# CS 142, Spring 2024

**Lecture 37:** closing thoughts


Thank you to Marty Stepp and Stuart Reges for parts of these slides

# Outside of Edmonds

- Online courses
  - Coursera, Udacity, edX
  - CS lectures on YouTube
- Online tutorials
  - Codecademy
  - Khan Academy
  - Stack Overflow
  - W3 Schools
- Practice coding
  - TopCoder
  - Project Euler
  - CodeFights
  - HackerRank
  - CodeStepByStep :)

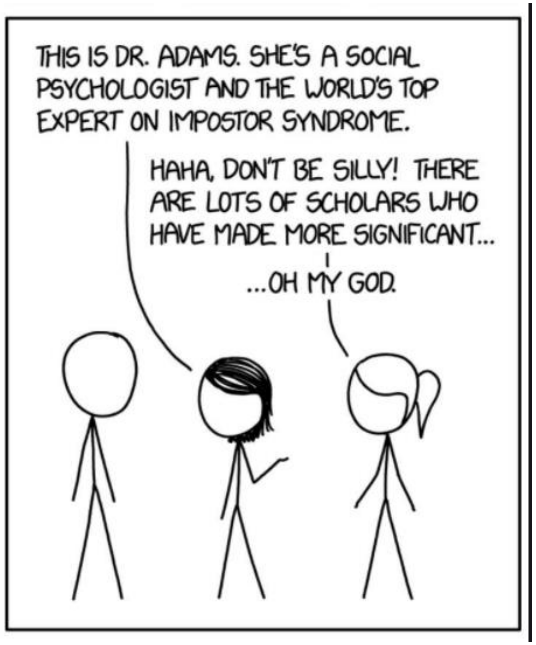



# Internships

- You could get a summer job based on what you learned this year.
  - A few companies hire summer interns with only intro CS skills.
  - A lot of CS interview questions are actually about 142/143-esque topics:
    - references / linked lists, array lists, hashing, complexity, recursion, binary trees, graphs, ...

- Don't assume you are unqualified.
  - Job ads usually list a huge amount of requirements that the company isn't actually expecting.

    You can do it!



# Freshman internships

**Warning:** There are lots of lists of freshman and sophomore internships. Most have many that do not exist anymore due to Covid.

- **Microsoft Explore**
  - https://careers.microsoft.com/v2/global/en/exploremicrosoft
- **Google STEP (Student Training in Engineering Program)**
  - https://buildyourfuture.withgoogle.com/programs/step
- **Google Summer of Code**
  - https://summerofcode.withgoogle.com/
- **Meta University**
  - https://www.metacareers.com/careerprograms/pathways/metauniversity
- **Fermi Lab**
  - https://internships.fnal.gov/community-college-internships-cci/
- **US Department of Energy**
  - https://science.osti.gov/wdts/cci
- **Research Experiences for Undergraduates**
  - https://reufinder.com/programs-for-community-college-students/

# Freshman internships 2

- **Khan Academy internships**
  - https://www.khanacademy.org/careers/interns
- **NSA internships**
  - https://www.intelligencecareers.gov/nsa/students-and-internships
- **Breakout Mentors** (mentor students age 9-14 as they learn programming)
  - http://breakoutmentors.com/
- **Salesforce Futureforce TechLaunchpad**
  - https://info.codepath.org/futureforce-tech-launchpad
- **Green River College Internships List**
  - https://www.devs.greenrivertech.net/internships

# CS Interviews

- Interviews vary company to company, but they usually involve at least some of:
  - Solving coding problems at a whiteboard or on a shared screen
    - No compiler access so you can't guess and check
    - Usually, you can pick any language
    - Syntax and exact built-in method names are not important
  - Describing an algorithm you would use to solve a particular problem
  - Discussing the pros and cons, and complexity (both space and time) of a programming solution

- Overall – they want to see your reasoning ability

# Create a CS Resume or CV

- Resumes used to get Computer Science jobs tend to be in a bit of a different format.

