CSE 374: Programming concepts and tools

Spring 2021

Instructor: Megan Hazen

You may use ‘chat’ to ask questions while we wait.

This course is scheduled to run synchronously at your scheduled class time via Zoom. These Zoom class sessions will be recorded. The recording will capture the presenter's audio, video, and computer screen. Student audio and video will be recorded if they share their computer audio and video during the recorded session. The recordings will only be accessible to the students enrolled in the course to review materials. These recordings will not be shared with or accessible to the public.

The University and Zoom have FERPA-compliant agreements in place to protect the security and privacy of UW Zoom accounts. Students who do not wish to be recorded should (1) change their Zoom screen name to hide any personal identifying information such as their name or UW Net ID, and (2) not share their computer audio or video during the recorded Zoom sessions.

What is this course?

CSE 374 is a practical course about

- Command line tools and scripts to automate tasks
- C programming with explicit memory management
- Tools for programming
- Software engineering practice
- Basics of concurrency

374 is also

- An introduction about how to learn what you want to know to move forward.

Who are we?

Your instructor: Dr. Megan Hazen

Your TAs: Nick Durand, Leah Perlmutter, Mohit Sana, Joyce Zhou

Who are you?

~95 Students

What are your disciplines?

Mostly EE, Math, sciences

Mostly Juniors

Today

My job:

- ❑ About this course
- ❑ Zoom Set-up
- ❑ Schedule and homeworks
- ❑ Resources
  - ❑ TAs
  - ❑ Links on homepage
  - ❑ EdStem Discussion
- ❑ Course Rules

Your job:

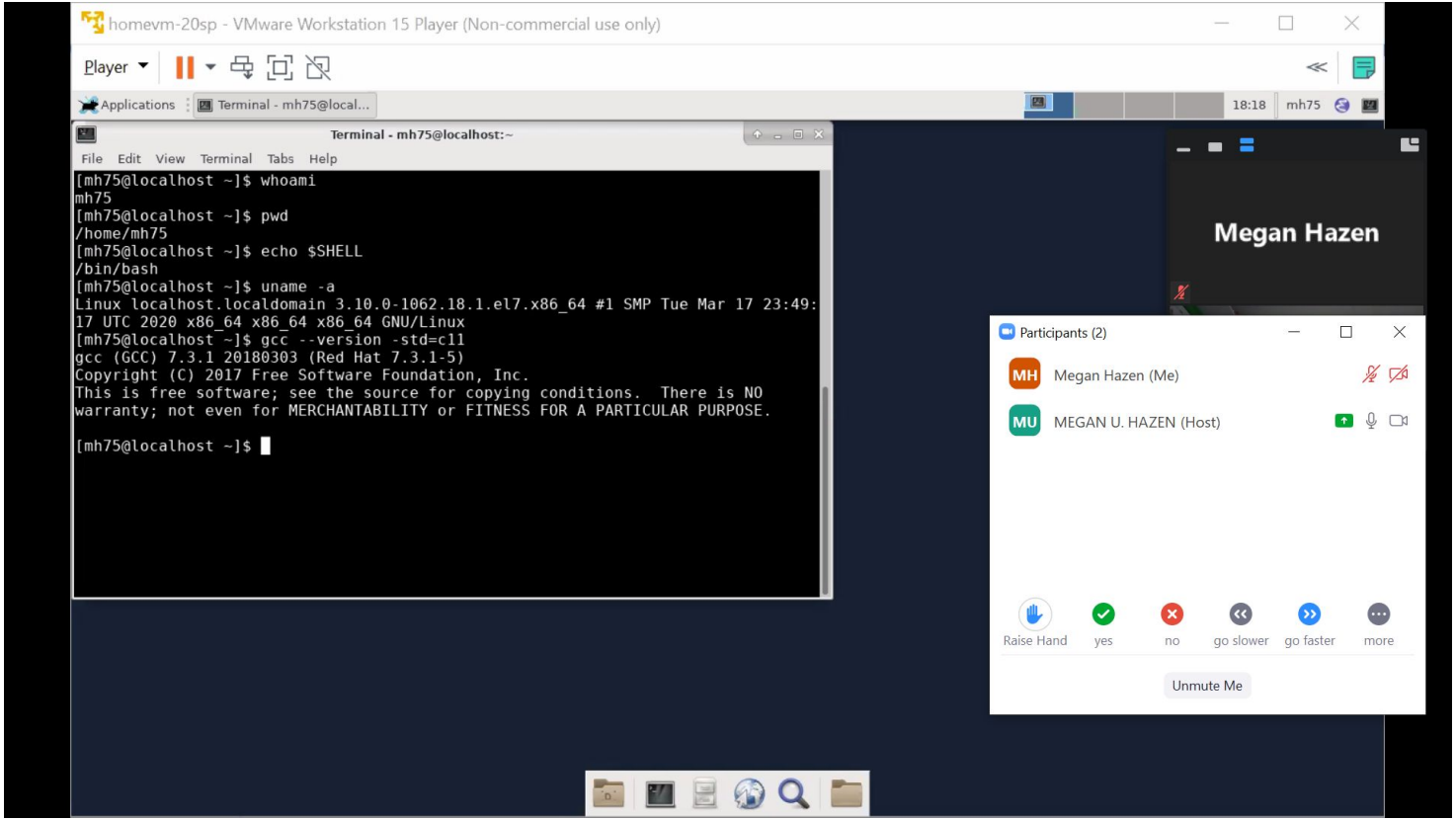
- ❑ Find your resources
- ❑ Work on getting technology sorted out
- ❑ Email CSE374_staff@cs.washington.edu for help

