CS111, Lecture 1
Welcome to CS111!

reading:

Course Syllabus

Honor Code and Collaboration Page

Plan For Today

- Introduction
- CS111 Course Topics
- CS111 Course Policies

Asking Questions

- Feel free to raise your hand at any time with a question
- If you are more comfortable, you can post a question in the Ed forum thread for each day's lecture (optionally anonymously)
- We will monitor the thread throughout the lecture for questions
- Please ask any questions you have - if you have a question, chances are someone else has the same question as well!
- Questions = candy! 

Visit Ed at edstem.org (first time access via Canvas):
https://canvas.stanford.edu/courses/198787/external_tools/28672?display=borderless
Today's Ed Thread:
https://edstem.org/us/courses/66858/discussion/5319804

Course Policies and Modifications

- We are committed to following the syllabus / policies as described here and in the course syllabus, including through long or short-term disruptions.
- There may be extenuating circumstances that necessitate some changes, and in those cases, we will communicate clearly and promptly.

Plan For Today

- **Introduction**
- CS111 Course Topics
- CS111 Course Policies

Teaching Team


Nick Troccoli


Shruti Verma (Head TA)


Andy Liang


Gabe Seir


Jay Chauhan

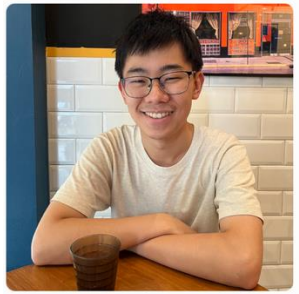

Jessica Chen


Usman Tariq


Cary Xiao


Elena Recaldini


Emma Escandon


Joe Tsai


Keely Podosin


Proud Mpala

About Nick Troccoli (troccoli@stanford.edu):

- Lecturer in CS, taught CS106X, CS107, CS110, CS111
- Stanford BS/MS (coterm) in CS
- Systems track undergrad, AI track grad

Companion Class: CS111ACE

- **CS111ACE** CS111ACE is a 1-unit CR/NC supplementary companion course to CS111. It's designed to provide extra support and problem-solving resources for students, particularly for those who identify as coming from underrepresented or under-resourced backgrounds.
- Entry by application; section Tues 4:30-6:20PM in 320-109
- In addition to all normal CS111 requirements such as section
- see **cs111.stanford.edu** for more details and the link to apply
- Applications are still open! If you apply, please come to the first class on **Tuesday 9/24**. Matthew will go through applications and send out enrollment codes to accepted students by the end of the week. The final deadline for applications is Friday 9/27 at 5pm! (for late enrollment, email Matthew)

Matthew Ayoob
mayoob@stanford.edu

A Wonderful Community

i love pickled pineapples!

I like math. :)

I love playing sudoku so much. I play it in transit or whenever I'm bored, and I have a 100 day streak going on since the start of summer.

I technically have 20 siblings

I learned how to ride a bike at Stanford!

I'm from the Midwest and hoping to meet more Midwestern Cardinals, too!

I hope to be able to meet other people in the class!

I'd like to get into reading Shakespeare or Dostoevsky

*Quotes taken from course welcome survey

Course Website

cs111.stanford.edu

*lecture videos / lecture grades on Canvas

Plan For Today

- Introduction
- **CS111 Course Topics**
- CS111 Course Policies

What is CS111?

CS107 (or equivalent) built up and expanded your breadth and depth of programming experience and techniques and showed you how machines really work.

CS111 leverages this programming experience to introduce operating systems and how they work.

What is an operating system?

What is an Operating System?

An operating system ("OS") is software that allows people to run programs on a computer.

