Course Information

Instructor Keith Schwarz (htiek@cs.stanford.edu)
Office: Gates 178
Office Phone: (650) 723-4350

TAs

Rakesh Achanta	(rakesha@stanford.edu)
Kyle Brogle	(broglek@stanford.edu)
Maurizio Calo Caligaris	(maurizio@cs.stanford.edu)
Justin Heermann	(justinlh@stanford.edu)
Nicholas Isaacs	(nisaacs@stanford.edu)
Jeffrey Jacobs	(jjacobs3@stanford.edu)
Michael Kim	(michael.kim@cs.stanford.edu)
Rose Marie Philip	(rosep@stanford.edu)
Remington Wong	(rwong2@stanford.edu)

Email The course staff can be reached at cs103-aut1213-staff@lists.stanford.edu. Please don't hesitate to send us emails! 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.

Lectures Mondays, Wednesdays, and Fridays, 2:15PM – 3:30PM in Braun Auditorium.

Units If you are an undergraduate, you should be enrolled for five units. If you are a 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 it easier for graduate students to enroll without exceeding unit caps.

Prerequisites This course has CS106A as a prerequisite. We want you to have at least a basic familiarity with computer programming before taking the course, since many of the results we'll be exploring will be intimately connected to computers, computing, and programming. That said, there will be no actual programming assignments in this course. If you have not taken CS106A but are still interested in taking this course, I would suggest dropping by office hours so that we can chat about whether the course is a good fit for you.

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

Office Hours We will try to hold office hours for at least four hours a day, six days a week. We'll announce the office hours schedule later this week.

Problem Sessions This course will have an optional problem session that meets once a week to work through additional exercises. Although session attendance is not mandatory, we **strongly suggest** that you attempt to attend each section. The problems covered in these sessions will mirror what will be asked on the homeworks and exams.

Each week, we will distribute a handout containing section problems. The solutions will be distributed at the problem session, as well as on the course website.

We will announce the time and location of the problem session in lecture.

Readings This course has two required readings:

> Chapter One of ***Discrete Mathematics and Its Applications, Sixth Edition*** by Kenneth Rosen. This chapter from Rosen's book on discrete mathematics covers mathematical logic, which we'll explore in the third week of the course.
>
> ***Introduction to the Theory of Computation*** by Michael Sipser. Sipser's excellent introduction to computation, computability, and automata theory stands as one of the best textbooks on the subject. We will be using this textbook through the second half of the quarter. We will support both the second and third editions of this textbook.

There are also online course notes for the first few weeks of material. This online course reader has been expanded since previous quarters, so we hope that it will provide more in-depth coverage of the content from lecture. Since this course reader is still a work in progress, we will award extra credit for finding typos or other mistakes in the course reader. We will make an announcement in-class about the specifics of this extra-credit policy.

Problem Sets CS103 is a course designed to teach you the mathematical foundations of computation, along with the techniques necessary to reason about structures that appear throughout computer science. Accordingly, the assignments in this course are designed to give you the chance to play around with the material and sharpen your skills with mathematical proofs, computability theory, and complexity theory. There will be **nine** homework assignments this quarter, each of which is weighted roughly evenly.

We will split some of the problem sets into two pieces – some "checkpoint" problems and a set of graded problems. You will submit the checkpoint portion of the problem set earlier than the rest of the problem set, and it will be graded on whether or not you have made a good honest effort to solve the problems, rather than on correctness. The TAs will then comment on the structure of the proofs in your checkpoint submission (looking at clarity, correctness, etc.) and return your solutions within a few days. We hope that this feedback will enable you to write better proofs for the remainder of the problem set, which will be graded on correctness.

Grading

In addition to the problem sets, there will be a midterm and a final exam. The midterm exam will be held on **Monday, October 29** from **7:00PM – 10:00PM**, location TBA. The final exam will be held on **Wednesday, December 12** from **12:15PM – 3:15PM**, location TBA. If you have a conflict that will prevent you from taking the exam, we'd be happy to try to find an alternate time. However, you must let us know about this no later than **one week** in advance.

Overall, your grade for this course will be determined as

Written Assignments:	**60%**
Midterm:	**15%**
Final Exam:	**25%**

Late Policy

This course is fast-paced and we'll be moving through material quickly. Because of the diversity of the material we'll be exploring, you may find that some homework assignments are easier or harder than others. To give you some extra flexibility, you may submit up to **three** assignments **72 hours** past the due date without penalty. You don't need to let us know that you'll be turning in the assignment late, though we'd appreciate a heads-up just so that we know to expect it. While you may use these late days on any assignments that you would like, **you may use at most one 72-hour extension per assignment**. Late days may be used on the checkpoint assignments, though we would strongly prefer if you tried to get them in on time since they're designed to help prepare you for the rest of the problem set. Turning it in 72 hours late will make it impossible for us to offer feedback before the main assignment comes due.

If you have already used all of your 72-hour extensions and submit assignments past the due date, we will assess a 10% penalty per day late. To make it possible to release solutions to problem sets on time, no submissions will be accepted more than 72 hours after the assignment due date, even if you use a 72-hour extension.

If you have any extenuating circumstances, such as a family or medical emergency, and need extra time to complete the assignments, please contact us. We're more than happy to accommodate. However, please let us know before the assignment comes due.

Regrade Requests

Everyone makes mistakes, and if we make a grading error on one of the homeworks or an exam, please let us know and we'll take a second look. Just hand the exam or problem set to one of us with a coversheet describing what error you believe we've made. However, if you submit an exam or assignment for a regrade, we reserve the right to regrade all problems on it, not just the ones you've indicated. Additionally, all regrade requests for problem sets must be received within **three days** of the date in which the problem set is returned, and regrade requests for exams must be received within **one week** of the date in which the exam is returned.

Honor Code

One of the major goals of CS103 is to teach you how to think about problems mathematically and rigorously. Just as you have to write a lot of code to become a good programmer, you need to work through a lot of problems to become a good mathematician. Consequently, when doing the homework assignments, it is extremely important that you do as much of the work as possible on your own without consulting anyone else or any other outside resources. It's surprisingly easy to fall into a trap where you learn to follow proofs without having any idea how to synthesize them on your own. As much as possible, please try to work on the assignments individually.

That said, I understand that you may want to work on the problem sets in groups. If you'd like to do this, that's totally fine. However, please be sure that on your problem set you include the names of the other people in the group. Additionally, **you must write up your own solutions** to all the problems. It is a violation of the Honor Code to copy answers from another student.

Some of the questions on the problem sets may have been used in past quarters (especially if they're really cool problems!) Because of this, **it is a serious violation of the Honor Code to consult graded problem sets or solution sets from previous quarters**. More generally, you **must not** copy solutions from any source. That's just dishonest.