If you are looking at this slide during a live or recorded Zoom lecture...

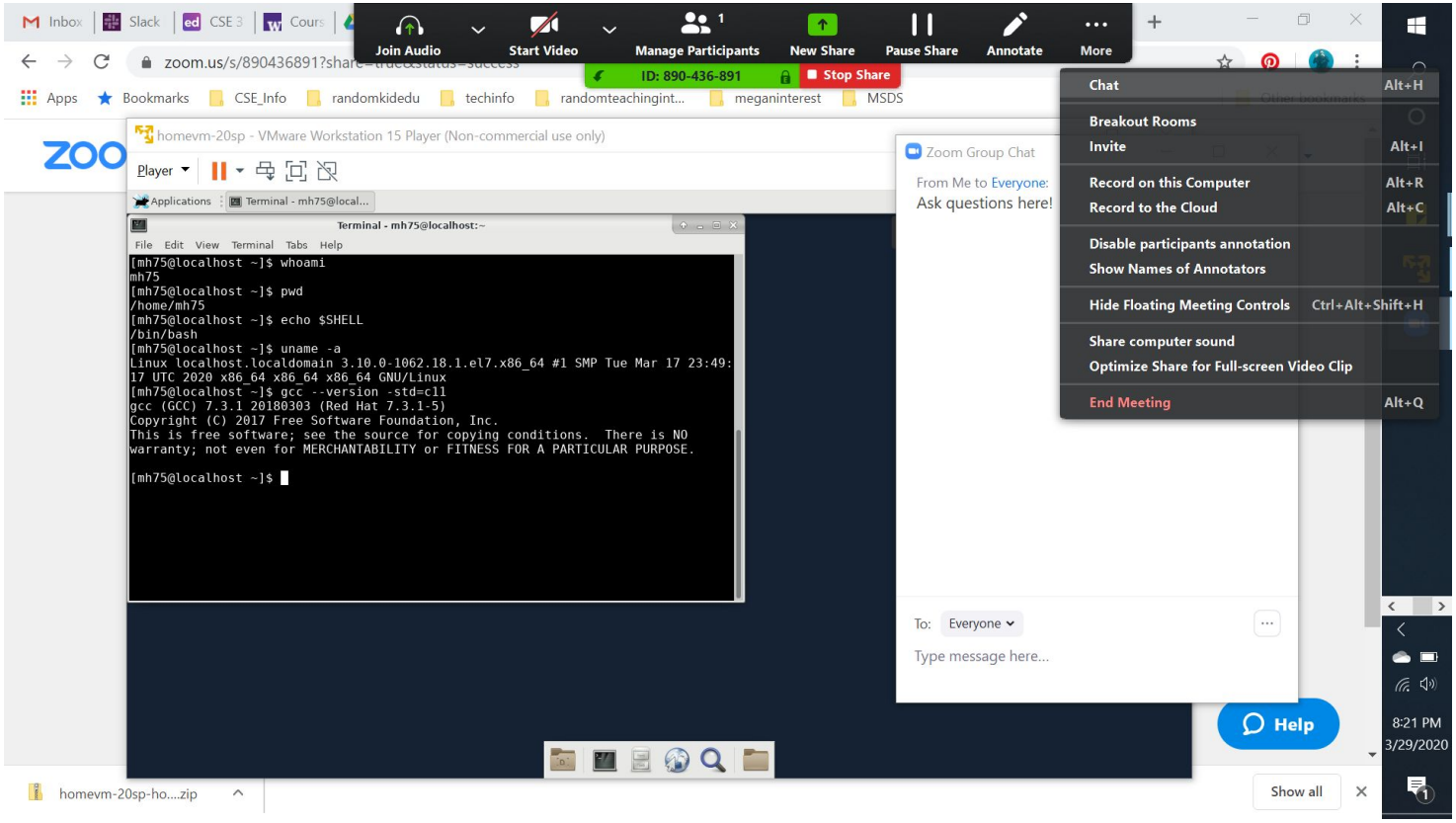
GOOD NEWS

You have figured out how to reach materials via Zoom! If you are viewing this slide as a PDF, please make sure you can use Zoom now.

Still having trouble? Check out the EdStem discussion on remote learning, or email cse374-staff@cs for help.



You will likely be muted by default. Choose ‘Participants’ from your zoom controls, and ‘raise hand’ to alert your professor that you’d like to speak.



Use the chat window to ask questions while the lecture is running.

Office Hours via Zoom

A little different ….

- Expect a ‘waiting room’ to control how many students a TA works with at once
- Expect to share your screen to share code
- These will NOT be recorded

Office Hours Schedule

We are still working on this. There will be some regular times, and also the ability to schedule appointments to avoid long waiting-room line ups.

Regular class sessions will also be extended by 10-20 minutes to accommodate ‘after class’ discussion.

Course requirements

Lecture Monday, Wed., Friday 9:30-10:20

~ 8 homework assignments

- ~ 3 on shells and scripting
- ~3 on C programming
- 1 is a team project

Weekly check-in quizzes

1 final exam

