

Course Information

| | |
|------------------------|---|
| Course Overview | This course is designed as a deep dive into the design, analysis, implementation, and theory of advanced data structures. Over the course of the quarter, we'll explore fundamental techniques in data structure design (isometries, amortization, randomization, etc.). In doing so, we'll see a number of classic data structures, as well as some more modern ones. By the time we've finished, we'll have seen some truly beautiful strategies for solving problems efficiently. |
| Instructor | Keith Schwarz (htiek@cs.stanford.edu) |
| TA | Kevin Tan. (Kevin is a veteran of CS166!) |
| Email | The course staff can be reached at cs166-spr2223-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 | <p>Lectures are on Tuesdays and Thursday from 3:00PM – 4:20PM in 200-034. Lectures are not recorded this quarter.</p> <p>Attendance at lectures is important, both because that's where we're expecting you to learn many of the topics and because it contributes to a sense of community. To that end, lecture attendance is a small component of your grade this quarter. Specifically, we'll be asking in-class questions on PolleEV starting in the third week of the quarter. You're allowed to miss three lectures without it impacting your course grade.</p> |
| Units | CS166 is offered for either three or four units. Undergraduates are required to enroll for four units, while graduate students can enroll for either three or four units. The course content and requirements are the same in the three-unit and four-unit versions of the course and the unit flexibility is purely to help graduate students stay under unit limits. |
| Website | The course website is http://cs166.stanford.edu and it's loaded with resources for this course. There, you'll find all the course handouts, the syllabus, links to readings, and all sorts of other resources. We will not host much on Canvas aside from the lecture Zoom links for the first week of the quarter and videos for the first two weeks. |
| Office Hours | It's important to us that everyone in the course feels that they can easily get in touch with the course staff when they have questions. We will be holding a office hours time slots throughout the week. We'll announce when these are in lecture at the end of the first week. |

Prerequisites

CS166 is a course in advanced data structures and is intended for advanced undergraduates and beginning graduate students. If you are looking for a traditional CS course that functions as an introduction to data structures, we recommend checking out CS106B.

The prerequisites for CS166 are CS161 and CS107.

From CS161, you should be able to design and analyze nontrivial algorithms and write proofs of correctness. You should be comfortable using asymptotic notation (\mathcal{o} , \mathcal{O} , Θ , Ω , and ω), solving recurrence relations, manipulating inequalities, and simplifying summations. We'll also expect that you're comfortable with divide-and-conquer algorithms, greedy algorithms, and dynamic programming; that you're familiar with randomized algorithms (and related concepts like universal families of hash functions); and that you're comfortable writing correctness proofs for algorithms of each of these types. You should also feel comfortable with standard algorithms like Dijkstra's algorithm, Prim's algorithm, quicksort, mergesort, etc. and feel comfortable with their runtimes and proofs of correctness.

The CS161 prerequisite, by transitivity, also means we assume you have the equivalent of CS103 (discrete mathematics, automata, and proofwriting) and CS109 (probability and basic combinatorics) as well. If you have never written a formal mathematical proof, or if the phrase “linearity of expectation” doesn't ring a bell, you may want to come talk to us before jumping into CS166.

From CS107, we expect that you're comfortable writing and testing nontrivial programs and working from the command line. You should also feel comfortable with binary representations of numbers. We'll expect that you've at least heard of the memory hierarchy and are comfortable with the idea that not all memory accesses take the same amount of time. Additionally, we expect that you'll be comfortable writing code in C++.

If you're unsure whether CS166 is the right place for you, please feel free to get in touch with the course staff.

Readings

The *recommended* reading for this course is *Introduction to Algorithms, Third Edition* by Cormen, Leiserson, Rivest, and Stein. We understand that not everyone has a copy of this book or can get a copy, and that's okay. There's nothing in that textbook that you'll absolutely need for the course, and it's mostly there as a reference in case you'd like to look at certain topics in more depth.

Additionally, there will be a variety of readings posted online (papers, course notes, slides, articles, etc.) Check the website for details on the readings for each lecture. I will try to present the salient features of each data structure in lecture, so depending on your learning style, you may find it useful to do the readings right before or right after lecture.

Assignments

This quarter, we'll have a total of `[[todo!]]` assignments. Aside from Assignment 0, which must be done individually, the assignments may be done either individually or in pairs. If you work in a pair with someone, you must make a single joint submission with both your names on it.

You are welcome to choose problem set partners however you'd like. Our only policy is that if you start working with someone on one of the assignments, you need to finish that assignment jointly with them. (You can switch partners between assignments however you'd like.)

Exams

We will be holding a midterm exam on Tuesday, May 30th from 7PM – 10PM, location TBA. If you cannot make this exam time, please let us know as soon as possible.

Grading

Your grade in CS166 is determined as follows:

Problem Sets: 55%
Lecture Participation: 5%
Midterm: 40%

We use raw scores for each component and apply a grading curve at the end of the quarter that can only help. We never curve down, so a 90% will always map to some flavor of A.

One caveat: in order to earn a passing grade in CS166, you must earn a passing score in both the “Problem Sets” and “Midterm” components of your grade. We will determine what “passing” means at the end of the quarter, but plan on setting the passing threshold for the problem sets at a 60% average score and the midterm at a 50%.

Incompletes

If you have a medical or family emergency and 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 project, 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 satisfactory academic performance as determined by the course instructor.

All incompletes are worked out on a case-by-case basis, and the instructor retains final discretion to approve or reject any requests for an incomplete.