- Examples: iOS, Android, Windows, macOS, Linux

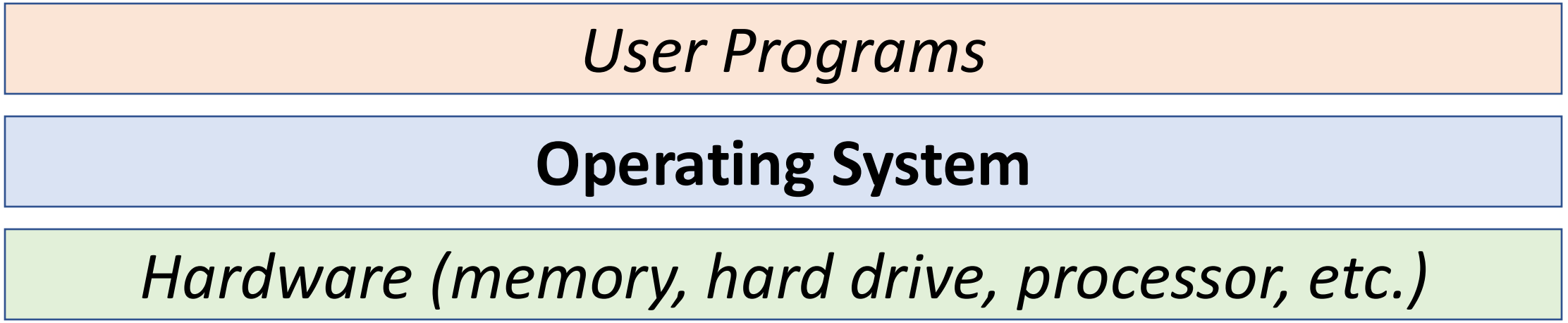
You may think mostly of the *user interface* of the operating system, but an operating system does so much more!

What is an Operating System?

The operating system sits between the hardware and user programs. It manages shared resources and provides functionality for programs to run.

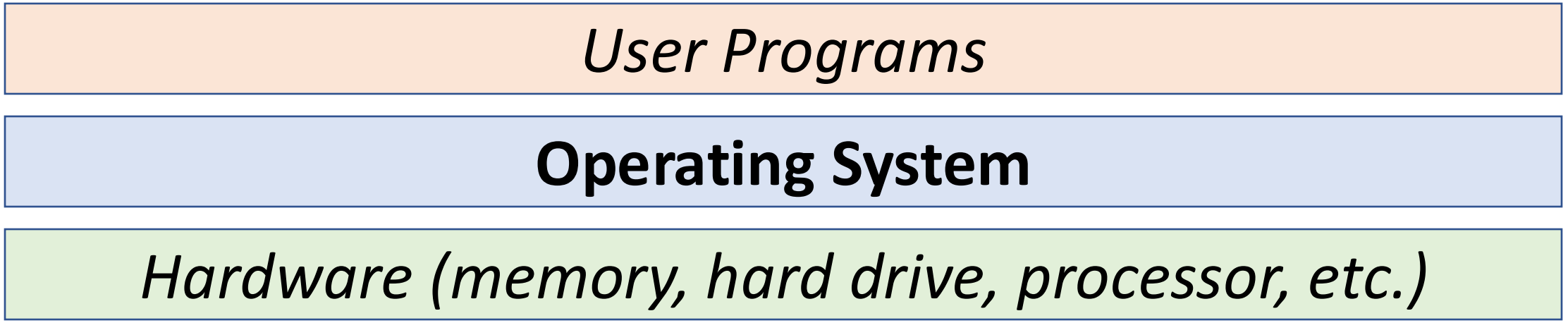
It manages things like:

- Processor (CPU): decides what program gets to do work and for how long
- Memory (RAM): decides what programs get to use what areas of memory
- Hard Drive: decides how the disk is used to store files



Key Operating System Responsibilities

- **Concurrency**: manages different tasks running simultaneously
- **Memory (RAM)**: allows system memory be shared among several tasks
- **Files:** allows many files, for many different users, to share space on disk
- **I/O devices:** allows many devices to operate concurrently and be shared
- **Networks:** allows groups of computers to work together
- **Security:** allows interactions while protecting participants from each other



What is an Operating System?

- So far, when you've written programs, you haven't had to think about any of this. That's the point! The OS is doing its job – it abstracts away complexity from programs.
  - Don't have to coordinate with other programs for who gets to use what memory
  - Don't have to coordinate with other programs for who gets to run when
- OSes work behind the scenes, but are extremely powerful
  - **Example:** devices with 1 CPU core (common through early 2000s) could really only execute 1 program at a time! OSes switch *very* quickly between different tasks to simulate appearance of multitasking.
  - **Example:** how can every program think it can use every address from NULL to 0xfff…? OSes tell programs *fake ("virtual") addresses* and behind the scenes it maps them to the actual ("physical") addresses that it organizes itself.

