Course Information

Are there “laws of physics” in computing? Are there fundamental restrictions to what computers can and cannot do? If so, what do these restrictions look like? What would make one problem intrinsically harder to solve than another? And what would such restrictions mean for our ability to computationally solve meaningful problems?

In CS103, we'll explore the answers to these important questions. We'll begin with an introduction mathematical logic, proofs, and discrete structures (sets, relations, functions). These mathematical tools will enable the real heart of the course, which is to rigorously answer questions like “what does it mean for a computer to solve a problem?” and “what makes some problems (sorting) inherently harder than others (searching)?”

In the course of the quarter, you'll see some of the most impressive (and intellectually beautiful) mathematical results of the last 150 years. In some ways, I like to think of this course as a course in both art appreciation and practice. I’ll bring you through a gallery and show you some of my favorite achievements of mathematical artistic beauty, and like a good tour guide help you understand what is special about what you’re looking at. You’ll also need to pick up the paintbrush yourself and write some proofs of your own. You'll learn how to think about computation itself and how to show that certain problems are impossible to solve. Finally, you'll get a sense of what lies beyond the current frontier of computer science, especially with regards to biggest open problem in math and computer science, the P - NP problem.

People

We’re Cynthia Lee (cbl@cs.stanford.edu) and Keith Schwarz (htiek@cs.stanford.edu). This is our second time co-teaching CS103, and collectively we’ve taught the course over twenty times.

We’re joined by a group of TAs:

- Anastasiya Vitko
- Andrew Sharp
- Dunia Hakim
- Harry Sha
- Jackie Yau
- Lucy Lu
- Meredith Xu
- Suyie Zhi
- Wenzheng Li

The course staff can be reached by at cs103-aut2021-staff@lists.stanford.edu, or on the Q&A forum Ed Discussion (us.edstem.org). You may send any kind of question to either forum (course logistics, homework, lecture confusion, etc). Remember that posting any part of problem set solutions where other students can see it is a violation of the Honor Code (make such posts “staff only” on Ed). Course instructors and TAs will have office hours throughout the week. The instructors are happy to talk about topics including not only the problem sets, but also course and career advising, internships preparation, making grad school plans, etc.

On Teaching and Learning in Tough Times

The time we're living in right now is one for the history books. The following is adapted from professor Brandon Bayne of UNC Chapel Hill, with some edits for CS103.

1. This is hard for everyone.
   - It's hard for you, the students.
   - It's hard for us, Cynthia and Keith.
   - It's hard for the TAs.
   - It's hard for all the wonderful support staff on campus.
   - **Hard doesn't have to mean *worse***, but it means we have reduced capacity and that everything takes more work than usual. Our productivity is taxed. So the question is, how can we reduce that tax?
2. The humane option is the best option.
   - We are going to prioritize supporting each other as humans.
   - We are going to prioritize simplicity.
   - We are going to prioritize sharing resources and communicating clearly.
3. We will foster intellectual nourishment, social connection, and personal accommodation.
   - Accessible asynchronous content for diverse access, time zones, and contexts.
   - Optional synchronous tutorial sections to learn together and combat isolation.
   - Mastery learning approach to take-home exam assessment that is designed around flexibility, second chances, and growth as first principles.
4. We will remain flexible and adjust to the situation.
   - Nobody knows where this is going and what we'll need to adapt.
   - Everybody needs support and understanding in this unprecedented moment.

Websites and Technology

Since CS103 is run remotely this quarter, we'll be making use of a number of different online and offline tools. When in doubt, go to the course website,

https://cs103.stanford.edu,

and follow the links there. Here's a rundown of the tools we anticipate using this quarter:

- Our main course website (cs103.stanford.edu) is the main hub for course information. It contains links to everything you'll need.
- We use Canvas (canvas.stanford.edu) to host lecture videos, slides, handouts, and other course materials.
- You'll submit quizzes, assignments, and exams at Gradescope (www.gradescope.com).
- We use Ed (us.edstem.org) as a Q&A forum.
- You will need to download and install Qt Creator (https://cs106b.stanford.edu/qt/) to complete the coding assignments.
- You may find it helpful to use Overleaf (www.overleaf.com) to typeset your problem sets. (Your @stanford.edu email address gives access to "Pro" features.)

