

Course and Requirement Overview

Here you can find information about the requirements for the Computer Science degree.

Overview

School of Engineering Requirements

- Math

- Science

- Engineering Fundamentals

- Technology in Society

CS Department Requirements

- Core

- Track

- Electives

- Senior Project & Writing in the Major

Deviations and Petitions

Overview

The CS curriculum provides knowledge that is applicable across many fields, including many areas of engineering, science, and medicine.

The core material of the computer science program has been streamlined into six classes that every CS student takes. Students then pick an area that they would like to study in more depth, which becomes their track specialization. The track typically consists of 4-5 classes. Finally, students round out their CS curriculum with 2-4 elective courses. The electives can be classes from different areas in CS if a student desires more breadth, or they can be additional classes related to the student's track if a student desires more depth. The track-related electives also include courses outside of the CS department for students whose interests cross departmental boundaries and who would like to do interdisciplinary work.

For more details about specific classes, check out course ranking sites Carta, old class webpages, and official course evaluations. Course ranking sites have student comments and some official registrar data about course difficulty and instructor quality. Class webpages from previous quarters will often contain handouts and old assignments that can give you a good feel for the class material. Most CS classes are accessible via a .stanford.edu subdomain with the course number as the subdomain. For example, the CS107 page can be found at cs107.stanford.edu. You can find official course evaluations on Axxess under Student > Course and Section Evaluations.

When planning a course load, it is a good idea to take into account not only the difficulty of a

class but also the type of assignments. Taking 3 classes that all have programming assignments and a large final project is probably not the best idea. Typically, a balanced schedule includes some mix of programming classes and problem set classes. More tips are available on the courseload webpage.

The information on this page is kept up-to-date as much as possible, but it is always a good idea to verify that it matches the undergraduate engineering handbook or your specific program sheet since they are the definitive sources of information on these matters.

School of Engineering Requirements

These requirements are set and approved by the School of Engineering. Changes to these requirements must be approved by the School of Engineering Office of Student Affairs in Huang 135 one quarter prior to a student's last quarter. For most students, this is winter quarter of senior year. More instructions and forms are available at the UGHB petitions website.

Using AP Credit to satisfy math and science requirements must also be approved by the Office of Student Affairs. To receive this approval, take a program sheet and unofficial transcript to Huang 135.

The math, science, and engineering fundamentals requirements for the biocomputation and computer engineering tracks differ from those listed here. Please see the program sheets for specifics.

Math

Calculus

CS majors may use Math 19/20/21, 41/42, or AP credit to satisfy the calculus requirement. For more information, refer to the Calculus sections of the Stanford AP Credit Chart.

Mathematics in Computer Science

The undergraduate curriculum requires two CS math classes: Mathematical Foundations of Computing (CS103) and Introduction to Probability for Computer Scientists (CS109). These classes give students the necessary mathematical fundamentals to be successful in computer science at Stanford and beyond.

Math Electives

Two additional math electives are required for the CS major. The electives can be any combination of classes from an approved set of courses, listed on the first page of the program sheets. The current electives list includes classes such as the Math 51, CME100, CME102 and

CME104.

Because of significant overlap in the material covered, certain class combinations cannot be counted towards the math requirement. MATH51 and MATH103, MATH103 and MATH113, or CS157 and PHIL151 may not be used in combination to satisfy the Math electives requirement. Students who are applying both MATH51 and MATH52 towards their program sheet, either as math electives or as track requirements, cannot also apply CME100 to their program sheets.

Science

Physics

CS majors must take both a Mechanics class (PHYSICS21, PHYSICS41, PHYSICS61) and an Electricity and Magnetism class (PHYSICS23, PHYSICS43, PHYSICS63). Mixing classes from different series is acceptable. Physics labs are not required for the CS major. In general, almost any college level, first year physics class that covers mechanics and E&M will be awarded transfer credit and count towards the physics requirement. So if you want to take physics at Foothill College, the local community college, or at home over the summer, that is usually fine. The School of Engineering approves this requirement so it might be wise to check with them ahead of time if you're planning to do this. AP credit will also satisfy the physics requirement. For more information, refer to the Physics sections of the Stanford AP Credit Chart.

Science Elective

The science elective is one or perhaps two classes to bring the total science units to at least 11. The list of approved electives includes the School of Engineering science electives list, plus PSYCH30 (Perception). A 5 on the AP Chemistry exam will count as 4 units of science elective credit. This year's Undergraduate Engineering Handbook will also have an up-to-date list of classes that can be counted towards the science elective.