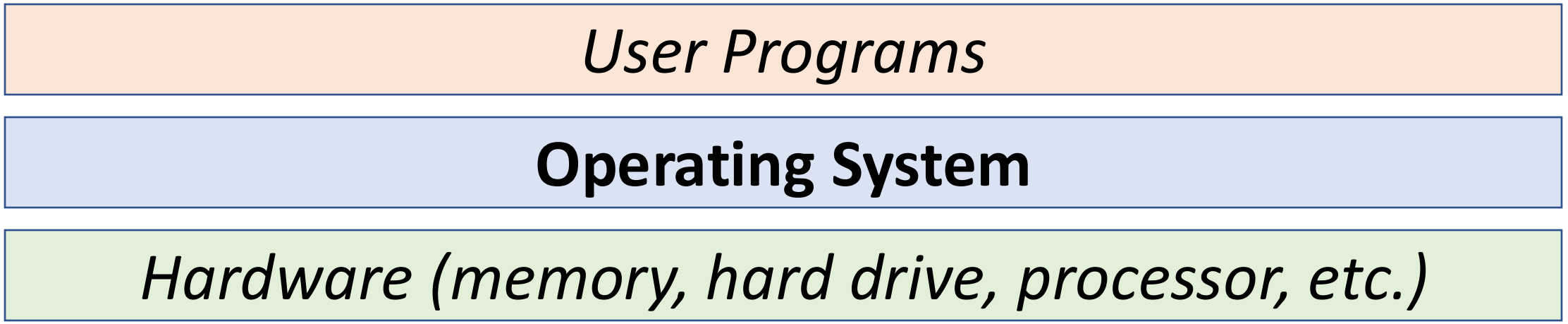
What is an Operating System?

- We can directly leverage operating system functionality in our programs.
- **System calls** are functions the operating system provides that we can call in our code. For example, in Linux:
  - the **open()** function lets us open a file on disk
  - the **fork()** function lets us spawn a new program instance (!)

What is CS111?

In CS111 we are going to explore both “sides” of operating systems:

- We’ll learn what functionality is exported by operating systems to make the programs that we write more powerful.
- We’ll learn how the operating system provides that functionality and how it acts as an **interface** to the computer hardware.



Course Overview

1. **Filesystems** - *How can we design filesystems to manage files on disk, and what are the tradeoffs inherent in designing them? How can we interact with the filesystem in our programs?*
2. **Processes** - *How can our program create and interact with other programs? How does the operating system manage processes?*
3. **Threads** - *How can we have concurrency within a single process? How does the operating system manage concurrency?*
4. **Virtual Memory** - *How can one set of memory be shared among several processes? How can the operating system manage access to a limited amount of system memory?*
5. **Modern Technologies and OSes** - *How do hardware advances impact the design of operating systems?*

