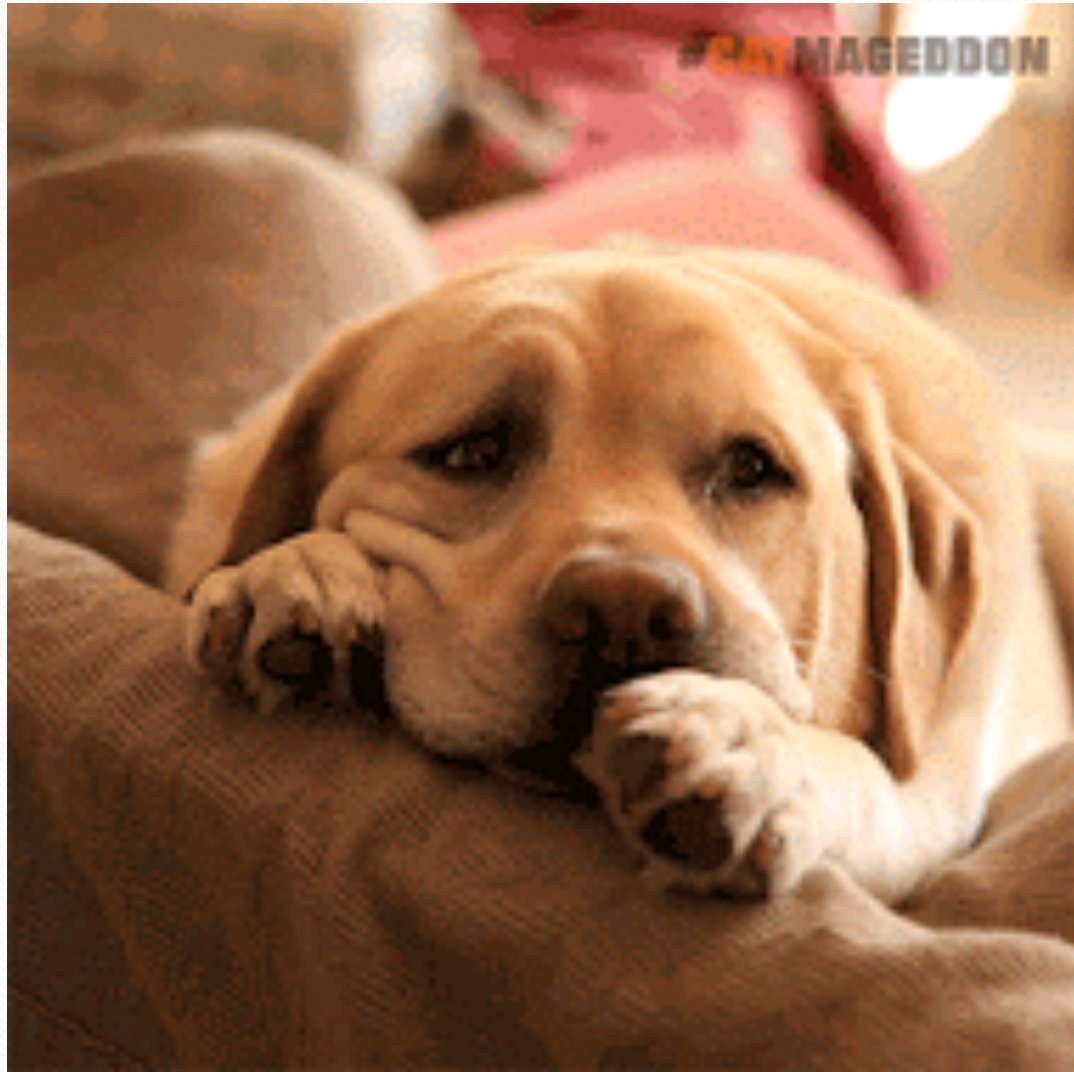






#CATMAGEDDON









CSE 142 vs CSE 143

CSE 142 / AP CS A

- You learned how to write programs and decompose large problems with:
 - Print statements
 - Methods
 - Control Structures
 - loops, if/else
 - File I/O
 - Arrays
 - Objects

CSE 143

- Return of the objects
- You learned to solve more complex tasks efficiently
 - Data structures to organize and model data
 - Algorithms for solving common tasks
 - More advanced language features
- Abstractions are important!