Prerequisites

CS103 has CS106B/X as a prerequisite or corequisite. This means that if you want to take CS103, you must either have completed or be concurrently enrolled in one of CS106B or CS106X (or have equivalent background experience).

Over the course of the quarter, we will be giving out a number of programming assignments to help you better understand the concepts from the course. Those assignments will assume a familiarity with C++ and programming concepts (especially recursion) at a level that's beyond what's typically covered in CS106A. The timing on these assignments is designed so that they'll sync up with what's covered in CS106B/X.

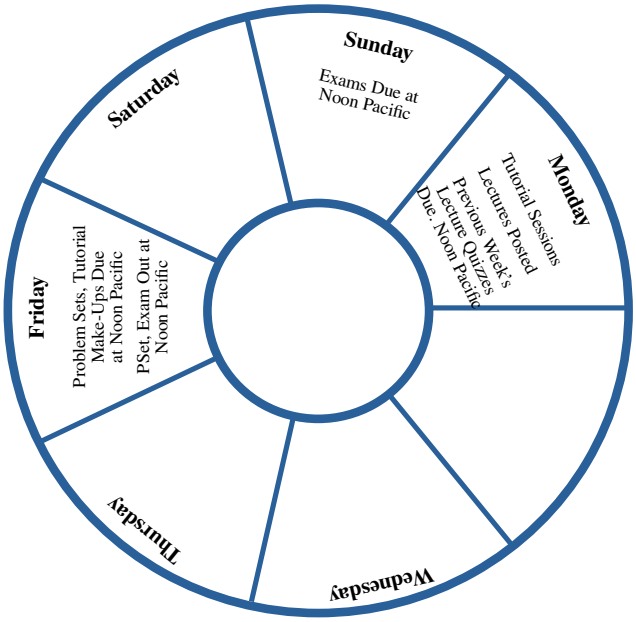
Although CS103 is a course on the mathematical theory behind computer science, the only actual math we'll need as a prerequisite is high-school algebra. We'll build up all the remaining mathematical machinery we need as we go. We've released another handout detailing the mathematical prerequisites for this course, so if you have any questions, check it out and see what you find!

Units

If you are an undergraduate or are taking this course through SCPD, you need to enroll in CS103 for five units (these are department and university policies, respectively). If you are a matriculated graduate student, you may enroll for anywhere between three and five units, depending on what best fits into your schedule. Regardless of how many units you are enrolled for, the course content and requirements will be the same. The unit flexibility is simply to make enrollment bookkeeping easier for matriculated graduate students.

The Workflow

Here's the general rhythm for how CS103 will operate this quarter:



There are four components to this course:

- ***Problem Sets***. Problem sets are due each Friday at noon. The problem sets are designed to be started Sunday at noon, but will be released on Friday at noon in case you want to get a head-start. We plan on having ten problem sets over the course of the quarter – an initial problem set to help you get your tech stack set up, plus nine "standard" problem sets.
- ***Tutorial Sessions***. On Mondays, Cynthia and Keith will hold group tutorial sessions. You are required to either (1) attend one of the tutorial sessions or (2) complete a make-up assignment consisting of some of the problems from that week's tutorial session. The make-up assignments are due on Friday at noon Pacific time.
- ***Lecture Quizzes***. Each of the lectures posted online has associated quiz questions. Those questions are due the Monday in the week *following* when the videos were released. This allows you to, say, watch Monday's lecture on Monday and complete the quiz, then watch Wednesday's lecture on Wednesday and complete the quiz, etc.
- ***Take-Home Exams***. Over the quarter, we will give four take-home exams that serve similar roles to midterms. Those exams will go out Fridays at noon in weeks three, five, seven, and nine and are due the Sunday after they're released. This gives you 48 hours per assessment, though they are not designed to take this long.

Our recommended workflow for the course goes like this:

- ***Sunday:*** If there's an exam due, submit it at noon. Start working on the problem set.
- ***Monday:*** Attend a tutorial session, watch lecture, and answer quiz questions.
- ***Tuesday:*** Continue working on the problem set.
- ***Wednesday:*** Watch lecture, answer quiz questions, and keep working on the problem set.
- ***Thursday:*** Finish up the problem set.
- ***Friday:*** Submit the problem set. If there's an exam, start working on it. Watch lecture and answer quiz questions.
- ***Saturday:*** If there's an exam, continue working on it. If not, take a breather, rest, and recharge!

