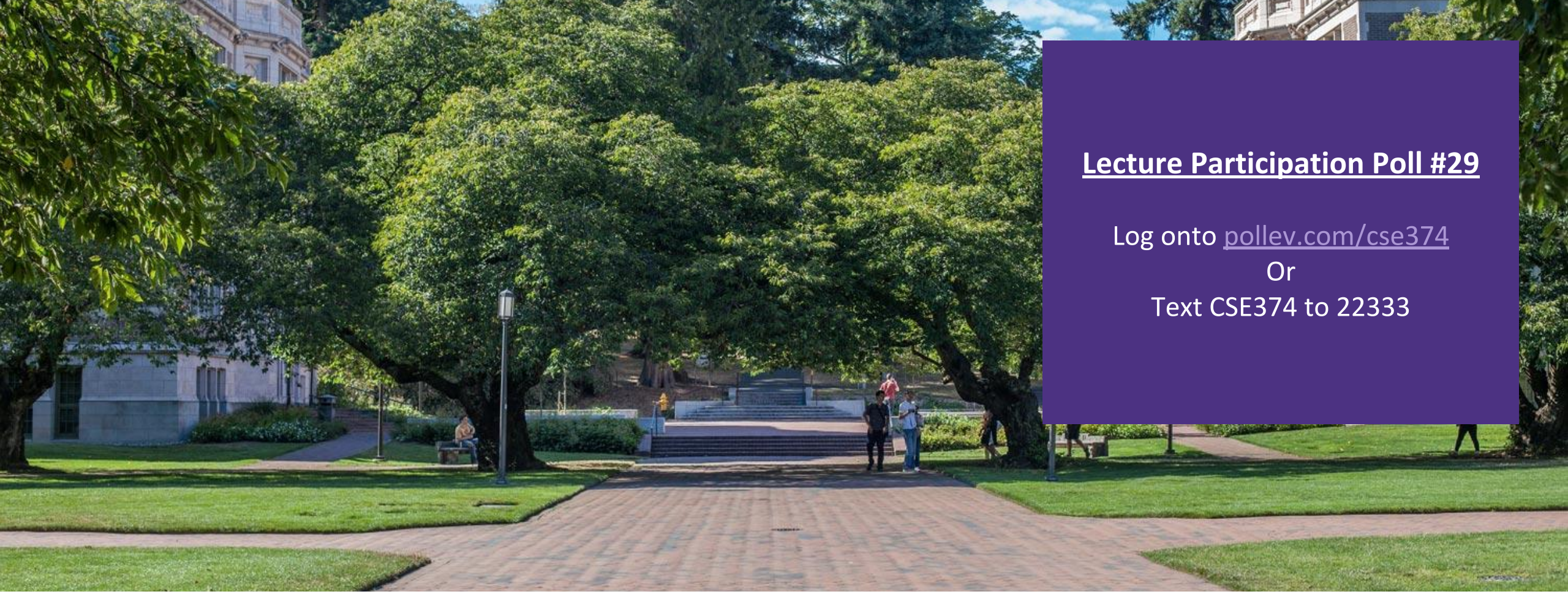


Lecture Participation Poll #29

Log onto pollev.com/cse374
Or
Text CSE374 to 22333

Lecture 29: Careers in tech

CSE 374: Intermediate Programming Concepts and Tools

Administrivia

- HW 5 (final HW) posted
- Final review assignment posted!
- **End of quarter due date Wednesday December 16th @ 9pm**
- **As long as you complete all assignments, you pass the class**

College Career Timeline

Freshman	**Things To Think About** Pay attention to what you like and don't like.	**Actions to Take** Be kind to yourself.
Sophomore	**Things To Think About** How are you incorporating your interests into your extracurricular activities?	**Actions to Take** Take Data Structures. Find a technical experience outside of taking class.
Junior	**Things To Think About** What type of role and what type of company do you want after college?	**Actions to Take** Find a summer internship.
Senior	**Things To Think About** What is life after college going to look like for you? How are you going to transition from the UW into the real world?	**Actions to Take** Get that offer. Negotiate. Live your life.

Pick at least one.

- Volunteer for a cause you care about
- Personal project (outside of class) to solve a problem
- Author a tech blog (or a fashion blog or a cat blog)
- Undergraduate research
- Teaching Assistant/Tutor
- Create a community for your peers.
- Join a consulting club on campus.
- Programming Contests
- Hackathons
- Leadership in student organizations
- Get good grades (if the only thing you are doing is studying, your GPA better be super high > 3.8)

Lies people tell you

Your GPA is the only thing that matters.

Bullshit.

You must have a CS degree to work in tech.

Bullshit.

Everyone but you knows what they're doing.

