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:
Adrian Avram, Diana Dai, Luong Dinh, Qingyuan Dong, Maxim Klyuchko, Alex Luo, Leah Perlmutter

Find contact information on the course webpage …
courses.cs.washington.edu/courses/cse374/23sp/
Who are you?

~100 Students

What are your disciplines this quarter?
**Who are you?**

**Concurrent Courses**
When taking the course

- 16% E E-469 (3.48)
- 14% CSE-373 (3.15)
- 11% E E-398 (3.78)
- 10% E E-201
- 9% E E-271 (3.42)

**Declared Majors**
When taking the course

- 40% electrical and comp engr
- 15% electrical engr
- 10% pre science
- 7% mathematics
- 4% pre major (a&s)
Today

My job:
- About this course
- Lecture 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
- Post on Ed for help
If you are looking at this slide during a live or recorded lecture...

GOOD NEWS

You have figured out how to attend or view class. If you are viewing this slide as a PDF, please check on Canvas to make sure you can find lecture recordings or meetings.

Still having trouble? Check out the EdStem discussion, or email cse374-staff@cs for help.
Office Hours Schedule

We are still working on this. There will be some regular times, and also some ability to schedule ad hoc meetings. Office hours will be a mix of remote (via Zoom) and in-person.
Course requirements

Lecture Monday, Wed., Friday
2:30-3:20pm

➢ Practice problems
  ○ Shorter homeworks assignments due frequently
  ○ 1-2 per week

➢ Projects
  ○ More involved homework on major topics
  ○ 1 scripting, 2 C,

What to expect

You are responsible for material on webpage. Follow links for more information.

Assignments may be more open ended than you are used to.

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: Edstem or email cse374-staff@cs

https://courses.cs.washington.edu/courses/cse374/23sp,

Resources list, ‘man’ pages, Google

Use Google as a starting place, be sure you understand

Use formal references for more detail

Recreate on your own; don’t just cut-and-paste

Tinker: Try things, experiment with new tools

Ask questions early and often!
Lectures

Attend for active learning

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
Active classes

If possible, bring your laptop to class. Try things out as we go.

Review materials BEFORE class.

Subscribe to this calendar (google, iCal, etc.)
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.
Submission (Hws)

Most homeworks are submitted via Gradescope, which has an autograder functionality.

The autograder is useful, but not perfect! Use it as a check, not a solver.

When you submit a homework you may resubmit it for a better score with the autograder.

Sometimes there is also a manually graded portion; Each homework will be manually graded once, after the initial due date.
Late Policy (Hws)

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.

Each student gets 10 free ‘late days’. You may use up to 2 ‘late days’ on any assignment; weekends don’t count.

Homeworks turned in early may be resubmitted for a better grade.

May drop one Practice Problem HW (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: Try looking around the course webpage
- Go to EdStem and participate in the first discussion
- Stay in touch - let us know how it's going
- Deep breaths
How are you feeling about this class?

- Excited and Eager
- I'm hoping it's useful
- Nervous and Overwhelmed
- Are we done yet?
- Look! A squirrel!