Problem Sets

There will be ten total problem sets in CS103, given out once per week. They will be posted on Friday afternoons and are due the following Friday at noon Pacific time. Although the problem sets are given out over seven days, they are designed to only take five days to complete, and we expect that you'll start working on them on Sunday afternoon.

Gradescope submission. This quarter, we will be using GradeScope to handle problem set submissions and grading. Your enrollment should automatically sync with Axess, but if you just joined us and want to manually add yourself to GradeScope, visit www.gradescope.com and enter this code:

9K2X58

Once you've signed up, you can submit your assignments by uploading them to GradeScope. You are responsible for the correctness, completeness, and timeliness of your submission to Gradescope. That means you need to *allow plenty of time before the deadline* for tagging each problem, for any hiccups in the process, and for double-checking to catch errors such as uploading the wrong file, forgetting your partner's name, not tagging each problem correctly, etc.

GradeScope only accepts electronic submissions. Because in the past we've had issues with low-resolution scans of handwritten work, you are required to type your assignment solutions and submit them as a PDF; scans of handwritten solutions will not be accepted. LaTeX is a great way to type up solutions.

Coding problems. Some of the questions on the problem sets will ask you to write C++ code. You'll code these in Qt Creator on your own laptop and upload the code on Gradescope. Note that programming questions and written questions for a pset will end up as two separate uploads on Gradescope.

Partners. You are allowed to work on the problem sets individually or in pairs *(no groups larger than 2)*. Regardless of how many people you work with, your problem set will be graded on the same scale. You are not required to work with the same people on each problem set – you're welcome to work in a pair on one problem set, individually on the next, in a pair with a different partner the next time, etc. If you do work in a pair, please note that both members of the pair are responsible for ensuring that each assignment is completed and submitted on time.

For more details about collaborating with other students, please read over our Honor Code policy.

For pairs, only one person should submit to Gradescope, and that person should then add their partner's name. We rely on GradeScope for our final grading spreadsheet, so we need your partner entry to be accurate. Partners—if you *aren't* the one submitting, double-check the Gradescope submission yourself in case there was some mistake such as your partner uploaded the wrong file or forgot to add you.

Grading standards. When grading assignments, we will grade both for intuition and for execution. When looking for execution, we will check whether your reasoning is correct, whether you prove the desired result, whether all your intermediary steps are valid, etc. If your proofs contain logical errors or prove statements other than the ones you needed to prove, we may deduct points for correctness. We will also grade your proof based on how clearly it lays out its argument and whether it adheres to the standard mathematical conventions governing proofwriting. If your proof proceeds on unnecessary tangents, doesn't clearly articulate where it's going, uses unnecessarily cryptic notation or shorthand, etc., then we may deduct points. *See the forthcoming "Proofwriting Checklist" handout for more details.*

Regrade requests. We do our best in this course to grade as accurately and as thoroughly as possible. We understand how important it is for your grades to be fair and correct. If there is an error, you're encouraged to contact the course staff and ask for a regrade. However, please understand that regrades are only for *errors* in grading, not that you disagree with the grading standards or rubric. The same rubric is applied to all students, so all students lose the same number of points for the same errors. Regrade requests must be submitted no later than one week after grades are released.

Late policy. We urge you to focus on being ready for the take-home exams by finishing the problem sets on time even if not perfectly. In the software industry, they often say, "Shipped is better than perfect." Therefore, ***there are no free "late days" this quarter.*** Please note that we take the square root of your

assignment scores when calculating your final grade, which has the effect of ***hugely raising*** lower scores. This is our way of supporting you in this "shipped is better than perfect" philosophy on our end by minimizing grade impact of imperfect or incomplete solutions.

Gradescope will not accept late submissions (though it does allow you to add a forgotten partner's name after the deadline, in case you ever find yourself in that situation). If your group needs an extension on a problem set due to extenuating circumstances, you will need to email your PDF and extension request to the course staff email **cs103-aut2021-staff@lists.stanford.edu**. (Please do not email individual instructors/TAs as we have a centralized system for processing.) Emergencies exempted, All requests for extensions must be received at least 24 hours in advance of the posted due date.

