

Syllabus

CS103 Course Syllabus

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.

Teaching Team

Please take a moment to get to know us by checking out the staff photos and bios on our [Ed forum intros post!](#)

The best way to reach us for any question on any topic, large or small, public or private, is by posting on the [Q&A forum Ed Discussion](#). If you need to send email, it's best to send to the staff list rather than the instructor or other individual, to be sure your message isn't missed cs103-win2223-staff@lists.stanford.edu.

Websites and Technology

The main CS103 website is where you are right now, cs103.stanford.edu. We have links to a bunch of other tools here. Here's the quick rundown:

- Main course website: cs103.stanford.edu
- Lecture videos: [Canvas](#)
- Homework submission: [GradeScope](#)
- Communication, Q&A: [Ed](#)
- Coding environment (required): [Qt Creator](#)
- Writing homework (recommended): [Overleaf](#) (Your @stanford.edu email address gives access to "Pro" features.)

Prerequisites

CS103 has CS106B 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.

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](#)

Teaching Team

[Websites and Technology](#)

[Prerequisites](#)

[Units](#)

[CS103A](#)

[Lecture Recording Legal Notice](#)

[Problem Sets](#)

[Exams](#)

[Readings](#)

[Withdraw / Incomplete Policy](#)

[Grading](#)

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.

CS103A

CS103A is an optional, one-unit companion course that runs alongside CS103. It's a great way to get extra practice with the course material and generally sharpen your theory skills. If you're interested in CS103A, you can [apply online here](#).

Lecture Recording Legal Notice

Video cameras located in the back of the room will capture the instructor presentations in this course. For your convenience, you can access these recordings by logging into the course Canvas site. These recordings might be reused in other Stanford courses, viewed by other Stanford students, faculty, or staff, or used for other education and research purposes. Note that while the cameras are positioned with the intention of recording only the instructor, occasionally a part of your image or voice might be incidentally captured. If you have questions, please contact a member of the teaching team.

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 2:30PM Pacific time. (Although we do therefore have two assignments that overlap with the weeks of our midterm exams, rest assured they will be much lighter than usual and have easier grading.)

Submitting Work

You will submit an assignment by uploading a PDF to GradeScope. Please practice being good engineers by leaving yourself a fault tolerance window of time before the due date/time, to allow for any hiccups in the uploading process, thank you! :-)

In the past we've had issues with clarity of handwritten work, so barring a very extenuating circumstance (contact course staff for approval), you are required to type your assignment solutions, not hand-write. *L^AT_EX* is a great way to type up solutions, and we'll help you get started with that tool if you like, but Microsoft Word or similar options are also acceptable.

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 computer 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 two people*). 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.

For more details about collaborating with other students, please read over our [Honor Code policy](#) and our [Guide to Partners](#).

For pairs, *only one person should submit to Gradescope*, and that person should then add their partner's name. (Should you find yourself in the situation that you forget to add your partner's name, Gradescope does allow you to add it even if the due date is passed. Partners—please double-check this each week!)

Teaching Team

[Websites and Technology](#)

[Prerequisites](#)

[Units](#)

[CS103A](#)

[Lecture Recording Legal Notice](#)

[Problem Sets](#)

[Exams](#)

[Readings](#)

[Withdraw / Incomplete Policy](#)

[Grading](#)

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 submit a regrade request on Gradescope. Regrade requests must be submitted no later than one week after that assignment/exam's grades are released.

Regrade requests that are not polite or that take issue with the rubric design (as opposed to misapplication of the rubric as it is) may be closed without comment.

Late Policy

We urge you to focus on being ready for the 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 automatic free late days or grace periods this quarter. Accommodations for disability or special circumstances are of course available by contacting the course staff via the staff mailing list cs103-win2223-staff@lists.stanford.edu.

Please note that we take the square root of your assignment scores when calculating your final grade, which has the effect of hugely raising low scores. This is our way of supporting you and meeting you halfway in this "shipped is better than perfect" philosophy, by minimizing the grade impact of imperfect or incomplete solutions.

If you need an extension, *submit whatever you have before the due date/time* and then contact us via the staff mailing list cs103-win2223-staff@lists.stanford.edu. That way, if you hear back that your situation is not something for which we can grant a due date extension, you at least have the credit for what was done. It's always better to reach out to us about extensions and special circumstances *as soon as at all possible*, rather than letting us know after due dates have passed.