Engineering Fundamentals

Programming Abstractions (CS106B or CS106X)

CS106X covers the same topics as CS106B, but with more in-depth coverage and at a faster pace. Note that it is also possible to take CS106X (rather than CS106B) after taking CS106A. It is also strongly recommended that you take CS106X even if you've had a considerable amount of prior programming experience, particularly if you do not know C++. This will ensure you have the necessary foundation for more advanced work expected in later CS classes.

Introductory Electronics (ENGR40)

ENGR40 is an introductory electronics class taught by the EE department. The class has weekly problem sets, labs, a midterm, and a final.

Engineering Fundamentals Elective

The engineering fundamentals elective is another class from the School of Engineering engineering fundamentals list. This year's Undergraduate Engineering Handbook will also have an up-to-date list of classes that can be counted towards the engineering fundamentals elective. Effective as of the 2017-2018 program sheets, you can now count an additional CS depth course (track or general elective) in place of the second engineering fundamentals course.

Technology in Society

The TIS class is designed to give engineers an understanding of their profession as a social activity. The TIS classes are devoted to exploring issues arising from the interplay of engineering, technology, and society. Any class from the School of Engineering technology in society courses list can be counted towards this requirement. This year's Undergraduate Engineering Handbook will also have an up-to-date list of classes that can be counted towards the technology in society requirement.

Computer Science Department Requirements

Core

Systems

Programming Abstractions (CS106B or CS106X)

CS106B introduces students to many fundamental programming concepts and software engineering techniques using the C++ language. The course will focus on teaching problem solving skills, basic abstract data types, and recursion. General topics include basic programming methodology (engineering, modularity, documentation), data abstractions (stacks, queues, linked lists, hash tables, binary trees, generics and templates), recursion (procedural, backtracking), searching and sorting, and basic algorithmic analysis (including Big-Oh notation).

Computer Organization and Systems (CS107)

CS107 transitions students to programming on the UNIX machines. The class aims to teach students about computer systems from the hardware up to the source code. Topics include machine architecture (registers, I/O, basic assembly language), memory models (pointers, memory allocation, data representation), compilation (stack frames, semantic analysis, code generation), and basic concurrency (threading, synchronization).

Principles of Computer Systems (CS110)

CS110 will teach students how to build larger scale systems using operating system and networking abstractions. Topics include processes (threading, context switching, interprocess

communication), storage and file management (file systems, virtual memory), networking (sockets, TCP/IP, routing) and an understanding of distributed systems.

Theory

Mathematical Foundations of Computing (CS103)

CS103 will give students the mathematical foundations necessary for computer science. Topics include proof techniques and logic; induction; sets, functions, and relations; an introduction to formal languages; DFA's, NFA's, and Regular Expressions; Context-Free Grammars, Turing Machines, and NP-Completeness.

Introduction to Probability for Computer Scientists (CS109)

CS109 is designed to teach students material from probability and statistics that is relevant to computer science. Topics include combinatorics; probability theory; conditional probability, and independence; probability distributions; Bayes' Theorem, Law of Large Numbers, and the Central Limit Theorem; and hypothesis testing. The class will also cover applications of probability including hashing, data analysis, inference, and an introduction to machine learning.

Data Structures and Algorithms (CS161)

CS161 gives students the tools to analyze data structures and algorithms. Students will also practice devising algorithms for various problems. These skills are widely applicable and alumni report CS161 as one of the most useful classes at Stanford. Topics include algorithmic complexity analysis (Big Oh, Omega, Theta), recurrence relations, and the master method. In addition, students learn about several different classes of algorithms and data structures, including randomized algorithms, divide and conquer strategies, greedy algorithms, hashing, heaps, graph algorithms, and search algorithms (including blind and A* search).

Track

The tracks allow students to develop depth in a particular specialization by taking 4-5 courses in a related area. Students must complete the requirements for any one track which typically includes 1-2 gateway classes, 1-2 classes from a menu of highly-related courses, and then a selection of classes from a list of more broadly related courses. Together, the track plus electives must equal at least 7 classes and 26 units.

The current tracks are Artificial Intelligence, Theory, Systems, Human-Computer Interaction, Graphics, Information, Biocomputation, Unspecialized, and Individually Designed. For details about individual tracks and advice for choosing a track, see the tracks page. For more details about the specific academic requirements for each track, please refer to the relevant program sheet.

