CS62 Lab0: Setup

Java Fundamentals

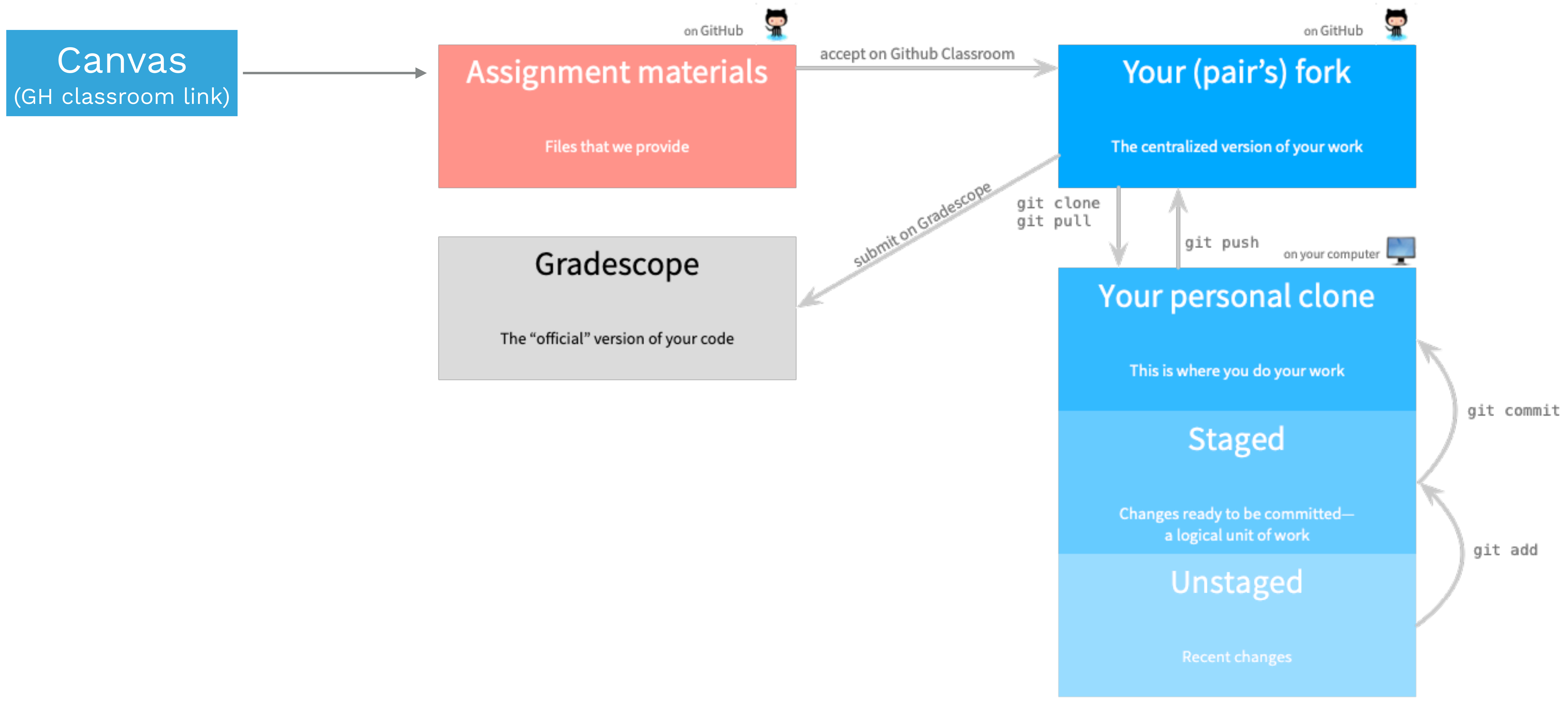


Image stolen from https://hmc-cs-131-spring2020.github.io/howtos/assignments.html

We have mentor hours!

