

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/230932/external_tools/28672?display=borderless

(Each day's thread is pinned at the top)

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



Alex Nguyen



Anushka De



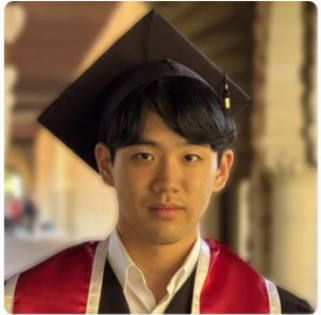
Sai Konkimalla



Sally Lee



Thomason Zhao



Elijah Song



James Varah



Peter Benitez

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/Thurs. 9:30-10:20AM in Lathrop 299 (starting week 2)
- 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! 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 9/25 at 11:59pm! (for late enrollment, email Matthew)



Matthew Mattei

mdmattei@stanford.edu

A Wonderful Community

I love scuba diving!

i love running!

I like a lot of avant-garde
prog and jazz music-wise.

I love tiramisu

Last year I did a 1000 piece
puzzle of a world map at
2am in the lounge instead
of sleeping (it took me 4
hours)



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

User Programs

Operating System

Hardware (memory, hard drive, processor, etc.)

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

User Programs

Operating System

Hardware (memory, hard drive, processor, etc.)

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.

User Programs

Operating System

Hardware (memory, hard drive, processor, etc.)

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.

CS111 Learning Goals

The goals for CS111 are for you to

- understand the **fundamental architecture** of how a computer works, and the **abstractions** implemented by the operating system
- gain exposure to **common systems design concepts** such as layering, virtualization, and logging.
- be able to **evaluate tradeoffs** between different system designs when implementing a project
- achieve proficiency in understanding and managing **concurrency and race conditions** in operating systems and in your own programs
- be able to **leverage operating-system-provided functionality** in your own programs
- work effectively with **large multi-file codebases**
- be able to think through the value of **trust**, the **responsibilities** that come with making trustworthy software, the repercussions of **breaches of trust**, and how to **build trust** into software

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. These ideas are relevant in the design of many projects.
- We can learn how we can maximally take advantage of the hardware and operating system software available to us in our programs.
- Skills like evaluating tradeoffs, navigating large codebases, understanding concurrent programs, and more, are extremely useful in many projects, whether you or AI tools are writing code.
- Operating Systems are constantly evolving and encountering new applications (e.g., large datacenters) and new challenges

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.

Plan For Today

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

Plan For Today

- Introduction
- CS111 Course Topics
- **CS111 Course Policies**
 - Prerequisites
 - Course website
 - Textbook
 - Grading
 - Getting Help
 - Honor Code

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

CGOE Students

cs111.stanford.edu/cgoe

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

*****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
*****	35%	Final Exam

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

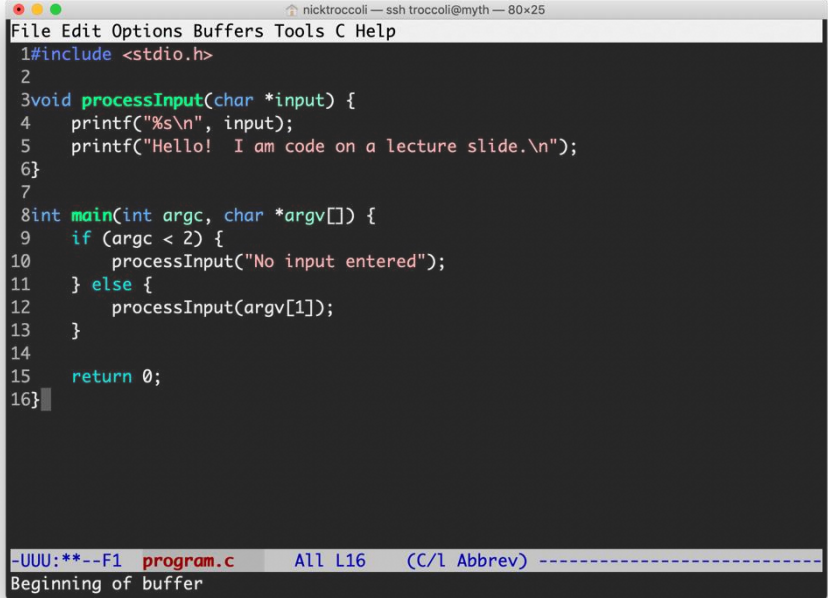
Grading

*****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

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



```
File Edit Options Buffers Tools C Help
1#include <stdio.h>
2
3void processInput(char *input) {
4    printf("%s\n", input);
5    printf("Hello! I am code on a lecture slide.\n");
6}
7
8int main(int argc, char *argv[]) {
9    if (argc < 2) {
10        processInput("No input entered");
11    } else {
12        processInput(argv[1]);
13    }
14
15    return 0;
16}
```

-UUU:**--F1 program.c All L16 (C/l Abbrev) -----
Beginning of buffer

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

Assignments are due at 11:59PM PDT, but also allow for late submissions up to 2 days after the published deadline, with a cap:

- Submitting by the published deadline does not incur any cap
- Submitting after the deadline but within 24 hours means you can get at most 95% of the points.
- Submitting between 24 hours and 48 hours after the deadline means you can get at most 87.5% of the points.

Submissions closed beyond 2 days, and this policy does not apply for the first or last assignment (assign0 and assign6), where no late submissions are accepted except for Head-TA granted extensions (extension form on course homepage).