Course Overview

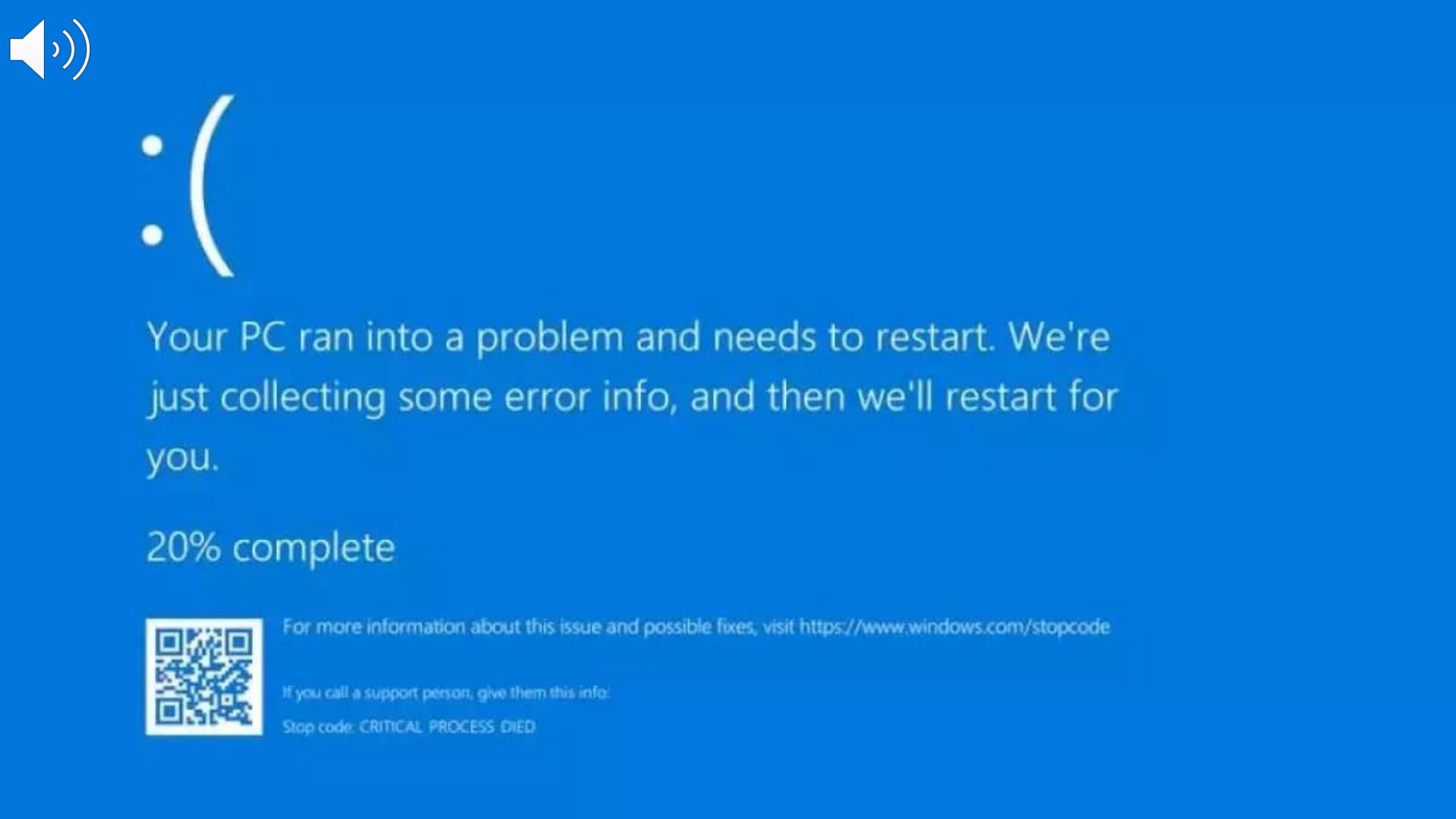
Cross-unit topic: **Ethics and Trust** - *Who/what do we trust, how do we decide, and what do we do when that trust is not upheld?*

- OSes are some of the largest pieces of software – example of large systems, their immense scale/impact, and how much we rely on them (perhaps without realizing it).
- Critical lens to evaluate how we trust technology, and how we choose who/what to trust.

Course Overview

Why is it useful to know about operating systems?

- Understanding computing at this level demystifies how these seemingly-complex systems work and can aid future projects you work on.
- OSes contain many examples of elegant ideas in computing (concurrency, virtualization) that apply well beyond OSes, and pull together ideas like data structures, algorithms, languages, etc.
- We can learn how we can maximally take advantage of the hardware and operating system software available to us in our programs.
- Operating Systems are constantly evolving and encountering new applications (e.g., large datacenters) and new challenges


:(
Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.
20% complete
For more information about this issue and possible fixes, visit https://www.windows.com/stopcode
If you call a support person, give them this info:
Stop code: CRITICAL_PROCESS_DIED

CrowdStrike Outage

July 19, 2024 update to Falcon protection software that caused IT outages worldwide.

- Falcon software runs in a protected part of the Operating System ("kernel"), increasing risk if something goes wrong
- CS111 will help us better understand what happened and why!

Previous slide image source, audio source

Operating Systems – A Brief History

Computers 1 user at a time in person. OSes just shared I/O libraries.

Computers more *shared* (batched jobs, run when user is away) and *concurrent*, OSes had more responsibility.

Timesharing support (multiple *terminals* for multiple users), added complexity.

OSes in personal computers and popping up elsewhere – e.g. small devices and datacenters.

CS111 vs. CS110

- CS111 focuses more specifically on operating systems and how they work
- Topics like filesystems, multiprocessing, and multithreading are similar, but covered in slightly different ways, along with some new assignments
- New topics like Virtual Memory and Dispatching/Scheduling
- We are continuously working to make the class the best it can be and appreciate any feedback!