Road Map

CS Concepts

- Client/Implementer
- Efficiency
- Recursion
- Regular Expressions
- Grammars
- Searching / Sorting
- Backtracking
- Hashing
- Huffman Compression

Data Structures

- Lists
- Stacks
- Queues
- Sets
- Maps
- Priority Queues

Java Language

- Exceptions
- Interfaces
- References
- Comparable
- Generics
- Inheritance / Polymorphism
- Abstract Classes

Java Collections

- Arrays
- ArrayList 
- LinkedList 
- Stack
- TreeSet / TreeMap 
- HashSet / HashMap 
- PriorityQueue

Major themes

- Abstraction
 - Leverage existing components without understanding details
 - Create components that can be used as black boxes
- Problem solving
 - Decomposing a large problem into smaller ones
- Design tradeoffs
 - Algorithm analysis - scalability and growth
 - Keeping code easy to read for maintainability
- Recursion
 - Reason about problems in terms of self-similarity
 - Write very short code to achieve complex behaviors

What project?

- Add a GUI to the random sentence generator
- Automate chemistry, physics, calculus problems, etc
 - Maybe even automate writing code with good style?
- Find quotes by keyword in books
- What are you currently doing that a computer could do?
- [List of some project ideas](#)

What language?

- Expanding your Java knowledge with a project is valuable
- Pick a project, see what similar projects use!
 - iOS: [Swift](#)
 - Android: Java, Kotlin
 - Client-side web: [Javascript](#) (many frameworks to choose from)
 - Beautiful visuals: [Processing](#)
 - Data Processing + Machine Learning: [Python](#)
 - Data Management: [SQL](#)
 - Embedded systems: C / C++
- Learn a new programming paradigm
 - Functional languages: [Racket](#), [Haskell](#), [Scala](#), (now, Java 8!)

Leveraging existing code

Here are just a FEW examples. There is so much more!

- Processing language
 - <http://nlp.stanford.edu/software/>
- Building games
 - <http://lwjgl.org/>
 - <http://jbox2d.org/> (with physics!)
- Processing biological data
 - http://biojava.org/wiki/Main_Page
- Accessing Facebook data
 - <http://restfb.com/>
- Making music
 - <http://www.jfugue.org/>

Courses?

- CSE non-majors
 - CSE 154: Web Programming
 - CSE 163: Intermediate Data Programming (Python)
 - CSE 373: Data Structures and Algorithms
 - CSE 374: Programming Concepts and Tools (C/C++, Linux, ...)
 - CSE/STAT 416: Machine learning (requires STAT 311 or 390)
 - CSE 131: Digital Photography
 - CSE 460: Animation Capstone (open to all majors)
 - And more!
- CSE majors
 - CSE 311: (Mathematical) Foundations of Computing
 - CSE 332: Data Abstractions (Data Structures and Algorithms)
 - CSE 331: Software Design and Implementation
 - CSE 341: Programming Languages
 - CSE 344: Intro to Data Management (and databases)
 - CSE 351: Hardware/Software Interface
 - And more!
- INFO, AMATH, HCDE, DXARTS, ...

