CS111, Lecture 1
Welcome to CS111!

reading:

Course Syllabus

Honor Code and Collaboration Page

😷 masks strongly recommended

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

Visit Ed (or access via Canvas):
https://edstem.org/us/courses/44415/discussion

Today's thread:
https://edstem.org/us/courses/44415/discussion/3508012

Guiding Principles For In-Person Class

- We are each starting the quarter in unique circumstances and may be facing a variety of uncertainties, responsibilities, and emotions.
- We will do everything we can to support you. We have designed the course to the best of our ability to provide flexibility.
- We will constantly evaluate and listen to ensure the class is going as smoothly as possible for everyone.
- Please communicate with us if any personal circumstances or issues arise! We are here to support you.

Guiding Principles For In-Person Class

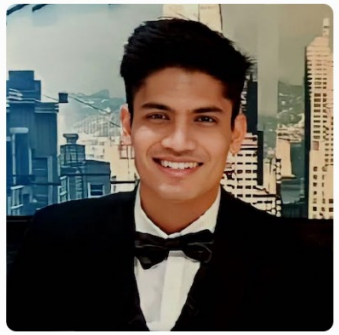
- Stanford University is currently strongly recommending the use of masks in classrooms and instructional spaces. We strongly encourage you to wear a mask in lecture, section and helper hours.
- Some of us have health conditions precluding our ability to wear masks. Students in this situation should work with the **Office of Accessible Education**.
- Some of us might feel more comfortable wearing masks/social distancing even when not required. All of our preferences are reasonable, and it is important that we treat each others' preferences with respect and care.

Plan For Today

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

Teaching Team


Nick Troccoli

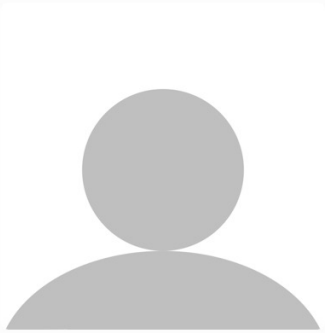

Aman Kansal

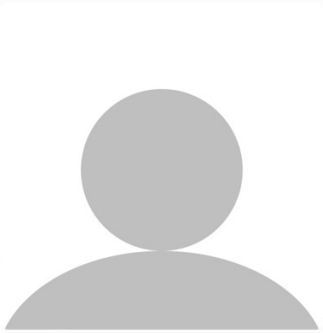

Diego Ahmad-Stein


Michael Cao

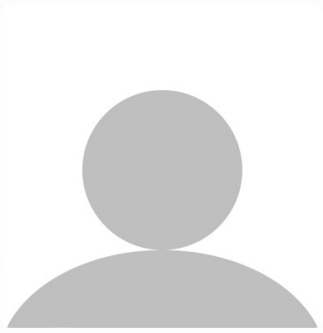

Michela Marchini


Michelina Hanlon


Eric Tang


Evan Saracay

Isaac Gorelik


Nirvik Baruah


Yash Govil

CS111 Embedded Ethics:


Dr. Benjamin Xie

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** (“CS111A”) is an extra 1-unit “Pathfinders” or “ACE” section for undergraduates with additional course support, practice and instruction.
- The ACE program seeks to provide strong supplemental support in technical classes, particularly for students from under resourced and/or minoritized backgrounds.
- Entry by application; section Tues 9:30-11:20AM in 50-51A
- see **cs111.stanford.edu** for more details and the link to apply
- Applications are still open! If you apply, please come to our first class on **Tuesday 10/3** at 9:30AM in 50-51A. I will go through applications and send out access codes to accepted students by the end of week 2 / early week 3. The final deadline for applications is Friday 9/29 at 5pm!