Honor Code Policy. Please see Handout #03, "CS103 and the Stanford Honor Code," for our Honor Code policies.

Tutorial Sessions

There will be several instructor-hosted, interactive tutorial sessions held Mondays each week via Zoom. You should choose one of them and plan to attend that same one regularly, in order to build community with your fellow sectionees and instructor. If you are unable to attend any of the sections (whether some particular week, or ever), you may make up your attendance by viewing a recorded tutorial and completing the exercises for that week's tutorial. Ideally you would do this as early in the week as possible, since the tutorial helps with that week's pset, but final submission of tutorial make-up work is due by Friday at noon Pacific time.

Lecture Videos

Binge-watching is good for Netflix shows, but a *terrible* way to learn from lecture videos. To encourage timely viewing and as a way for you to check your understanding, there will be 3-question quizzes on each lecture. These are given on Gradescope, and are due by noon Pacific time on Monday the following week. In other words, each Sunday, lectures for Monday, Wednesday, and Friday will be released, and the 9-question quiz about them will be due the following Monday at noon Pacific time. You may resubmit the quiz as often as you wish before the deadline, so we encourage you to watch Monday lecture on Monday and submit just those 3 questions, then watch Wednesday's on Wednesday, resubmitting with your additional Wednesday responses added, and so on.

Gradescope will let you know if your answers are incorrect, and if they are you should resubmit. We'll only take your last score.

Exams

There will be ***four*** take-home exams, which will go out on Fridays at noon Pacific time and are due on Sundays at noon Pacific time. Exams are given out in weeks three, five, seven, and nine. That is:

- Midterm 1 goes out October 2nd and is due October 4th at noon Pacific time.
- Midterm 2 goes out October 16th and is due October 18th at noon Pacific time.
- Midterm 3 goes out October 30th and is due November 1st at noon Pacific time.
- Midterm 4 goes out November 13th and is due November 15th at noon Pacific time.

Unlike the problem sets, the take-home exams are strictly individual work. Even course staff assistance will be limited to clarifying questions of the kind that might be allowed on a traditional in-person exam. The problems on each take-home test are directly related to the problems on the problem set that is due the day the take-home test goes out. They are intended to ensure that, whatever open and collaborative process you may have used to complete the problem set, it ultimately resulted in you personally understanding that week's content.

To earn a passing score on an exam, you need to earn a raw score of 90% or higher. This is to enable us to certify that all alumni of CS103 have developed competency in all of the primary learning objective areas of the course. (At our discretion, we may lower this cutoff line for certain exams based on difficulty, but we will never increase it.)

Understanding that it is a tall order to ask every student to earn a 90% score first try, if you do not earn a passing score on the exam, you will be asked to revise it and resubmit it by the following Sunday at noon Pacific time. Passing the exam after a resubmission earns the same score as if you had passed the exam on your first try.

From that point forward, we will offer more opportunities to revise and resubmit the exam, though at a cost to your overall grade in the course. We'll also offer coaching and mentoring to help you grow through the resubmission process.

Readings

There are online course notes for the first few weeks of material. They go into a *lot* more depth than what we're going to end up covering in CS103, but hopefully you'll find them useful for getting a deeper understanding of the material. The course notes are still a work in progress, so please feel free to contact us with corrections of all sorts – logic errors, grammatical issues, formatting problems, etc. We also will release a bunch of handouts over the quarter to provide additional supplementary reading material. Additionally, we'll release a number of graphical guides to various concepts covered throughout the quarter. These materials are available on Canvas.

There are two *recommended* textbooks for this quarter. The first is ***How to Read and Do Proofs*** by Daniel Solow, which is a great resource for learning how to approach mathematical problem-solving. The second is ***Introduction to the Theory of Computation, Third Edition*** by Michael Sipser. You might find this book useful in the second half of the quarter. Some of the readings in the syllabus are taken from this book, but we will not directly test you on any material in Sipser that is not covered as well in lecture or the problem sets.

There are copies of each of these books in reserve in the Engineering Library.