- Make sure to include

  - **projects**: either from a class or written on your own

  - **languages**: divide into "proficient" and "working knowledge of"

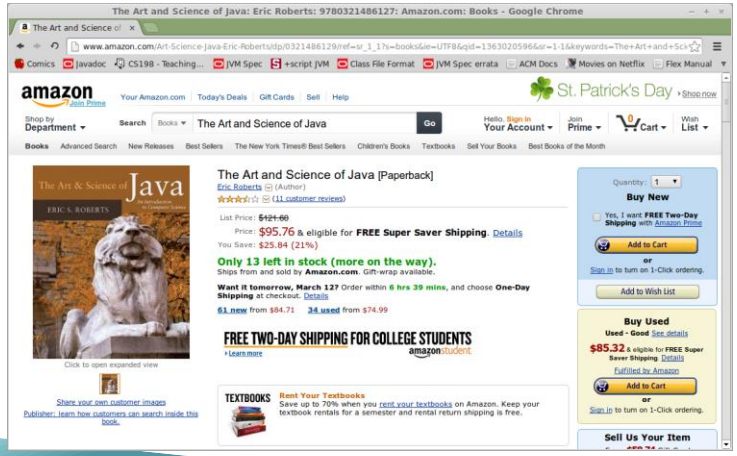
  - technologies and tools: Git, command line, etc.

  - **standard categories**: work experience, education, awards, extra curricular projects / interests and activities

# Create a project of your own

- If there is a CS topic that interests you, go learn more about it!
  - Learn how to make a web site (resume? blog? photos?)
  - Teach yourself a new language and try rewriting 14X HW in it.
  - Grab cool data from reddit /r/datasets and process it
  - Create a tool to help you solve a small annoying problem you have in your everyday life
- Any personal project can look great on your resume if you are excited about it

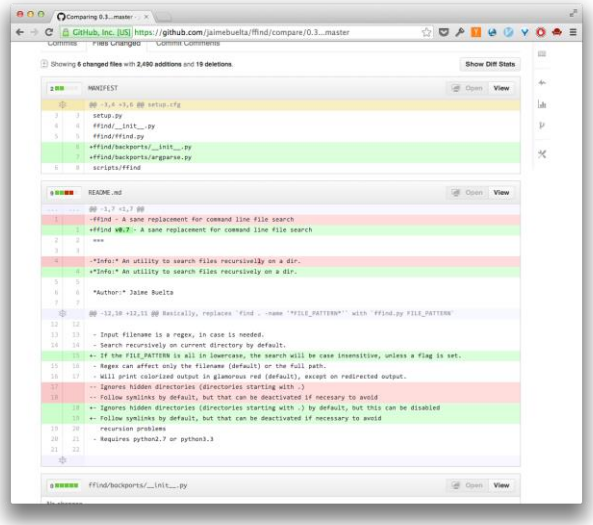
Page content is specified in **HTML**. You can learn its basics in a weekend.




Fonts, colors, and layout are specified in **CSS**. You can learn its basics in a weekend.

# Teach yourself Git

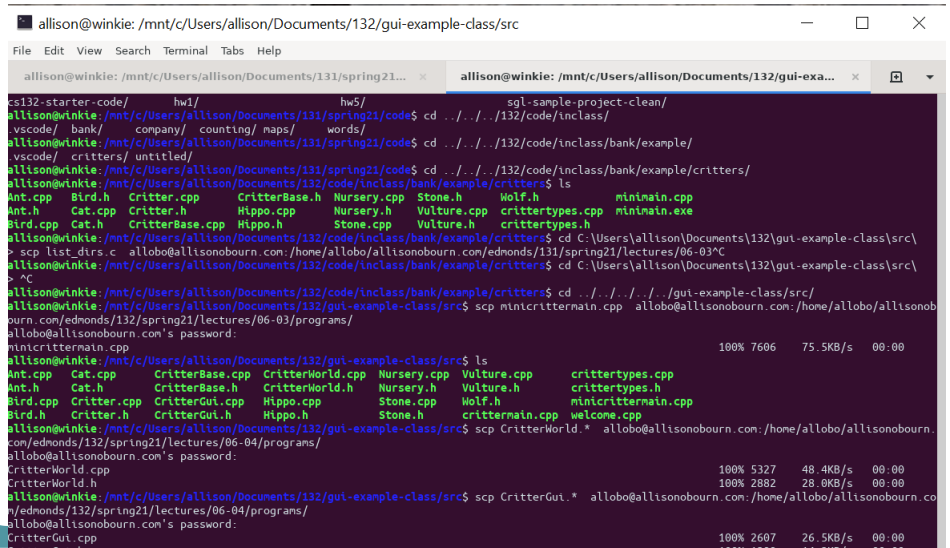
- **Git** is a version control system for managing files in a project.
- **GitHub** is a web site that helps host Git projects.
  - Many companies and courses use Git/GitHub.
  - Git can be tricky, but you can learn its basics in a short time.
  - The Edmonds CS Support Club did a workshop on this earlier this quarter
  - http://guides.github.com/




- Please do not post your 142 project solutions publicly!