Plan For Today

- Introduction
- CS111 Course Topics
- **CS111 Course Policies**

Prerequisites

The prerequisite for CS111 is CS107 (or equivalent). You want to have:

- practical C/C++ skills and be able to write programs with complex use of memory and pointers and dynamic memory allocation (malloc/realloc/free/new/delete).
- an understanding of C++ classes, methods, and be able to work with appropriate data structures (arrays, maps, etc.) and standard algorithms (searching, sorting, hashing).
- familiarity with programming in a Unix/Linux environment using tools such as make, gcc/g++, valgrind, gdb, etc. and have a working understanding of the basics of computer architecture (x86-64 from 107, or other)

Course Syllabus and Calendar

cs111.stanford.edu/syllabus

cs111.stanford.edu/calendar

Textbook(s)

- There's no required textbook for the course, but if you're looking for additional reading, we recommend **Operating Systems: Principles and Practice (2nd Edition)** by Thomas Anderson and Michael Dahlin
- C and C++ references will also come in handy throughout the quarter; use the manual pages (**man** on myth)**,** visit cppreference.com or cplusplus.com as needed, etc.

Grading

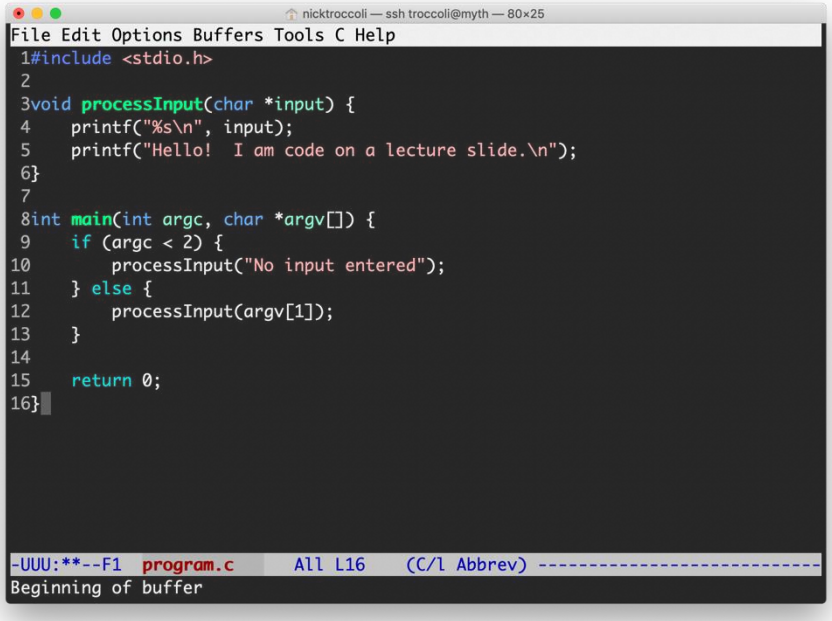
******	52%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	16%	Midterm Exam
***	22%	Final Exam

Grading

******	52%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	16%	Midterm Exam
***	22%	Final Exam

Assignments

- ~7 programming assignments in C/C++ completed individually using **Unix command line tools**
  - Free software, pre-installed on Myth machines / available on course website
  - We will give out starter projects for each assignment
  - assign0 and assign2 are weighted half as much as any other individual assignment
- Graded on **functionality** (behavior) and **style** (elegance)
  - Functionality graded using *automated tools,* given as point score – may also include some TA review
  - Style graded via automated tests and TA code review, given as bucket score



The Style Bucket System

great	An outstanding job; reflects code that is notably clean, elegant and readable, with no issues present.
minor-issues	A good job; reflects code that demonstrates solid effort and is fairly successful at meeting expectations, but also has opportunities for improvement.
major-issue	Has more problems, but shows some effort and understanding. There was either a large concern, or several smaller concerns, in the submission.
multiple-major-issues	Has significant issues, either several large issues or a multitude of smaller ones, that together constitute very poor style work.
0	No work submitted, or barely any changes from the starter assignment.

Assignment Late Policy

- **Start out with 5 "free late days"**: each late day allows you to submit an assignment up to 24 additional hours late without penalty. (No late days permitted for the first or last assignment)
- **Hard deadline 48 hours** after original due date
- Penalty per day after late days are exhausted (1 day: 80% cap; 2 days: 60% cap)
- Late days are "pre-granted extensions" – additional extensions for exceptional circumstances must be approved by the **Head TA** and received **prior to the assignment deadline** or as soon as possible if extenuating circumstances occur later, or extenuating circumstances prevent reaching out prior to the deadline. Please communicate with us! We are here to accommodate you as much as possible.