Bullshit.

Happy Hustling!

Kim Nguyen | @hellokimwin | http://bit.ly/kimLinkedIn

Kasey Champion | @techie4good | http://bit.ly/kaseyLinkedIn

Kim’s Ultimate Resume Guide: http://bit.ly/cseresumeguide

Kim’s College Recruiting Guide for Tech Roles: http://bit.ly/csecareerguide

The typical (tech) recruiting process

Step 1	Express initial interest *Career fairs, events, or via email*
Step 2	First round of the technical interview *Coding challenge, 30 - 60 minute technical screen*
Step 3	Final round of interviews *In person, 3 - 6 interview rounds lasting ~60 minutes each*
Step 4	Decision & offer *Good luck!*

Components of a technical interview

1. Introductions
2. Project discussion
3. Coding exercise
4. Your questions

Introductions

- Develop your pitch
  - Who are you?
  - What are your interests? Goals?
  - Why are you interested in the position?
  - 30 seconds – 1 minute
- Know SOMETHING about the company and why you're interviewing with them

Project Discussion

- Pick 1-2 projects off your resume you can speak in depth about
  - Pick your biggest or most technically interesting project
  - It's ok to talk about school projects
- Don't assume subject domain expertise, but be able to go into detail when asked
- Avoid "we"

Project Discussion

- "I spent this summer working at an advertising network, specifically trying to drive engagement on our video ads by A/B testing new ad content and formats. I worked primarily in the backend and used Python and R for data analysis. I produced a 8% improvement in click-through rates across the board over six weeks of testing."

Follow up questions:

- How long did you work on this project?
- How big was the team working on this, what was your role specifically?
- Why did you choose that technology stack?
- What was the biggest bug you encountered and how did you fix it?
- If you redid the project what would you do differently?

Coding Exercise – Before

Practice Practice Practice

- Treat the interview like a standardized test
- Practice coding without an IDE/Compiler/Computer
- Practice coding and talking aloud at the same time
- Essential Practice Resources:
  - Cracking the Coding Interview
  - LeetCode
  - Data Structures and Algorithms (edX course)

Picking Your Language

- Strongly recommended: Pick something OOP
- Syntax typically doesn't matter
- Review helpful APIs
  - String -> Int
  - String manipulation
  - Popular data structures
  - Searching and Sorting algorithms
- Be able to talk about why you picked that language

Coding Exercise – In the Interview

Question Patterns

- **String or Array manipulation** - Great for tech screens, shorter, sometimes mathy
- **Linked Lists** - Often used in whiteboard interviews because they *expect* you to draw pictures
- **Trees** – BSTs, self balancing. Often used when building up directories or searching for something ie phone trees
- Sorting & Heaps -
- **Hash Tables** - If you are organizing data for lookups… chances are the answer is a hash table

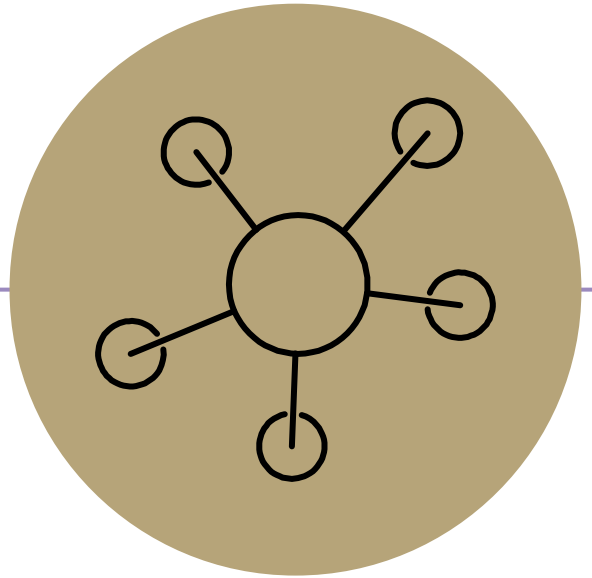
Structure Your Thoughts

- **Talk** - clarify the question
- **Example** - talk through sample input and expected output
- **Brute Force** - what is the simplest way to solve this?
- **Optimize** - can you save run time or memory?
- **Walk Through** - clarify your algorithm
- **Implement** - write the code!
- **Test** - list test cases, does your code address these?

Nervous during your technical coding interview? TEBOW IT!

Your Questions

- Don’t drop the ball!
  - You better have some questions
    - “What is your favorite part about working for X?”
    - “What are some projects you’ve worked on at X?”
    - “Where do you see X in 5 years?”
  - Don’t ask rude questions
    - “Did I pass?”
    - “How much do you make?”
- Show your interest
- Actually look for a good fit



Questions