Kellie Au (she/her)

Tues 8-10pm

Adrian Clement (he/him)

Mon 7:30-9:30pm

Asya Lyubavina (she/her)

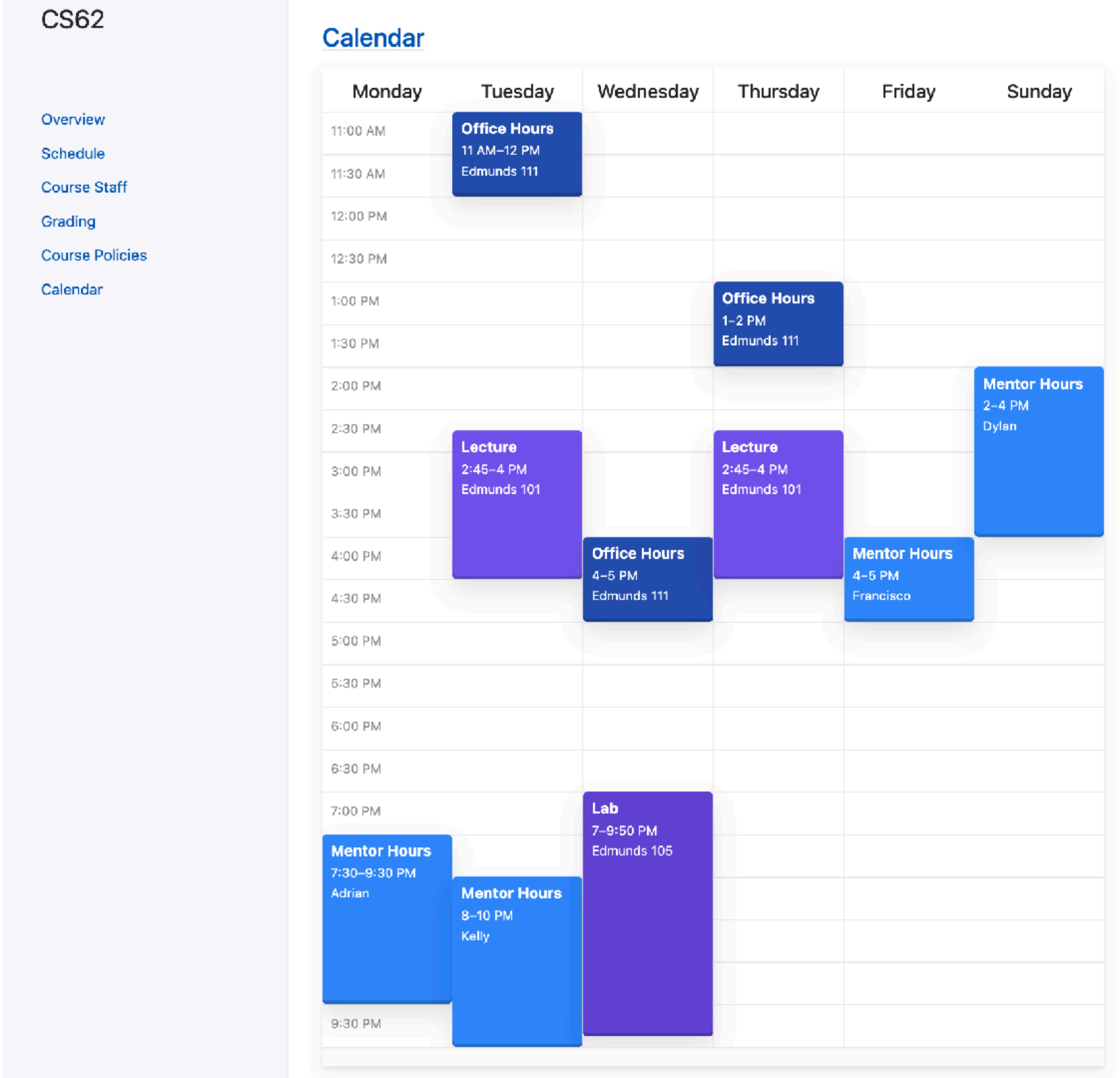
Lab mentor + grading

Dylan O'Connor (he/him)

Sun 2-4pm (+ maybe lab mentor)

Francisco Morales Puente (he/him)

Lab mentor + Fri 4-5pm



CS62

- Overview
- Schedule
- Course Staff
- Grading
- Course Policies
- Calendar

Calendar

	Monday	Tuesday	Wednesday	Thursday	Friday	Sunday
11:00 AM		Office Hours 11 AM–12 PM Edmunds 111				
11:30 AM						
12:00 PM						
12:30 PM						
1:00 PM				Office Hours 1–2 PM Edmunds 111		
1:30 PM						
2:00 PM						Mentor Hours 2–4 PM Dylan
2:30 PM		Lecture 2:45–4 PM Edmunds 101		Lecture 2:45–4 PM Edmunds 101		
3:00 PM						
3:30 PM						
4:00 PM			Office Hours 4–5 PM Edmunds 111		Mentor Hours 4–5 PM Francisco	
4:30 PM						
5:00 PM						
5:30 PM						
6:00 PM						
6:30 PM						
7:00 PM			Lab 7–9:50 PM Edmunds 105			
	Mentor Hours 7:30–9:30 PM Adrian					
		Mentor Hours 8–10 PM Kelly				
9:30 PM						

Just look on the course calendar

Like in CS51P, mentors should announce their hours (or moving them) + location on the Slack.