Electives

Some people use the electives to explore a broader array of topics in CS by taking additional introductory classes in different areas. Others use them to pursue a particular topic in more depth by taking some of the higher level graduate classes. This provides students additional flexibility to build a broader base of knowledge in CS or related areas, or focus on a more specific topic. For students doing research, professors or advisors will be able to recommend relevant classes that can be used as electives.

The elective requirements can be fulfilled by taking classes from either of two lists, the general CS electives list or the list of classes that fulfill the elective requirements specific to each track. These track-specific electives often include classes outside the CS department to allow for interdisciplinary work. Check out a program sheet for the full list of available classes.

If you think a class should be on the list of approved general CS electives or a track specific electives list, send an email to petitions@cs with the course you would like added and a brief rationale for why the course deserves to be on the electives list. Your petition will be reviewed by the appropriate academic council members and will get back to you typically within a few weeks. Classes that are numbered below 100 or above 400 are not usually approved as electives. The 193's are also unlikely to be approved as electives.

Senior Project / Writing In the Major

There are several options available for completing the senior project requirement. The senior project class, CS194, is the most common way. In this class, students work in teams to build a significant programming application, learning and practicing useful strategies for software design and engineering. At the end of the class, students demo their projects in front of the other students, course staff, and representatives from many local companies. CS294 fulfills the WIM requirement.

A second popular way to fulfill the senior project requirement is to take CS210, Project Experience with Corporate Partners. This is a two-quarter sequence where students work on loosely defined challenges defined by corporate partners R&D labs. Each team functions as a small startup company with dedicated space, a discretionary budget, and a technical advisory board comprised of the instructional staff. The two-quarter sequence affords time to experiment, benchmark and try out different ideas before zeroing in on a particular direction. The two-quarter time frame also means provides enough time to build something of lasting impact. This class is also a great opportunity to develop relationships with a company, and learn skills that are helpful for managing a team or starting a company.

Another option for senior project is to undertake individual research with a faculty member under the auspices of CS191 or CS191W. CS191W is the writing intensive version that involves

several writing assignments and fulfills the WIM requirement. Doing a research based senior project can provide additional valuable experience for students interested in a research career. Taking CS191 requires finding a faculty member who will oversee the project. See the research page for more information on how to do this. Students taking CS191 must also fill out a senior project proposal form and get the approval of their advisor and Patrick Young.

Another research-based option to fulfill the senior project requirement is to do a senior honors thesis. Writing an honors thesis includes a year-long research project supervised by a CS faculty member. Applications for the honors program are due by May 1. For more information, see the Honors Page.

A final option for the senior project is CS294, the directed research classes. If you'd like to take part in active research but don't have a specific project in mind or would like some guidance, CS294 might be for you. The class is typically focused on a single large project of the professor's interest. People choose or are assigned parts of the larger project to work on for the quarter. For example, in the past, CS294 students contributed to the award-winning DARPA Grand Challenge robot. The topics for CS294 change each year, though it's frequently been led by an AI professor. Professors may expect students to be comfortable with techniques used in the field. For example, an AI CS294 might expect you to have taken CS221. Like CS191, students can opt to do a writing-intensive version (CS294W) that fulfills the WIM requirement.

Almost everyone fulfills the WIM requirement when they do their senior project. If you took CS201 for your TIS requirement, then you do not need to fulfill WIM with senior project. WIM is the only requirement that can be fulfilled by a class that also fulfills another requirement.

See the senior project page for more details.

Deviations and Petitions

In general, students must follow the program sheets in order to graduate. Students who wish to deviate from the program sheet can petition to ask for exceptions or deviations to the stated requirements.

Undergraduate CS Petitions

For petitions to undergraduate Computer Science requirements (found on the back side of the undergraduate program sheet), students can send an email to the CS Petitions Committee at petitions@cs.stanford.edu with a description of what change you would like to make and why. Changes to the front side of the program sheet (School of Engineering requirements) are handled through a separate process in the School of Engineering.

If you would like more information about the undergraduate petition process, you can contact Meredith Hutchin.

Master's CS Petitions

Master's students get one advisor-approved deviation from their program sheet, which is at the discretion of their advisor. Additional petitions go through Claire Stager.

© Stanford University Computer Science Department
Last updated on Wednesday, 23-Jan-2019 17:26:12 PST.