Assignment Late Policy

Only the Head TA can grant extensions of the on-time deadline (via form submission on the course homepage) due to medical/emergency/extenuating circumstances, or extensions beyond the 2-days-late period. All extension requests must be received in advance of the assignment deadline, or as soon as possible if extenuating circumstances occur later, or extenuating circumstances prevent reaching out prior to the deadline. Please do not hesitate to reach out to the course staff or the instructor if any personal circumstances or issues arise!

Grading

****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

TA 1:1s

- Twice during the quarter, you'll meet in person 1:1 with a TA for 15-20 minutes to discuss your implementation of your recent homework assignment.
 - Week 4 + Week 8 – Week 4 1:1 focuses on assign1
- Consists of you talking through your implementation and your process for completing it and applying what you learned on the assignment in other contexts.
- **Goals:** provide a chance for you to show your knowledge of your implementation and communicate about your work and learnings!
- **Intent:** no further preparation needed beyond thoroughly working through the assignment and possessing a thorough understanding of your full implementation, process, and key assignment takeaways.
- Each TA 1:1 is worth 2.5% of the overall grade, and TA 1:1s will be offered at various times during these weeks. More information about the structure, how to sign up, and how 1:1s are evaluated will be provided around assign1's release.

Question Break!

What questions do you have about the overall course goals, textbook, assignments, or TA 1:1s?

Grading

****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

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 **now** and **are not first-come first-serve**. You may submit your preferences anytime until this **Friday 9/25 at 1PM PDT**. Sign up on the course website.

Grading

****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

Lecture Recordings

- Because CS111 is offered on CGOE (for professional development students) this quarter, the course is being recorded for later viewing.
- 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 goal is to incentivize staying current with lectures to enable you to **start early on assignments** and **get the most out of labs**.

- Starting Mon, we'll use [Poll Everywhere](#) and in-class worksheets for in-class activities. If you submit answers to all poll/worksheet questions in a lecture (regardless of correctness), you get credit for that lecture.
- We provide **3 pre-excused absences** for when you are ill, in COVID-19 isolation, or have other extenuating circumstances. They are intended only for these scenarios! Further absences reduces the overall course grade by 1%, up to a maximum of the lecture weight (5%).
- Further excused absences are granted by the **Head TA** only in cases where you have already used your excused absences for extenuating circumstances and further extenuating circumstances necessitate additional accommodations.

Lecture Points

Polls require being present in the classroom – CGOE students can complete per-lecture quizzes by the end of the week each week.

We'll do a PollEV dry run (doesn't count) in lecture Friday, and lecture points will start with lecture on **Mon 9/28**.

Grading

****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
****	35%	Final Exam

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

Exams

- **Midterm exam** – Wednesday, October 28, 7-9PM outside of class
 - Submit our exam information form by 11:59PM on Friday, Oct 9 If you have an academic or University conflict with this time, and absolutely cannot make the regularly scheduled midterm; we will offer 2 alternate midterm exam times. For any concerns, reach out to me (Nick).
- **Final exam** – Monday, December 7, 3:30-6:30PM
 - No alternate final! You **MUST** be able to take the final exam at the scheduled time (except for university athletics or OAE accommodations). If you have any concerns, please reach out to me (Nick).
- Both exams are in-person, proctored, closed-book, closed-note, closed-computer paper-and-pencil exams. You are provided with a reference sheet during the exam.
- CGOE students have a window during which to take the exams
- Submit our exam information form if you have a midterm conflict, are a CGOE student wanting to take exams on campus, or if you want a left-handed desk!

Exams

All exams are proctored this quarter, meaning that some OAE exams will be administered by the Centralized Testing Center (CTC).

OAE exam accommodations must be submitted to us at least 10 calendar days prior to the midterm, and no later than November 20 at 5:00 pm for the final exam. For urgent OAE-related accommodation needs that arise after the deadline, please consult your OAE adviser.

Grading

*****	30%	Assignments
*	5%	TA 1:1s
*	5%	Section Participation
*	5%	Lecture Points
**	20%	Midterm Exam
*****	35%	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 posted / hours start today.
- Mix of in-person-only and online-only helper hours
- Best for **group work, in-depth code questions (with TAs only!) or longer course material discussions**

Extra: Tea Hours! (time where we can chat about non-homework / course-concept-related questions such as casual course chats, CS courses in general, or anything else that might be on your mind). (See helper hours calendar)

Course Staff Contact Information

- 3 forms on the course homepage for OAE, extensions, and exam info.
- Email the **grader** on your assignment grade report for questions about assignment style or manual review grades - for questions about assignment autograder test scores, please email the **Head TA**.
- 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 **Head TA** for other requests, such as about: assignment autograder test scores, Office of Accessible Education accommodations or other accommodations, enrollment questions, auditing, or other personal matters.
- Can also email the **instructor** for questions about private/personal matters.
- 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! We will have a form posted shortly where you can submit information about any OAE accommodations – please hold off on contacting us until the form is available. In particular, please let us know of any needed exam accommodations by **Fri 10/9 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 extensions
- 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!

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, design 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.

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

The resources that we designate as off limits are designated as such because:

- We want the work you submit to be your own
- This is the best way we believe we can achieve the learning goals for CS111
- The learning happens in the process; appropriating someone else's work as your own, or using help beyond what is allowed, means you deprive yourself of that process.
- 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 or other work up to 5 days after the on-time deadline, no questions asked (except for the final exam).

Honor Code and CS111

We have sophisticated tools and detection mechanisms. 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 5 grade bucket decrease (e.g. B to D+) up to failing the course.

We tell you this to show you how seriously we take this – we want to foster an environment where everyone can do their best work with integrity.

Assign0

Assignment 0 is posted on the course website and is due next week on **Mon. 9/28 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