Extra credit: minimal, but exists for more challenge

What to expect

You are responsible for material on Canvas.

Assignments may be more open ended than you are used to - you'll need to figure out whether you are doing the right thing.

Learning how to learn is part of the plan

- Get used to looking at documentation and searching for answers
- Plan to understand, not just re-create

Tinker -expertise comes from experience

Course Resources

Instructor and TAs

- Office hours TBD, but frequently.
- Use office hours to get ‘unstuck’

Edstem Discussion Group

- For each assignment plus more!

Communications: Email cse374-staff@cs

Course website

- https://courses.cs.washington.edu/courses/cse374/21sp

Books, ‘man’ pages, Google

- Use Google as a starting place, be sure you understand
- Recreate on your own; don’t just cut-and-paste

Tinker

- Try things, experiment with new tools

Ask questions early and often!

Lectures

Log in to pay attention - Review for details.

- Plan to learn big picture approaches and concepts
- Jot down key words and ideas to look up later

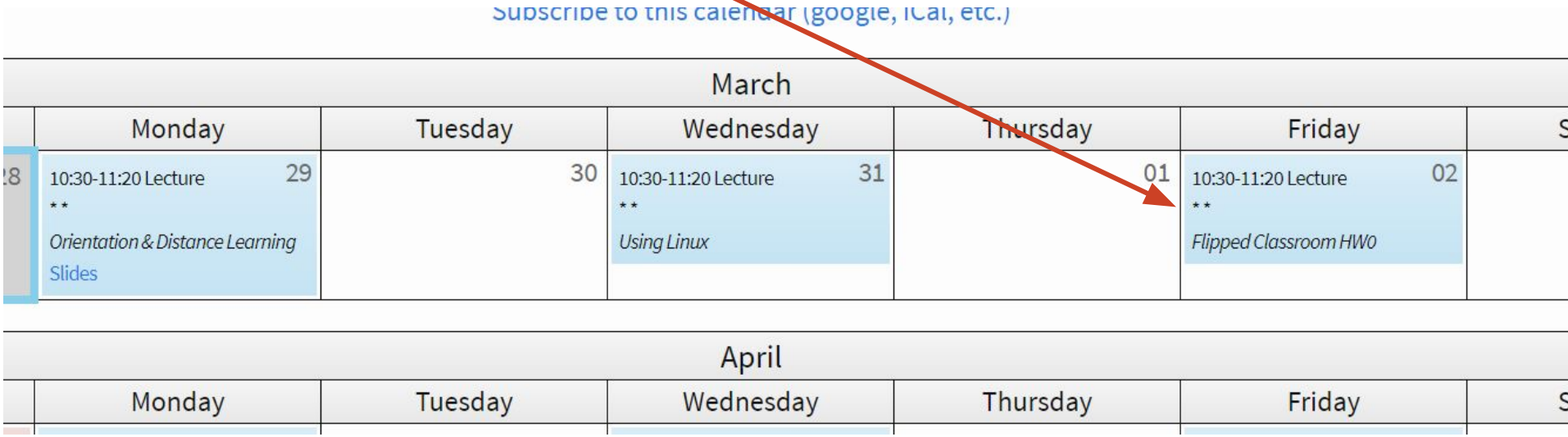
Advice: plan to be an active learner

- review notes, look up documentation, try ideas in the same day
- ask questions early and often

