UC Berkeley
EECS
Lecturer
Pierce Vollucci

The Beauty and Joy of Computing

Lecture #1
Welcome; Abstraction

BJC: YOU'LL LOVE IT!

Watch the student testimonials about the course, what it means to them, and how it has changed their lives. Inspiring!

inst.eecs.berkeley.edu/~cs10/

BJC in one slide

- Big Ideas of Programming
  - Abstraction
  - Algorithms (2)
  - Recursion (2)
  - Functions-as-data, λ (2)
  - *Programming Paradigms*
  - *Concurrency*
  - *Distributed Computing*
- Beauty and Joy
  - "CS Unplugged" activities
  - All lab work in pairs
  - Two 2-week projects in pairs
    - One of their own choice!!
  - One writeup
    - Of students' own choice!!
- Big Ideas of Computing
  - HowStuffWorks
    - 3D Graphics + Video Games
    - Internet
  - Research Summaries
    - AI
    - HCI
  - The Power of Data (big, small, etc)
  - Apps that Changed the World
  - Social Implications of Computing
  - Saving the World with Computing
  - Cloud Computing
  - Limits of Computing
  - Future of Computing

Garcia + Vollucci

Format & Textbooks

- Format (14 hrs/wk * 7 wks)
  - (cue Weekly Schedule)
- Selected Reading
  - Taken from great book ("Blown to Bits" by Abelson, Ledeen & Lewis) + articles + videos
  - Current events EVERY LECTURE (e.g., IBM's Watson vs Jeopardy)
  - Feel free to suggest items!
- All resources FREE
  - Even clickers!
- Pair Programming!

BLOWN TO BITS
contributed articles
Designing Games With A Purpose
IS ABSTRACTION THE KEY TO COMPUTING?
contributed articles
Scratch: Programming for All

…

Garcia + Vollucci

Week at a glance (cue calendar)

- Everything mentioned in the Semester long calendar appears in a week to week form under "Courseware."

Garcia + Vollucci

Pro-student Grading Policies

- EPA
  - Rewards good behavior
  - Effort
    - E.g., Office hours, doing every single lab, hw, reading Discussion pages
  - Participation
    - E.g., Raising hand in lec or discussion, asking questions on Discussions
  - Altruism
    - E.g., helping other students in lab, answering questions on Discussions
- You have 2 "Slip Days"
  - You use them to extend due date, 1 slip day for 1 day extension
  - You can only use one for any given assignment.
  - They follow you around when you pair up (you are counted individually)
    - E.g., A has 2, B has 0. Project is late by 1 day. A uses 1, B is 1 day late
  - Late is 1/3 off for up to one day late. 0 Beyond that

Garcia + Vollucci

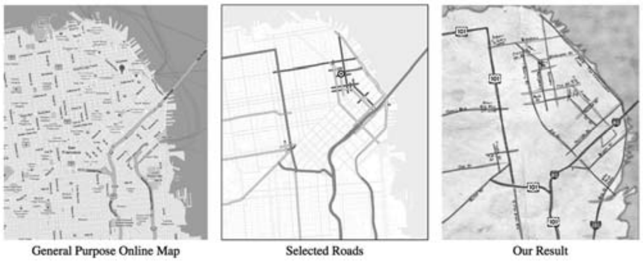
Abstraction

- 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 + Vollucci

Detail Removal



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

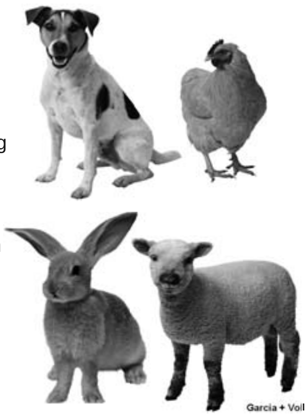
Detail Removal (in BJC)

- 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
- Experts are often brought in to know what to remove and what to keep!

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 BJC)

- 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)

The Power of Abstraction, everywhere!

- Examples:
  - Functions (e.g., sin x)
  - Hiring contractors
  - Application Programming Interfaces (APIs)
  - Technology (e.g., cars)
- Amazing things are built when these layer
  - And the abstraction layers are getting deeper by the day!

We only need to worry about the interface, or specification, or contract NOT how (or by whom) it's built

Above the abstraction line

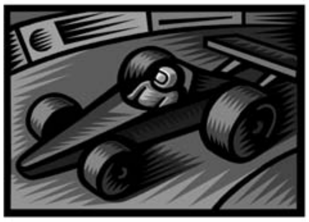
Abstraction Barrier (Interface)
(the interface, or specification, or contract)

Below the abstraction line

This is where / how / when / by whom it is actually built, which is done according to the interface, specification, or contract.

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