Question Break!

What questions do you have about the overall course goals, textbook or assignments?

Grading

******	52%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	16%	Midterm Exam
***	22%	Final Exam

Weekly Sections

- Weekly 50-minute in-person sections led by a CA, starting *next* week, offered on Wednesdays, Thursdays and Fridays.
- Hands-on practice in small groups with lecture material and course concepts. Designed to act as prep/preview of homework!
- Graded on attendance + participation
- Section preference submissions open **tomorrow (Tues) at 9AM** and **are not first-come first-serve**. You may submit your preferences anytime until **Thursday 9/26 at 11:59PM PST**. Sign up on the course website.

Grading

******	52%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	16%	Midterm Exam
***	22%	Final Exam

Lecture Recordings

- Because CS111 is not on CGOE (for professional development students) this quarter, the course is not officially recorded.
- However, we have recording equipment in the lecture hall and will do our best to record each lecture ourselves. There may be quality issues and some content may be lost, but we'll do our best!
- See the calendar page (or lecture dropdown) on the course website for slides and lecture code. Materials are posted the evening before each lecture.

Lecture Points

At the same time, staying current with the material is essential to your success this quarter! Our primary goal is to incentivize staying current with lectures to enable you to **start early on assignments** and **have the material you need to work through section material.**

Two ways to get credit for a lecture:

1. Get 100% for the lecture if you attend in person and respond to all Poll Everywhere polls! (regardless of correctness).
2. Watch the recording and complete a corresponding Canvas quiz by **30 min prior to the start of the next lecture**. Quizzes will take about 10-15min each and are graded for correctness but permit 3 total attempts.

Lecture Points

We will also **drop the two lowest lecture scores**, which is intended to cover for lectures where you are both unable to attend in person and also unable to complete the quiz by the next lecture.

Further excused misses are granted by the **Head TA** only in cases where you have already used your 2 excused misses for extenuating circumstances and further extenuating circumstances necessitate additional accommodations.

We'll do a PollEV dry run (doesn't count) in lecture Wednesday, and lecture points will start with lecture on **Fri 9/27** (first real poll Fri., first quiz released after lecture Fri.).

Grading

******	52%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	16%	Midterm Exam
***	22%	Final Exam

Exams

- **Midterm exam** – Wednesday, October 30th, 7-9PM outside of class
  - Contact the course staff by 11:59PM on Friday, October 11th if you have an academic or University conflict with this time, and absolutely cannot make the regularly scheduled midterm. Please include all times 10/29, 10/30 and 10/31 when you can take the exam.
- **Final exam** – Friday, December 13th, 8:30-11:30AM
  - No alternate final - you **MUST** be able to take the final exam at the scheduled time (except for university athletics or OAE accommodations or other last-minute emergencies)

Both exams are in-person closed-book paper exams, with an allowed notes sheet.

Grading

******	52%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	16%	Midterm Exam
***	22%	Final Exam

Read our full course policies document:
https://cs111.stanford.edu/syllabus.html

Question Break!

What questions do you have about section, lecture or exams?

Getting Help

- Post on the **Discussion Forum**
  - Online discussion forum for students; post questions, answer other students' questions
  - Best for course material discussions, course policy questions, short debugging questions or general assignment questions (DON'T POST ASSIGNMENT CODE!)
- Visit **Helper Hours**
  - Sign up in a queue for 1:1 TA help; schedule will be posted / hours start tomorrow.
  - Mix of in-person-only and online-only helper hours
  - Except for assign0 if needed, course staff cannot look at assignment code
  - Best for **group work**, **in-depth code questions (with TAs only!)** or **longer course material discussions**

Course Staff Contact Information

- Email the **Head TA** for requests of a personal nature, such as about: assignment autograder test scores, Office of Accessible Education accommodations, extension requests or other accommodations, enrollment questions, auditing, or other personal matters.
- Email the **instructor** for questions about private/personal matters.
- Email your **section TA** for questions about section attendance grades, or for section accommodations (e.g. missing a section due to extenuating circumstances).
- Email the **grader** listed at the top of your assignment grade report if you have questions about assignment style or manual review grades - for questions about assignment autograder test scores, please email the **Head TA**.
- We are not able to answer course material or assignment questions via email; instead, take advantage of Ed or Helper hours resources!