Trip Master

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

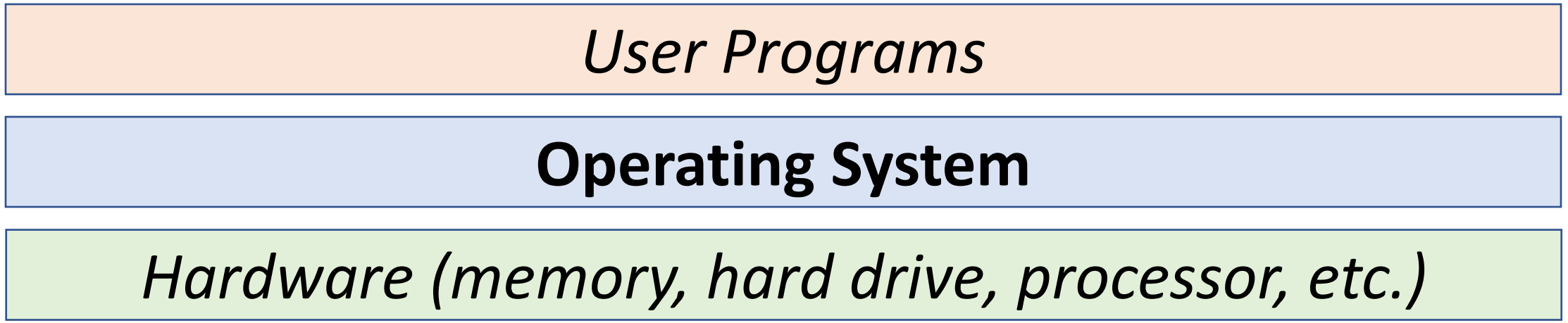
You may think mostly of the *user interface* of the operating system, but an operating system does so much more!

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.

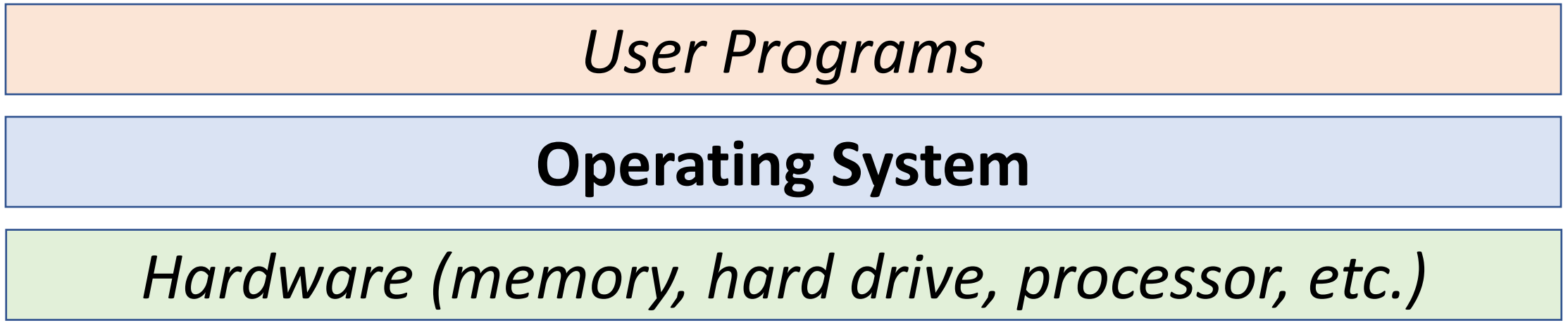
It manages things like:

- 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.

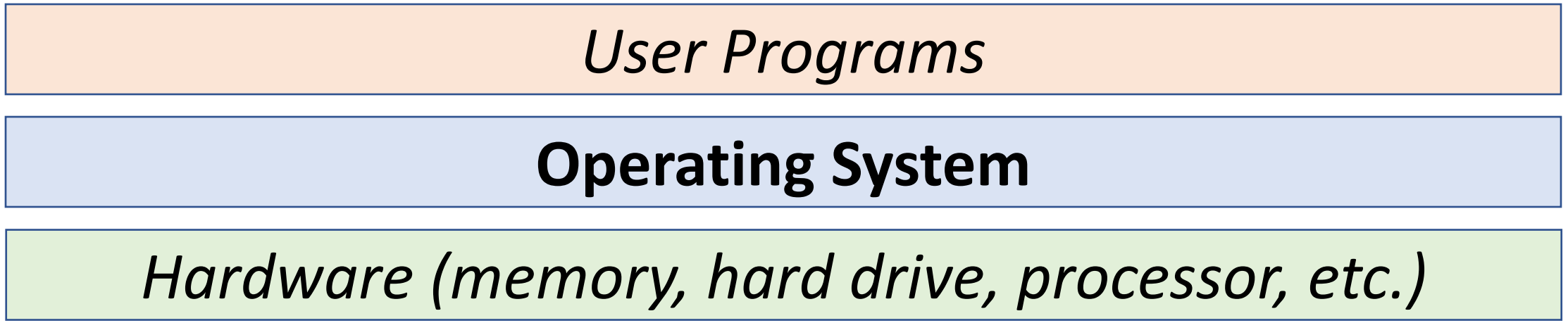
What is an Operating System?

- 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 (!)

