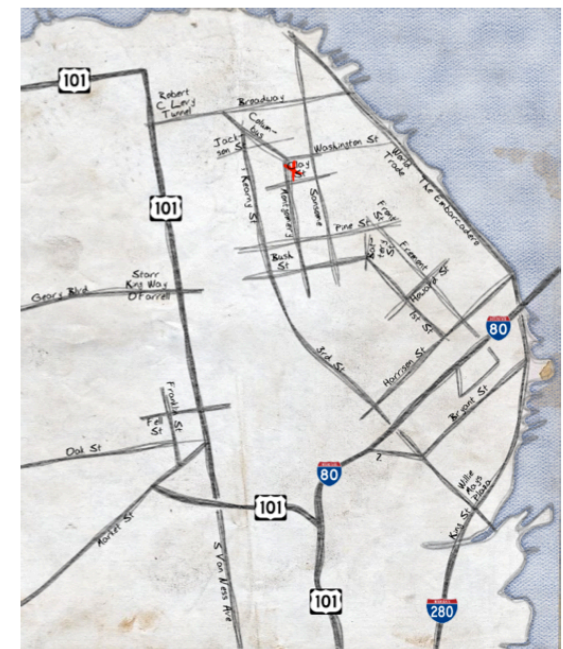
# Detail Removal



General Purpose Online Map



Selected Roads



Our Result

**Automatic Generation of Detail Maps**  
Maneesh Agrawala (UCB EECS), among others



# Detail Removal (in CS10)

- You'll want to write a project to **simulate a real-world situation**, or play a game, or ...
- Abstraction is the idea that you **focus on the essence**, the cleanest way to map the messy real world to one you can build



The London Underground 1928 Map & the 1933 map by Harry Beck.



# Generalization Example

- You have a farm with many animal kinds.
- Different food for each
- You have directions that say
  - To feed dog, put dog food in dog dish
  - To feed chicken, put chicken food in chicken dish
  - To feed rabbit, put rabbit food in rabbit dish
  - Etc...
- How could you do better?
  - To feed <animal>, put <animal> food in <animal> dish



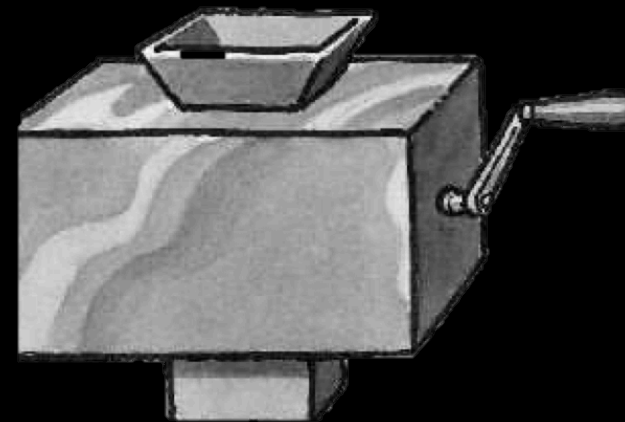
# Generalization (in CS10)

---

- You are going to learn to write functions, like in math class:

$$y = \sin(x)$$

- You should think about **what inputs make sense to use so you don't have to duplicate code**



“Function machine” from *Simply Scheme* (Harvey)



# Summary

- Abstraction is one of the big ideas of computing and computational thinking
- Think about driving. How many of you know how a car works? How many can drive a car? Abstraction!



Someone who died in 1930 could still drive a car today because they've kept the same Abstraction!  
*(right pedal faster, left pedal slow)*