Lab0 agenda

- 0. Let’s talk about AI usage and come up with a course policy
- 1. Make a Github account (use your .edu email so you can sign up for Github education)
- 2. Go to Canvas to get the Github Classroom invite link for “lab0”
- 3. Follow the instructors on the README.md file (you can view it in your web browser)
- 4. Edit the src/Employee.java code (address the comments) and commit and push to Github (the VSCode UI will do it automatically for you, later in the semester we’ll learn how to do it from the command line)
- 5. Submit assignment on Gradescope by selecting your Github repo
    - 6. We need to approve Gradescope accessing Github before you can submit - ask the prof or a TA to approve the authorization link
- 7. Once your code is on Gradescope, find Prof. Li to get checked off! Then you can go!
- 8. Bonus: Get started on HW1. Or if you forgot to submit the survey, last chance to get points for it.

Class collaboration policy

- Same as CS51P: In general, you should not be looking at anyone else's working code unless you're pair programming. Clarifying concepts with friends/TAs, great. Talking at a high level/writing code on whiteboard, great. The main goal is to help each other learn in a community.
- I acknowledge most cheating happens when circumstances are dire. If you feel like you are at this place, please talk to me - we can always grant extensions/accommodations. The important thing is for you to learn the material.
- Gradescope has an automatic plagiarism checker (+ ChatGPT/Co-pilot/LLM checker) and we manually review each assignment for similarity! This is enforced!

AI usage policy

- In general, students thought this was OK:
  - Creating practice problems
  - Clarifying concepts, maaaybe including generating code
- This is not OK:
  - Using it to solve assignments (note: what does “solve” mean? Just copy/pasting the spec? Using it at all?)
- Other use cases to talk about:
  - A student uses chatGPT to get code snippets explaining concepts they’ve been struggling to understand in lecture. They copy/paste this code for an assignment, but it doesn’t fully work, so they show up to a mentor session to try to get help debugging it.
  - Is it OK to copy/paste code released in lecture into a LLM?
  - A student’s code works for 99% of cases, but fails 1 case and they don’t know why. No more mentor hours are available before the project is due. They want to use ChatGPT to help them debug.

- Form a group of ~4 and make a list of specific DOs/DON’Ts. Youngest one can be spokesperson
- We’ll share out from each group and write a policy together
  - What is allowed, what isn’t allowed
  - What to do when you’re unsure
  - Consequences of violating the policy
- Prof Li’s suggestion: If you use AI for any part of an assignment (including clarification questions), you have to cite it and provide a link to your conversation history in the header comments. If a link isn’t available, you copy/paste your prompt and the response.

Other course norms

- Sample norms (from my elective):
  - It's OK to be wrong! No bad or stupid questions. It's good to ask questions!
  - Acknowledge each other's efforts
  - If you participate a lot in one class, try to wait a few seconds before answering to give other people a chance to answer
- Other course norms:
  - Be nice! Be respectful! Be collaborative
  - Check in if lecture is going too fast
  - We all come from different backgrounds and levels of experience
  - Be open to helping & asking for help

- In your group, discuss:
  - What has been beneficial to your learning in the past? (Particularly in CS classes, but also any class)
  - What has hindered your learning?
  - How does a lecture vs lab environment change the norms?
  - Any norms you want to propose for the course?
  - What about Leetcode style problems? (They are optional :))
- Oldest is the spokesperson

HW1: 3 Beginner Problems

- Learning goals: to practice writing more code in Java, to warm up your problem solving skills. (It’s a hand-holdy assignment!)
- 3 “canonical” CS problems:
  - Fizzbuzz
  - Check if string is palindrome (you did this in Haskell!)
  - Two Sum (your first Leetcode problem! It’s rated “easy”.)
- Note: You probably need tomorrow’s lecture (we’ll talk about Arrays) to fully complete the assignment. If you’re an early starter, I recommend solving the problems in pseudocode and then writing Java after tomorrow’s class.
- Due Tues 1/28 11:59pm on Gradescope (make sure you’re committing your progress on Github intermittently!)