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?*

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

Operating Systems - A Brief History

- Initially, computers were just one user a time, working directly at a computer. "OSes" were just I/O libraries shared by users for convenience and efficiency.
- Over time, computers became more *shared* (jobs were batched, users didn't need to be in front of the machine) and *concurrent* (e.g., reading in next job while a job is running), and OSes took on more responsibility.
- Later computers supported *timesharing* (a "terminal" was a screen and keyboard plugged into a machine, and there could be multiple terminals for multiple users). This introduced complexities with filesystems and systems getting bogged down with concurrent users.
- Now, we have personal computers – OSes manage concurrency, filesystems, networking, and more
- OSes popping up in even more places – very small devices, and very large (datacenters / cloud)

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 Virtual Machines
- CS111 is a relatively new class, and we're continuously working to make it the best it can be. We appreciate any and all 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

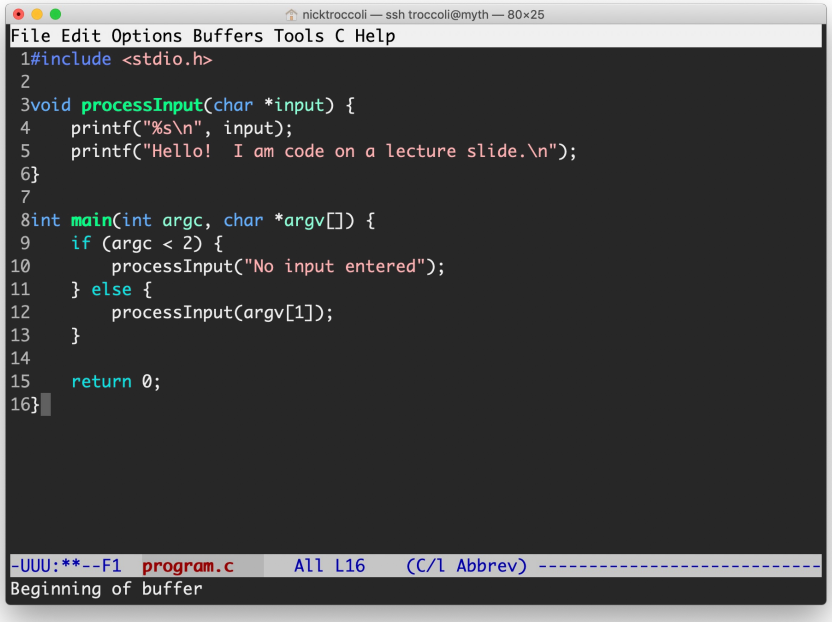
******	55%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Grading

******	55%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	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
- 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
  - Grades returned via course website



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 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 **instructor**. 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

******	55%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	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. Great preview of homework!
- Graded on attendance + participation
- Section preference submissions open **immediately following lecture today** and **are not first-come first-serve**. You may submit your preferences anytime until **Saturday 9/30 at 5PM PST**. Sign up on the course website.

Grading

******	55%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Lecture Recordings

- Because CS111 is not on SCPD (for professional development students) this quarter, the course is not officially recorded.
- However, we have recording equipment provided 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 the **start of the next lecture**. Quizzes are graded for correctness but permit 3 total attempts and should take at most 10-15min.

Lecture Points

We provide **2 pre-granted misses** for extenuating circumstances where you are unable to complete either option. Further excused misses are granted by the **instructor** 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 Friday, and lecture points will start with lecture on **Mon 10/2**.

Grading

******	55%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

Exams

- **Midterm exam** – Wednesday, November 1st, 7-9PM outside of class
  - Contact the course staff by 11:59PM on Friday, October 13th if you have an academic or University conflict with this time, and absolutely cannot make the regularly scheduled midterm. Please include all times 10/31 through 11/2 when you can take the exam.
- **Final exam** – Friday, December 15th, 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

******	55%	Assignments
*	5%	Section Participation
*	5%	Lecture Points
**	15%	Midterm Exam
**	20%	Final Exam

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

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**
- **Email** the Course Staff
  - Email instructor for **private matters** (e.g. OAE accommodations, extension requests, other personal matters).
  - Email your grader for grading questions about a particular assignment

Question Break!

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

OAE Accommodations

- Please email the instructor 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 10/13 if possible.**
- We are eager to do everything we can to support you and make you successful in CS111!

Course Flexibility

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 instructor 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 retraction policy that permits 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.

Question Break!

What questions do you have about course support or the honor code?

Assign0

Assignment 0 will be posted by 6PM today and is due in one week on **Mon. 10/2 at 11:59PM PDT**.

This assignment is an introductory assignment meant to help you get set up on the myth machines and get going with the terminal, C, and C++. It does not require any lecture material.

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 setup 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