Welcome to CS88

David E. Culler

CS8 – Computational Structures in Data Science

http://inst.eecs.berkeley.edu/~cs88

Lecture 1

August 27, 2018

Welcome

- **We are all here to learn:**

```
Knowledge(end) – Knowledge(start)
```


CS88 Team

CS88 Team - uGSIs

Ting Ding
tingding96@berkeley.edu

Jessica Gao
gaojessicaping@berkeley.edu

Alex Kassil
alexkassil@bereley.edu

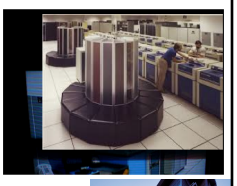
Amir Shahatit
ashahatit@berkeley.edu

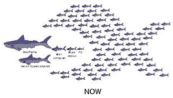
Andrew Tan
andrewtan@berkeley.edu

John Yang
john.yang20@berkeley.edu

CS88 Team - me

- **David Culler (culler@berkeley.edu)**
  - **Hearst Field Annex / 465 Soda Hall (amplab)**
  - **http://www.cs.berkeley.edu/~culler**
  - **Office hours: Mon 3-4 + TBD**
- **Build things**
  - **Cray Time Sharing System**
  - **OS386, OS286**
  - **Active Messages**
  - **Massive High Performance Clusters**
  - **TinyOS / Berkeley Motes, …**
  - **LoCal, BOSS, …**





Goals today

- **Introduce you to**
  - **The field**
  - **The course**
  - **The Team**

- **Answer your questions**
- **Big Ideas**
  - **Algorithm**
  - **Data type**
  - **Representation**

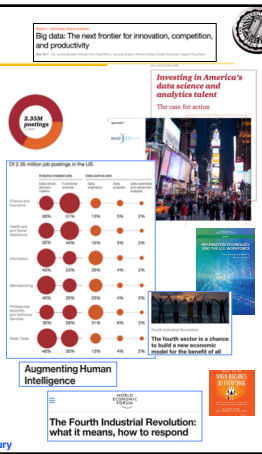
Data Science

Nearly every field of discovery is transitioning from "data poor" to "data rich"

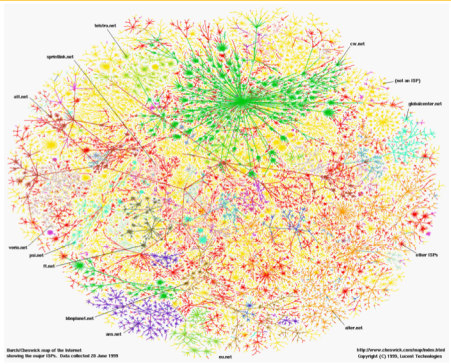
A National Challenge

Increasingly US jobs require data science and analytics skills. Can we meet the demand? The current shortage of skills in the national job pool demonstrates that business-as-usual strategies won't satisfy the growing need. If we are to unlock the promise and potential of data and all the technologies that depend on it, employers and educators will have to transform.

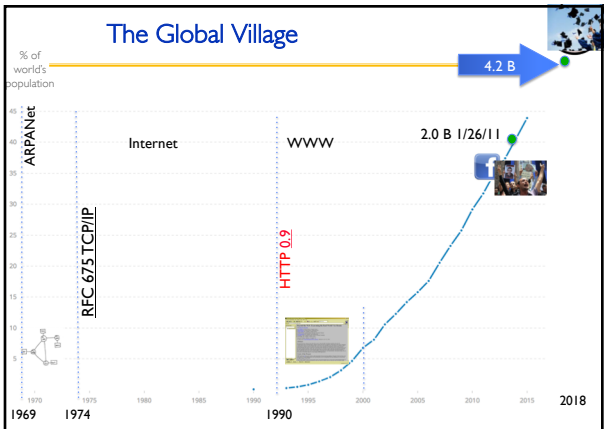
By 2021, **69% of employers expect** candidates with DSA skills to get preference for jobs in their organizations.
Only **23% of college** and university leaders say their graduates will have those skills.