OAE Accommodations

We are eager to do everything we can to support you and make you successful in CS111! Please email the Head TA as soon as possible with any accommodations you may need for the course. In particular, please let us know of any needed exam accommodations by **Fri 10/11 if possible.**

Course Flexibility / Accommodations

If you are ever sick or encounter an emergency or other exceptional circumstance, we have a variety of accommodation mechanisms, including:

- Assignment late days
- Makeup sections or excused absences
- Lecture excused misses
- Exam accommodations for emergencies/illness
- Ability to attend all helper hours remotely with Head TA permission

If you feel ill or are sick, **please stay home and take care of yourself**. We never want you to feel that you must attend class or helper hours if you are not feeling well. And if you are ill or have another emergency or exceptional circumstance, please reach out to us so that we can help!

Stanford Honor Code

From http://honorcode.stanford.edu (newly updated Honor Code):

The Honor Code is an undertaking of the Stanford academic community, individually and collectively. Its purpose is to uphold a culture of academic honesty.

Students will support this culture of academic honesty by neither giving nor accepting unpermitted academic aid in any work that serves as a component of grading or evaluation, including assignments, examinations, and research.

Instructors will support this culture of academic honesty by providing clear guidance, both in their course syllabi and in response to student questions, on what constitutes permitted and unpermitted aid. Instructors will also not take unusual or unreasonable precautions to prevent academic dishonesty.

Students and instructors will also cultivate an environment conducive to academic integrity. While instructors alone set academic requirements, the Honor Code is a community undertaking that requires students and instructors to work together to ensure conditions that support academic integrity.

Honor Code and CS111

It is your responsibility to ensure you have read and are familiar with the honor code guidelines posted on the main page of the CS111 course website. Please read them and come talk to us if you have any questions or concerns.

https://cs111.stanford.edu/collaboration

Please help us ensure academic integrity:

- Indicate any assistance received on HW (books, friends, etc.).
- Do not look at other people's solution code or answers
- Do not give your solutions to others or post them publicly on the web or our Ed forum.
- Tutoring is not appropriate for help with work that will be submitted for a grade.
- Do not use AI tools to write code/responses for you on assignments or any graded work.

Honor Code and CS111

https://cs111.stanford.edu/collaboration

- Assignments are checked for similarity with help of robust software tools and processes. Concerns are reported to the Office of Community Standards.
- Any cases determined by the OCS process to be Honor Code violations will result in zero credit for the work of concern plus a course grade penalty of at least a one grade bucket decrease (e.g. B to B-) up to failing the course.
- If you need help, please contact us and we will help you.
  - We do not want you to feel any pressure to violate the Honor Code in order to succeed in this course.
  - We also have a late citation / retraction policy that permits a late citation or retracting all or part of previously-submitted assignment work at any time no questions asked, up to the start of the final exam.

Use of AI Tools

- AI tools can be extremely valuable in certain contexts and enable easier development / coding.
- However, for CS111 our focus is on not just the artifact but also the process: developing skills to write code, debug code, and think critically about existing code and code you write, all of which will make you a more powerful computer scientist and let you work more effectively and efficiently!
- For these reasons, you should only use AI tools in the same way that you would ask a friend in the class for help – high level questions, citations where needed, etc. You **should not use AI tools to write code/responses for you on assignments or any graded work**. Doing so is a violation of the Stanford Honor Code.

Assign0

Assignment 0 is posted on the course website and is due next week on **Mon. 9/30 at 11:59PM PDT**. **No late submissions accepted** except for OAE/Head TA accommodations.

Introductory assignment to help get up to speed with course logistics, set up myth machine access, and get going with the terminal, C, and C++. It does not require any lecture material.

Check out our C/C++ guide on the course website, as well as review videos on Canvas about pointers/memory and C++ classes!

Getting started guide for working on assignments: cs111.stanford.edu/getting-started.html

Topic 1: Filesystems - How can we design filesystems to manage files on disk, and what are the tradeoffs inherent in designing them? How can we interact with the filesystem in our programs?

Recap

- CS111 is a class in C/C++ that teaches you about operating systems and how they work.
- Please visit the course website, cs111.stanford.edu, where you can read the General Information page, information about the Honor Code in CS111, and more about CS111 course policies and logistics.
- Check out assign0 on the course website for getting set up to work on assignments for the quarter.
- Our first topic is filesystems – understanding how we can store files on disk.

We're looking forward to an awesome quarter!

Next time: more about filesystems