# Teach yourself the terminal

- Computer scientists tend to use terminals a lot and companies expect you to know how to get around one
- There are a huge number of commands! No one knows all of them.
- You will likely take a class where you will learn this once you transfer
  - It can be helpful to learn a little on your own earlier as this information comes a little late in some programs
- There are lots of terminals – I recommend bash (a Unix terminal)
  - If you have Windows, install Windows Subsystem for Linux or Cygwin



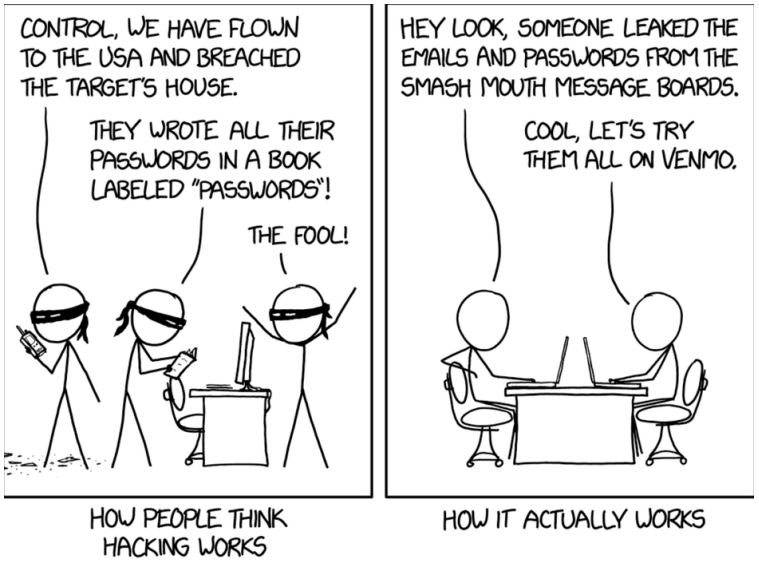
# Teach yourself Linux

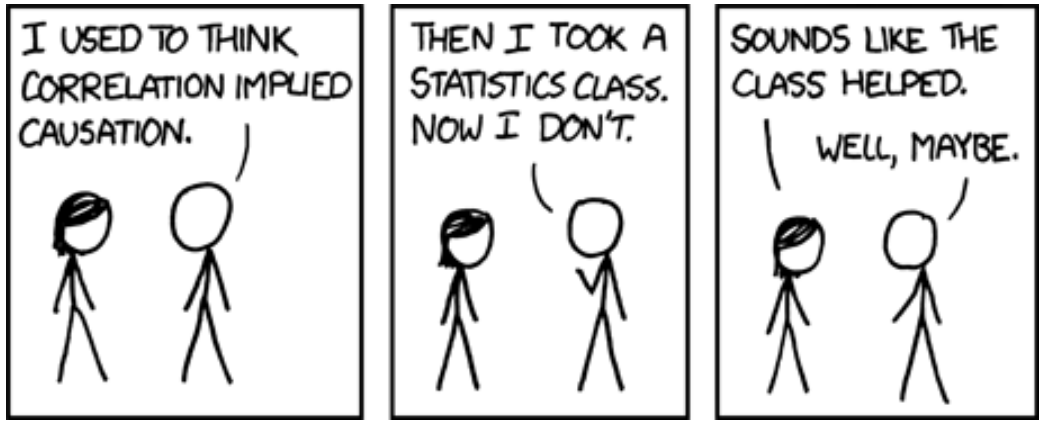
- **Linux** is a free open source operating system for PCs and Macs.
  - You can install it on any computer, or on a **virtual machine** (VM).
  - Many courses and companies use Linux;  knowing it is useful.
  - Get a free VM at: http://virtualbox.org/
  - Get Ubuntu Linux at: http://ubuntu.com/

# Read about CS topics

- **Challenge**: Pick a CS topic or sub-field, and go read about the basics of it over the break.
  - *suggestion:* Pick a topic that aligns with a CS "track."

**Machine Learning**, **Neural Networks**, **Computer Vision**, ...



Thanks for a great quarter!