Greatest Artifact of Human Civilization …

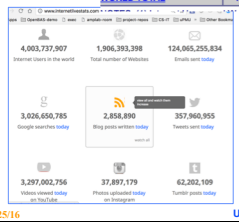


The Global Village

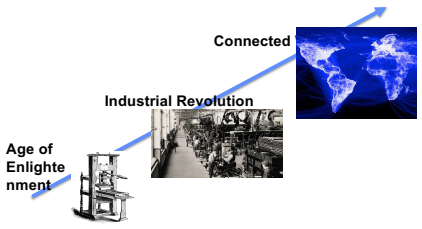


WORLD INTERNET USAGE AND POPULATION STATISTICS DEC 31, 2017 - Update

World Regions	Population (2018 Est.)	Population % of World	Internet Users 31 Dec 2017	Penetration Rate (% Pop.)	Growth 2000-2018
Africa	1,287,914,329	16.9 %	453,329,534	35.2 %	9,941 %
Asia	4,207,588,157	55.1 %	2,023,630,194	48.1 %	1,670 %
Europe	827,650,849	10.8 %	704,833,752	85.2 %	570 %
Latin America / Caribbean	652,047,996	8.5 %	437,001,277	67.0 %	2,318 %
Middle East	254,438,981	3.3 %	164,037,259	64.5 %	4,893 %
North America	363,844,662	4.8 %	345,660,847	95.0 %	219 %
Oceania / Australia	41,273,454	0.6 %	28,439,277	68.9 %	273 %
WORLD TOTAL	7,634,758,428	100.0 %	4,156,932,140	54.4 %	1,052 %



Era of Transformation



A Connected World of Data

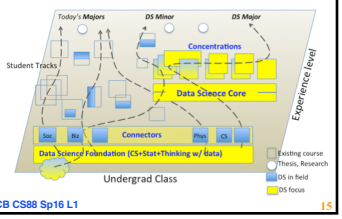
- The world's knowledge at our finger tips
- *Digitialization* of life, industry and society
- Intimately connected to billions of us, globally
- Explosion of observational instruments
  - Genomics, Microscopy, Astronomical, …
- Vast Computational power to do analytics
- Synthetic design exploration thru simulation
- Machine reading of everything
- Statistical machine learning algorithms to "discover" structure

What if I could … ?

- See the world's digital footprints?
- Read everything that's ever been written?
- Take it all in and dive down anywhere as far as the science can take me?
- Learn the physical/chemical/biological /sociological/neurological… models from the data?
- Explore billions of designs and pick the one I want?
- … ?

Data 8 – Foundations of Data Science

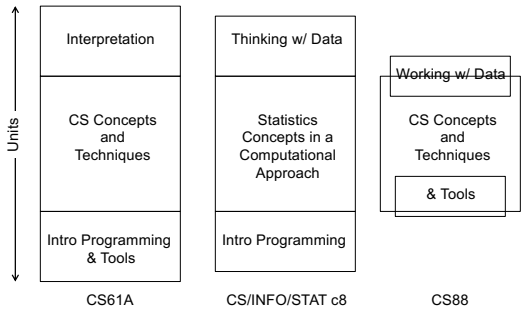
- Computational Thinking + Inferential Thinking in the context of working with real world data
- Introduce you to several computational concepts in a simple data-centered setting
  - Authoring computational documents
  - Tables
  - Within Python3 and "SciPy"



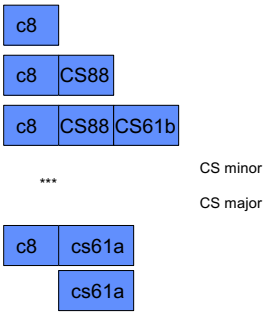
CS88 – Computational Structures in Data Science

- Deeper understanding of the computing concepts introduced in c8
  - Hands-on experience => Foundational Concept
  - How would you create what you use in c8 ?
- Extend your understanding of the structure of computation
  - What is involved in interpreting the code you write ?
  - Deeper CS Concepts: Recursion, Objects, Classes, Higher-order Functions, Declarative programming, …
  - Managing complexity in creating larger software systems through composition
- Create complete (and fun) applications
- In a data-centric approach

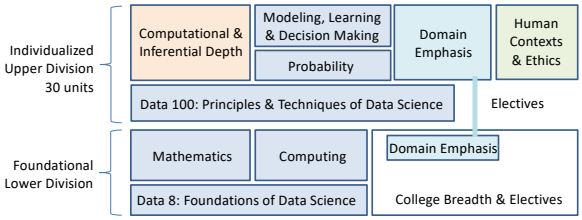
How does CS88 relate to CS61A ?



