Course Information

Course Overview

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 to mathematical proofs and discrete structures, which will enable us to model problems that arise in computer science. In the course of doing so, we'll explore mathematical logic, discrete structures, and the mathematical nature of infinity.

We'll continue by exploring finite automata (mathematical models of computers with finite memory) and from there will explore context-free grammars and Turing machines (mathematical models of computers with unbounded memory). As we explore these models, we'll see their strengths and their weaknesses and will explore questions like "what does it mean to solve a problem?" and "why does this problem seem to resist a solution?" Finally, we'll conclude with a quick introduction to complexity theory and explore what we know – and what we don't – about efficient computation.

In the course of the quarter, you'll see some of the most impressive (and intellectually beautiful) mathematical results of the last 150 years. You'll see what proof-based mathematics is all about and will gain confidence using mathematics to model and solve problems. You'll learn about various discrete structures that arise throughout computer science. 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 on the frontier of computer science, especially with regards to the $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ problem.

Instructors

Keith Schwarz (htiek@cs.stanford.edu)
Office: Gates 178

TAs

Joshua Kravitz (kravitzj@stanford.edu) *(Head TA)*

Julian Alvarez
Ellis Hoag
Timothy Le
John Melloni
Teresa Noyola
Divya Saini
Magdy Saleh
Ryan Smith
Pranav Sriram
Ruben Mayer
Michael Zhu

Website

The course website is cs103.stanford.edu and it's loaded with resources for this course. There, you'll find all the handouts and lecture slides, along with additional links you may find useful. I would suggest periodically polling the website to stay on top of any important developments in the course.

Email

The course staff can be reached at cs103-staff@cs.stanford.edu. Please don't hesitate to email us! We're here because we genuinely love this material and want to share it with you. If you have any questions on the material, or if you're interested in exploring more advanced content, please get in touch with us. We'd be happy to help out.

Piazza

We have a class Piazza forum you can use to ask questions about the material and to get help and advice on the problem sets and discussion problems. Our policies regarding Piazza use are covered in our Problem Set Policies handout.

Lectures

Mondays, Wednesdays, and Fridays, 3:00 – 4:20 in NVIDIA Auditorium. Lectures will be recorded and are available through SCPD. Attendance is highly encouraged.

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 life easier for matriculated graduate students.

Five-unit courses at Stanford vary greatly in their difficulty. Based on past student experiences, you should expect that this course probably will require a time investment proportional to its unit load. Expect to put in around 15 hours each week – including lecture time – working on CS103. We'll offer a lot of support through office hours, extra practice problems, and practice exams, and if you're willing to put in the effort to learn the material, the course staff will be behind you every step of the way.

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!

If you're interested in taking this course but feel that you might not have a sufficient mathematical background, you may want to check out our add-on course, CS103A, which is discussed later in this handout.

Office Hours

Keith and the TAs will be holding *lots* of office hours during the week so that you can stop by and ask questions about the material. Feel free to stop on by if you need any help. We'll post a schedule later this week.

CS103A

CS103A is an optional, one-unit add-on course for CS103. CS103A meets once a week for two hours and offers extra review and practice problems related to the current course content. If you're interested in taking CS103 but feel like you might need a little bit of extra practice and review, we'd strongly recommend checking out CS103A.

Due to the structure of CS103A as a course, CS103A is not recorded over SCPD. However, all materials from CS103A will are available to everyone enrolled in CS103. Check out the CS103A course website (https://cs103a.stanford.edu) to get the most recent sets of practice problems.

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.

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.

Problem Sets

There will be ten total problem sets in CS103, given out about once per week. With the exception of Problem Set 0, which must be done individually, you are welcome to work on them individually or in pairs. Our full policies with regards to problem sets (late policy, regrades, etc.) are in the Problem Set Policies handout.

Exams

In addition to problem sets, there will be a two midterm exams and a final exam. The first midterm exam will be held on Monday, October 22nd from 7PM – 10PM and the second on Monday, November 12th from 7PM – 10PM, both locations TBA. The final exam will be held on Monday, December 10th from 3:30PM – 6:30PM, location TBA. SCPD students will receive information over email about taking the exam remotely.

In accordance with university policy, with the exception of OAE accommodations, we will not offer any alternate final exam times. If you are unable to take the final exam at the stated time, you will need to take this class in another quarter.

Additionally, with the exception of OAE accommodations, we generally do not offer alternate midterm exam times. You should not enroll in CS103 unless you can make all three of the exam times.

Grading

We compute final grades based on three subscores: one for problem sets, one for midterm exams, and one for the final exam.

Your problem set score is computed as

$$\text{PSet Score} = (\text{Points Earned / Non-Extra-Credit Points Possible})^{0.5}.$$

In other words, we will take your raw composite problem set score, then take the square root. This has the effect of boosting your problem set grades. For example, if you have a 81% raw score on the problem sets, you'd end up with a 90% for your assignment score. Similarly, if you had a 64% raw score on problem sets, you'd end up with an 80% for your assignment score. The problem sets are where you will do most of the learning in this course, and it's important that you complete each of them. Therefore, we do not drop your lowest problem set score.

Your midterm score is computed by weighing your two midterm scores as follows:

$$\text{Midterm Score} = {}^{2}/_{3} \cdot \text{Higher Midterm Score} + {}^{1}/_{3} \cdot \text{Lower Midterm Score}.$$

For example, if you earned a 75% on the first midterm and a 90% on the second midterm, your midterm score would be an 85%. If you earned a 100% on the first midterm and a 50% on the second midterm, your midterm score would be an 83.3%. Aside from this dynamic weighting, we do not curve midterm exam scores.

Your final exam score is computed as an unmodified raw score:

$$\text{Final Exam Score} = \text{Points Earned / Points Possible}.$$

Note that, in particular, this means that we do not curve final exam scores.

With these three subscores in mind, we will compute your final raw score at the end of the quarter as

$$\text{Raw Score} = {}^{1}/_{3} \cdot (\text{PSet Score} + \text{Midterm Score} + \text{Final Exam Score}) - \text{Late Penalty}.$$

(The late penalty accounts for late problem set submissions and is described in the Problem Set Policies handout). We then apply a grading curve to raw scores to assign letter grades. Historically, we've used the median raw score as the B/B+ cutoff. 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-.

Your final grade will be determined solely according to the grading curve as applied to the raw score computed above. We do not offer any make-up work.

Honor Code

We want to foster a collaborative and supportive atmosphere in CS103. This is why, for example, we will have so many office hours sections and why we let you work in pairs on the assignments. We expect you to abide by the letter and the spirit of the Stanford Honor Code in CS103. You are required to read and abide by the policies detailed in our handout on the Honor Code as it applies in CS103, which among other things discusses our expectations for what is and is not permissible collaboration on the problem sets.

We hope that you will respect the Honor Code, comport yourself with integrity, and work to create a learning environment where everyone feels supported.

Incomplete Policy

If a serious medical or family emergency arises and you cannot complete the work in this course, you may contact Keith – not the TAs – to request an incomplete. We reserve incompletes only 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.

In order to be eligible for an incomplete, you must have completed all of the assignments (except possibly the most-recently-due assignment) and must have a solid academic performance in the course, as determined by the instructor. The instructor has the final say in whether to grant or deny incompletes. While the above criteria indicate certain cases in which incompletes will not be granted, there are no situations in which the instructor is obligated to offer an incomplete.