Honor Code Policy

Please see our [Honor Code](#) page for more information.

Exams

There will be *two* midterm exams. We are planning on holding exams in-person this quarter, though this is subject to change based on university guidance. The exams will run on the following days:

- Midterm 1 is Tuesday, Feb 7 from 7:00PM - 10:00PM.
- Midterm 2 is Tuesday, Feb 28 from 7:00PM - 10:00PM.
- Final Exam is Tuesday, Tuesday March 21, 8:30AM - 11:30AM.

For exam locations and more information, see the [Preparing for the Midterm](#) page.

We understand that very few students have written formal mathematical proofs before this class, and that there is a learning curve to aligning with our expectations for writing style, level of detail, etc. Therefore, we have a special "revise and resubmit" policy for the first exam, to help students who unfortunately discover on that exam that they were far out of alignment. What that means is that after we hand back your graded midterm, you'll have a window of time to change your answers and have your exam regraded. Your old score can be completely thrown out in favor of the new score, though we do apply a 90% "discount" for this second attempt. Specifically, your score on the exam is given by the maximum of your original score and $.90 \times$ your new score. So, for example, if you earned 50% the first time and 100% after the resubmission, your new score would be $100 \times .9 = 90\%$. (There is no "revise-and-resubmit" for the second midterm or the final exam.)

Readings

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. We will never directly test material available only in the textbooks; the course materials we provide will be all you need.

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

A helpful note from the School of Engineering:

[Teaching Team](#)

[Websites and Technology](#)

[Prerequisites](#)

[Units](#)

[CS103A](#)

[Lecture Recording Legal Notice](#)

[Problem Sets](#)

[Exams](#)

[Readings](#)

[Withdraw / Incomplete Policy](#)

[Grading](#)

“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](#).”

Teaching Team[Websites and Technology](#)[Prerequisites](#)[Units](#)[CS103A](#)[Lecture Recording Legal Notice](#)[Problem Sets](#)[Exams](#)[Readings](#)[Withdraw / Incomplete Policy](#)[Grading](#)

Withdraw / Incomplete Policy

If a serious emergency arises and you cannot complete the work in this course, you may contact us by the staff mailing list to request an incomplete. By [University policy](#), incompletes are reserved for unforeseeable emergencies that arise late in the quarter during which a student was otherwise performing well (“...restricted to cases in which you have satisfactorily completed a substantial part of the coursework...”), so we do not do incomplete grades for ongoing performance struggles or overloaded course schedules or busy jobs. Retaking is the appropriate option in those circumstances.

Grading

Your raw score in CS103 is determined as follows:

$$\text{Raw Score} = 0.25 \cdot \text{PSet Score} + 0.75 \cdot \text{Exam Score}.$$

Here, your problem set score is computed as

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

Taking the square root of each problem set score provides a boost to each problem set grade. For example, if you score an 81% raw score on one problem set, we’d count it as though you’d earned a 90%. We do not drop your lowest problem set score.

Your exam score is computed as

$$\text{Exam Score} = \frac{1}{6} \cdot \text{First Midterm Percent Score} + \frac{1}{3} \cdot \text{Second Midterm Percent Score} + \frac{1}{2} \cdot \text{Final Exam Percent Score}.$$

Your midterm and final exam scores are computed using raw percentage scores. Note that, in particular, this means that we do not curve midterm or exam scores.

We never assign letter grades that are lower than the decile of your raw score; for example, a 90% will never map to anything lower than an A-. Historically, the median raw score has ended up somewhere near the B/B+ cutoff. If grades end up lower than that, we may “curve” upwards by letting the A- grade cutoff go down to 89% or 88%, etc.

To earn a passing grade in CS103, your PSet Score and your Exam Score, as computed above, must each be passing work. (A 60% cutoff for problem sets, and 50% exams, though these may be lowered to allow more students to pass in some quarters.) If your score in at least one area is below this cutoff, you will receive a non-passing grade. This is to ensure that students take the assignments, and don't ride high assignment scores (potentially done with a partner) to pass without demonstrating independent mastery on the exams. Historically, very few students are impacted by this rule, since usually having a non-passing score in either area results in having a low overall composite score anyway.

All course materials © Stanford University 2022.

Website programming by Julie Zelenski with minor edits by Keith Schwarz. This page last updated 2023-Feb-03.