A helpful note from the School of Engineering:

> *"All students should retain receipts for books and other course-related expenses, as these may be qualified educational expenses for tax purposes. If you are an undergraduate receiving financial aid, you may be eligible for additional financial aid for required books and course materials if these expenses exceed the aid amount in your award letter. For more information, review your award letter or visit the Student Budget website (https://financialaid.stanford.edu/undergrad/budget/index.html)."*

Equity and Inclusion

We strive to create an inclusive and equitable classroom. Since much of your experience in the course is with your peers, we also depend on you to help each other feel welcome and obtain excellence, rather than mistaking Stanford or this class for a zero-sum game.

- ***Office of Accessible Education accommodations:*** If you have an OAE letter, please present it to us (by email to the staff list) at your earliest convenience, so we can ensure that the course materials and staff support comply with your needs.
- ***Course assistance and personal support:*** In every quarter we have taught, there have been individual students who have encountered life-altering challenges, so it is not the case that empathy and compassion have only just become relevant. However, the magnitude of the current crisis underscores the need to support each other. If you feel overwhelmed for any reason—by work for this class, or a family issue, or just the weight of the present moment for the globe, please don't hesitate to reach out. We aren't trained as psychologists, but we would be more than happy to connect you with one if that's what you need. If you just need to talk, or have us send you a kitten video, or if you need extra tutoring support in the class, we are here for you. Please ask us. **Please ask us.**

Withdraw / Incomplete Policy

If a serious emergency arises and you cannot complete the work in this course, you may contact Keith and Cynthia – not the TAs – to request an incomplete. We reserve incompletes for emergencies, so we do not grant incomplete grades for poor performance on the assignments or exams, nor do we offer incompletes for busy work schedules. Withdrawing is the appropriate option in those circumstances.

In order to be eligible for an Incomplete, University policy says you must have completed a "substantial" part of the course work in "satisfactory" fashion. This means that incompletes are appropriate for serious medical or family emergencies that occur late in the quarter, which prevent you from completing the course despite having done well up to that point.

Grading

Let's start with the obvious – we are living in a pandemic in a year that will likely go down in history for its social protests. Many of us are stressed, nervous, anxious, angry, etc. Not everyone is in an environment that is conducive to learning. And yet we are still required to assign grades at the end of the quarter. Given everything we've just mentioned, our grading policy is as follows.

Your assignment score is computed as follows:

$$\text{PSet Score} = \frac{\text{sum of square roots of problem set scores}}{\text{sum of square roots of problem set point totals}}.$$

For example, if you earned a raw score of 81% on all of the assignments, you'd end up with a problem set score of 90%. ***Don't forget about the square roots here; they give a huge boost!***

Your tutorial score is computed as follows:

$$\text{Tutorial Score} = \text{number of tutorials completed or made up}.$$

Your quiz score is computed as follows:

$$\text{Quiz Score} = \text{number of quizzes where you eventually earned a 100\% score.}$$

Your exam score is computed as follows:

$$\begin{aligned}\text{Exam Score} = {} & 10 \cdot \text{number of exams passed on first submission} + \\ & 10 \cdot \text{number of exams passed on second submission} + \\ & 5 \cdot \text{number of exams passed on third submission} + \\ & 0 \cdot \text{number of exams passed on fourth or later submission}\end{aligned}$$

For example, if you earn a passing score on all four exams on the first submission, you'll earn an exam score of 40. If you earn a passing score on three exams on the first submission and a passing score on one exam on the second submission, you'll earn a score of 40. If you pass all the exams only on a third submission or later, you'll earn a score of 20.

Your letter grade is determined by picking the highest row where your scores are at or above (greater than or equal to) the cutoffs in ***all*** of the given categories.

	PSet Score (out of 100%)	**Tutorial Score (out of 9)**	**Quiz Score (out of 81)**	**Exam Score (out of 40)**
A	90%	7	72	40
B	80%	6	63	35
C	70%	5	54	30
NP	0%	4	0	0

For example, if your average assignment score (after applying square roots) is an 88%, you attended seven of the ten tutorial sessions, you earned 100% on all of the lecture quizzes, and you passed all four exams on your first submission, you'd earn a B grade overall. (Do you see why?)