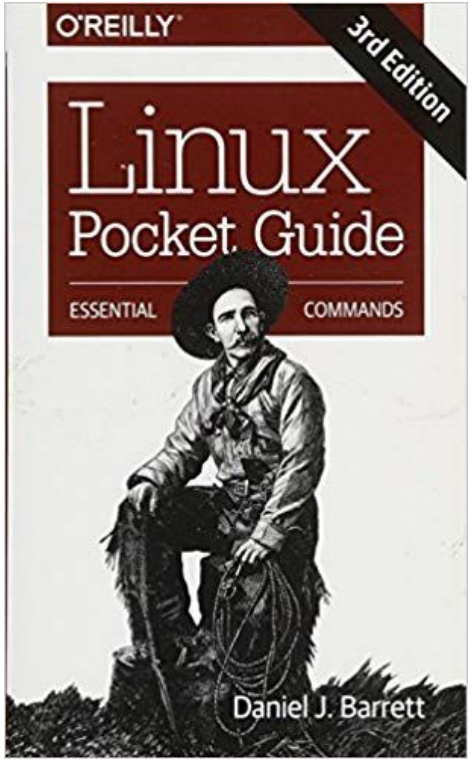
‘Flipped’ classes

During lecture, but will have break-out room time to work on exercises.

Review materials BEFORE class.

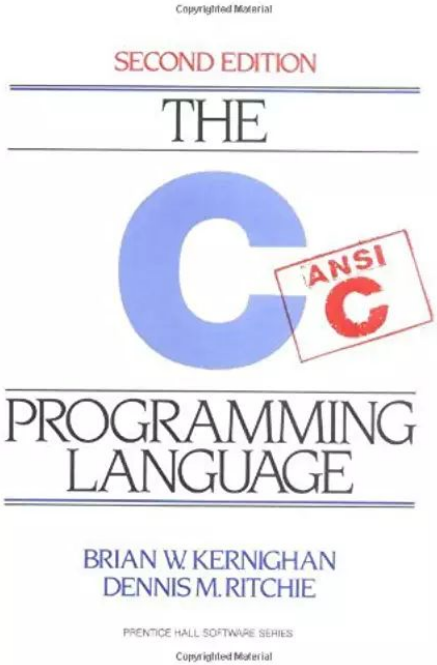


Books



Web searches provide great starting places, and good short reminders

For context and understanding nothing beats a book



Academic Integrity

https://cs.washington.edu/academics/misconduct

Policy on the course webpage

Do your own work, be ready to explain it

Integrity is everything - have high standards

Unless otherwise specified all work in this course is independent

- Do not share code; discuss approach

When in doubt - ask and be honest

Academic Integrity

https://cs.washington.edu/academics/misconduct

Rule 1: You must indicate on your submission any assistance you received. ***Comment in code!***

Rule 2: You must not share actual program code with other students.

Rule 3: You must not look at solution sets or program code from other years, nor should you make your own solutions publicly available even after the due date.

Rule 4: You must be prepared to explain any program code you submit.

Rule 5: Modifying code or other artifacts does not make it your own.

Late Policy

Turn things in on time
Plan ahead

Due Dates are not suggestions, they are strict deadlines.

There will be no late submissions graded for weekly quizzes or the exam.

(Contact instructor for truly exceptional circumstances; before deadline if possible.)

Late Policy

Turn things in on time
Plan ahead

Due Dates are not suggestions, if you fall behind on homework it can be hard to catch up.

20% deduction for each 24 hrs late up to 5 days.

3 24 hr ‘late day’ pass per student

(Contact instructor for truly exceptional circumstances; before deadline if possible.)

Major Ideas of 374

1. Command line and scripting tools
    a. Linux, Bash, automation
2. C programming and memory management
    a. Lower level than Java
3. Tools for programming
    a. Compilers, debuggers
4. Software development and testing
    a. Software specs, tests, and teamwork
5. Concurrency
    a. Using multiple processors at once

Your job

- Explore the syllabus and tools
  - Look forward at the due dates
  - Hint: Canvas is the best starting place
- Answer the survey
- Stay in touch - let us know how its going
- Deep breaths