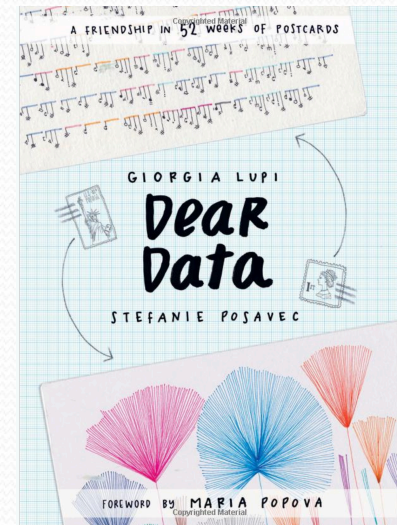
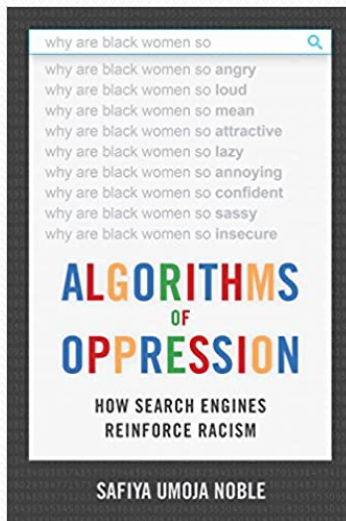
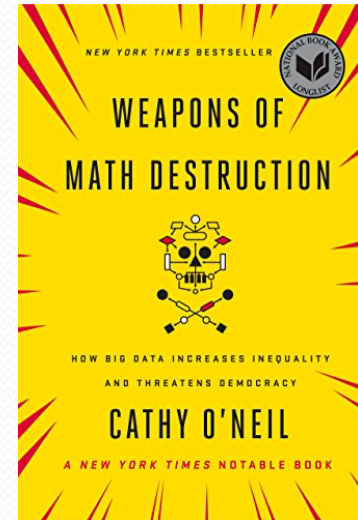
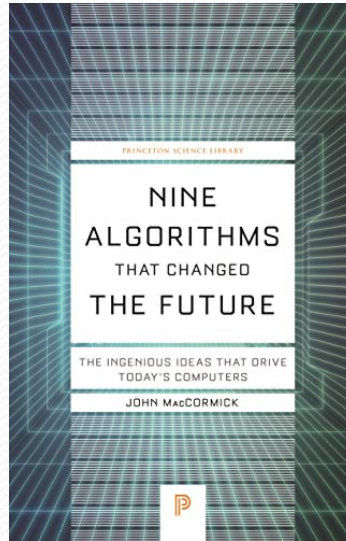
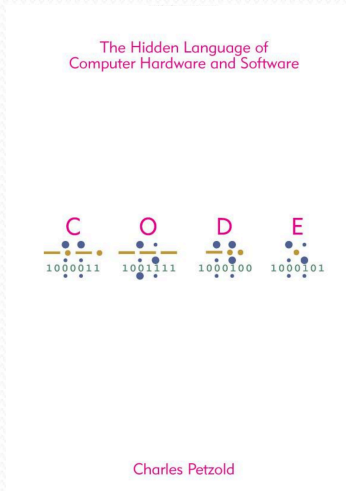
Beyond programming