Opportunities for students

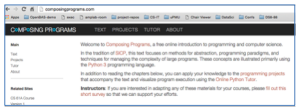


A New Data Science Major soon



Course Structure

- Monday Lecture + Friday Lab/Discussion
- Lecture introduces concepts (quickly)
- Lab provides concrete detail hands-on
- Homework cements your understanding
  - Out Friday, Due Thursday
- Projects (3) put your understanding to work in building complete applications
- Readings: composingprograms.com
  - Same as cs61a



Course Culture

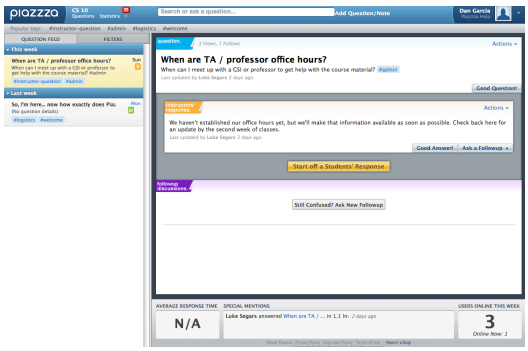
- Learning
- Community
- Respect
- Collaboration
- Peer Instruction

Collaboration

Asking questions is highly encouraged
- Discuss all questions with each other (except exams)
- Submit lab assignments individually (graded on completeness)
  - If you come to lab, you can collaborate liberally
  - If you choose not to come to lab, you must work alone
- Submit homework individually and list collaborators
- Submit projects in pairs; find a partner in your lab

The Limits of collaboration
- Don't share solutions with each other (except project partners)
- Copying solutions will result in failing the course

Piazza for {ask,answer}ing questions



Where will we work?

- datahub.Berkeley.edu
- The computer you carry around
- inst.eecs.Berkeley.edu

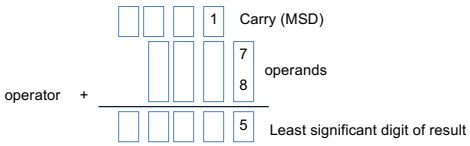
Lab Sections Assignments

- We will collect availability on Wednesday
- Attend any lab section on Friday.
- Assignments effective following Friday.

Algorithm

- An algorithm (pronounced AL-go-rith-um) is a procedure or formula for solving a problem.
- In mathematics and computer science, an algorithm is a self-contained step-by-step set of operations to be performed.
- An algorithm is an effective method that can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function.

Algorithms early in life



Algorithms early in life (in binary)

A Simple Algorithm in Class

- Count the number of students

More interesting one, …

- Betcha people in here share a birthday?

https://en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States_by_date_of_birth

Presidents?

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"*

Experiment

Where are you from?

Possible Answers:

- China
- California
- The Bay Area
- San Mateo
- 1947 Center Street, Berkeley, CA
- 37.8693° N, 122.2696° W

All correct but different levels of abstraction!

Detail Removal (in Data Science)

- You'll want to look at only the interesting data, leave out the details, zoom in/out…
- 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.

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

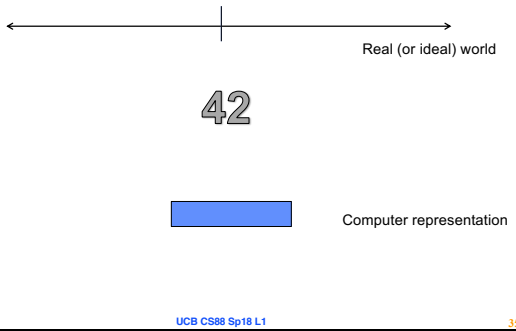
Abstraction Barrier (Interface)
(the interface, or specification, or contract)

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.

Abstraction in CS: Data Type

- What's this?



Data Types and Operations

- **Set of elements**
  - with some internal representation
  - E.g. Integers, Floats, Booleans, Strings, …
- **Set of operations on elements of the type**
  - e.g. +, *, -, /, %, //, **
  - ==, <, >, <=, >=
- **Properties**
  - Commutative, Associative, … , Closure (???)
- **Expressions are valid well-defined sets of operations on elements that produce a value of a type**

Questions

- What's the difference between '==' and '=' ?

Lab and HW this week

- Lab will get you to where you have a *program development environment*
  - Even on your computer
- HW will give practice and explain subtleties of types, operators, and expressions
  - In a program development environment

Question of the week

- How many "things" can you represent with **N** bits