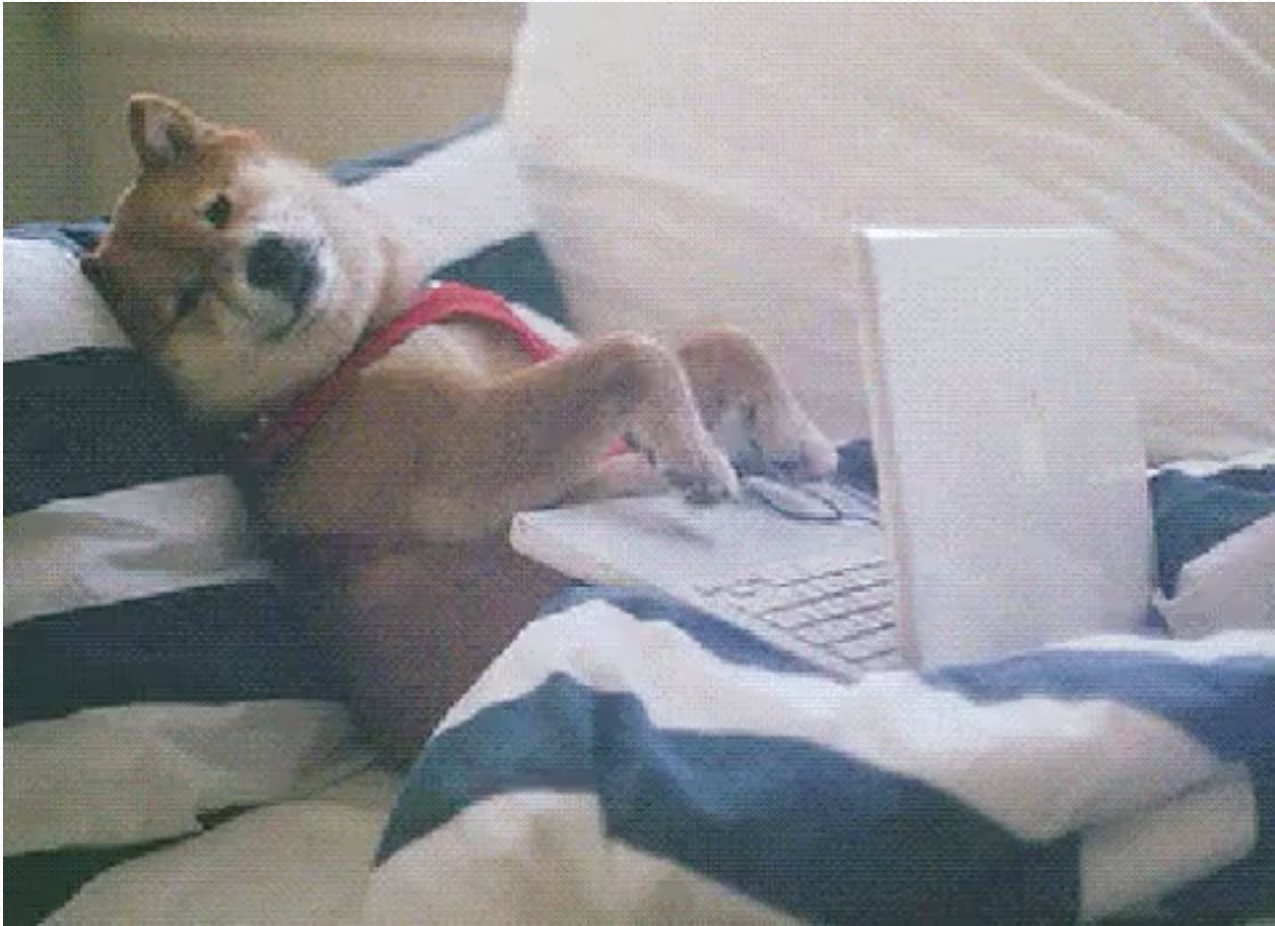
- Investigate how to best distribute relief funds
- Digitize basketball players
- Help deaf/hard-of-hearing people identify sounds
- Detect and prevent toxicity online
- Recognize disinformation online
- Make movies
- Improve digital collaboration
- Design algorithms that are more fair and better respect privacy
- Fix Olympic badminton
- And so much more!

Weekly meetings

- [Change](#) – technologies for low-income regions
- [Dub](#) – human-computer interaction and design
- [ComputingEd@UW](#) – computer science education

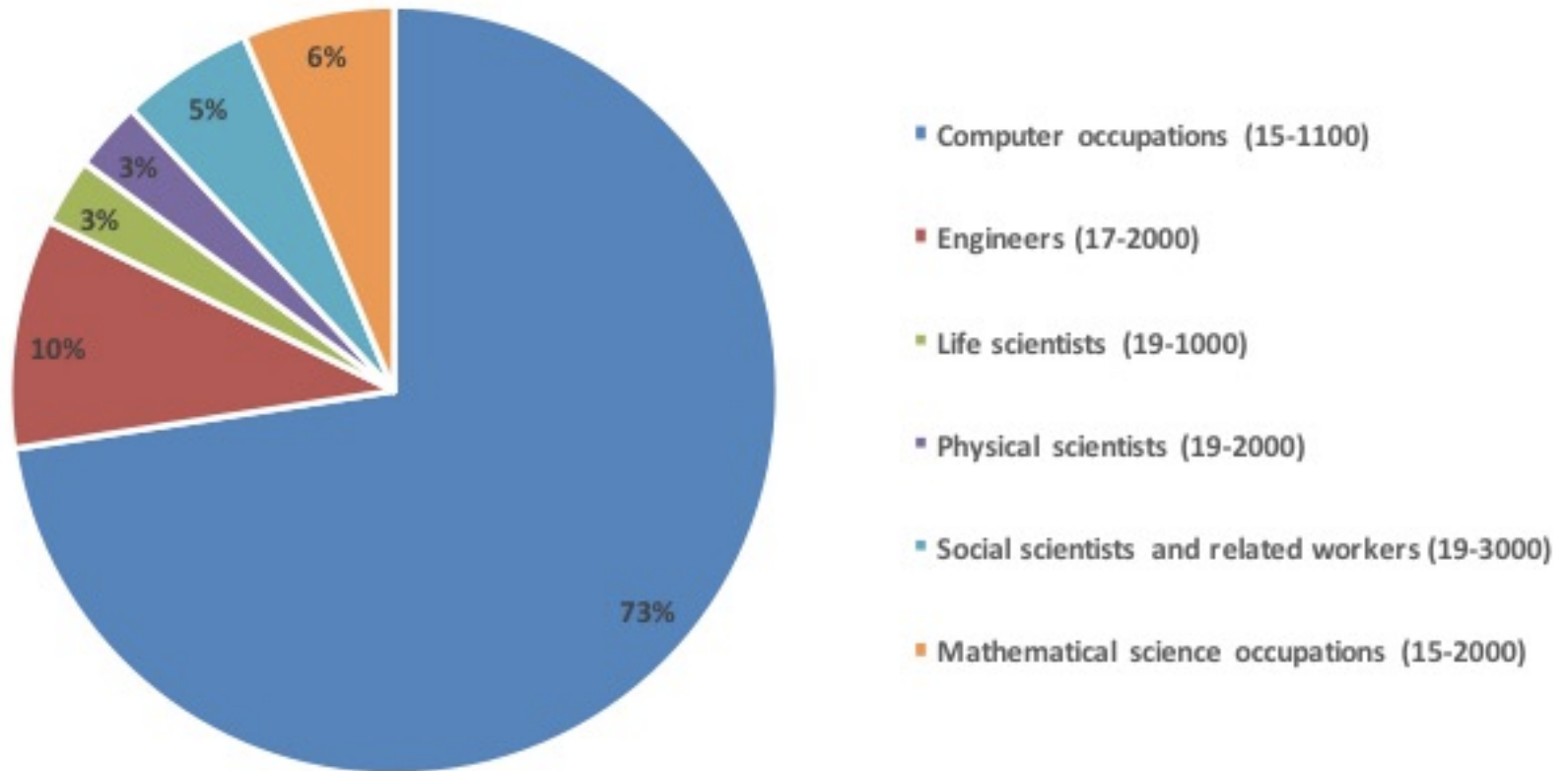
Computer Science Books





Computing & Jobs

Job Growth, 2014-24 - U.S. Bureau of Labor Statistics



Data from the spreadsheet at <http://www.bls.gov/emp/ind-occ-matrix/occupation.xlsx>

Internships

- Various career fairs around campus.
- Start looking early!
- Cast a broad net and interview lots of places. Don't be afraid of getting rejected!
- For those just starting out
 - [Microsoft Explorer Program](#)
 - [Google Engineering Practicum](#)
 - [Code.org suggestions](#)

Roles in Industry

- Software Developer/Software Engineer
 - Builds and designs software
 - Includes designing and engineering architecture of a software system as well as programming
- Product Manager (PM)
 - Designs and makes decisions regarding the overall product
 - Works with people across disciplines at the company
 - Role can be different at different companies
- Test/QA
 - Write and design tests of the product
- Site Reliability Engineer (SRE)
 - Responsible for ensuring that systems and services are available and responsive

Small vs Big Company?

- Small Company
 - Lots of autonomy and impact within the company
 - Often move quickly
 - Breadth – get to work on many projects and with many types of people
- Large company
 - Large data sets, impact many users
 - Lots of support and infrastructure to do your job well
 - Depth – get to focus on specific areas of a project
 - Will you be doing work that helps society broadly?

What Do I Do?

- I'm teaching professor in the Paul G. Allen Center of Computer Science & Engineering. My job is to teach and get you all excited about computing!
- Topics in CS that interest me:
 - Data Science
 - [Machine Learning](#) and [Data Visualization](#)
 - Theoretical Computer Science
 - Approximation and randomized algorithms
 - Theoretical backings of machine learning
 - Computer Science Education
 - Introductory programming and introductory data science
 - Making complex topics in machine learning and data science accessible to more students!
 - Scaling classes to handle increased enrollments

Where Have I worked?

- **Redfin**

- Job: Full-stack engineer (frontend and backend)
- Languages: Java + Javascript

- **Socrata** (Seattle City Data)

- Job: Mostly data science, a little of backend work on search
- Machine Learning: Python
- Search Backend: Scala + ElasticSearch

- **Sift**

- Job: Machine learning infrastructure
- Language: Java + Python
- Libraries: Spark

AMA
(ask me anything)