

CS107 — General Information

Lecturer: Nick Troccoli

E-mail: troccoli@stanford.edu

Office: Gates 193

Office Hours: *see course website*

Lectures: Monday and Friday, 12:30PM–1:50PM in Hewlett 200 (lectures are not recorded this quarter, as CS107 is not on SCPD)

Course Overview

CS107 is the third course in Stanford's introductory programming sequence. The CS106 courses provide you with a solid foundation in programming methodology and abstractions, and CS107 follows on this to build up and expand your breadth and depth of programming experience and techniques. The course will work from the C programming language down to the microprocessor to de-mystify the machine. With a complete understanding of how computer systems execute programs and manipulate data, you will become a more effective programmer, especially in dealing with issues of debugging, performance, portability, and robustness. Topics covered include: the C programming language, data representation, machine-level code, computer arithmetic, elements of code compilation, optimization of memory and runtime performance, and memory organization and management.

CS 107A: CS107A, also called Pathfinders (or ACE), is a supplementary instruction program that meets for a weekly section and holds Pathfinders-specific review sessions. CS107A is application-only; please see the FAQ on the course website for more information.

Prerequisites

The prerequisite for CS107 is CS106B/X (or equivalent). You should have practical C/C++ skills using recursion, dynamic data structures (pointers, linked lists, trees, graphs), data abstraction, classic data structures (lists, stacks, queues, sets, maps), and standard algorithms (searching, sorting, hashing). You should have an appreciation of the intrinsic value of good engineering and design and you will be expected to produce well-decomposed, readable code. Come talk with us if you need help determining the right placement for you.

Course Goals

The goals for CS107 are for students to gain **mastery** of

- writing C programs with complex use of memory and pointers

- an accurate model of the address space and compile/runtime behavior of C programs

to achieve **competence** in

- translating C to/from assembly
- writing programs that respect the limitations of computer arithmetic
- identifying bottlenecks and improving runtime performance
- writing code that correctly ports to other architectures
- working effectively in a Unix development environment

and have **exposure** to

- a working understanding of the basics of computer architecture

Course Environment

In CS107, there will be significant programming assignments and you can expect to work hard and be challenged by this course. Your effort can really pay off - once you master the machine and advance your programming skills to the next level, you will have powerful mojo to bring to any future project! To make that happen for everyone, we strive to create an inclusive and equitable class. We further depend on you to help each other obtain excellence rather than mistaking Stanford or this class as zero-sum. We ask that you do your part by seeking to promote the success of others, and by treating each other in ways that respect and celebrate the diversity of talent that is drawn to our exciting field of Computer Science. Here are several aspects of our policy for creating an inclusive and equitable class:

Preparation: Perhaps more than many other subjects, students come to computer science with greatly varying previous exposure to the subject. Regardless of your background, as long as you meet the course prerequisites, you are well-prepared to succeed in CS107. Just ask the students who are now your TAs!

Classroom Environment: We are happy to answer questions you have through various course support channels (office hours, Piazza, etc.). However, during lectures, we request that any questions you ask be clarifications for yourself or others of the course material being discussed, rather than a question or comment intended to demonstrate additional knowledge beyond the scope of the current topic or the course, which may intimidate or discourage other students. If you do have such questions or comments, we are more than happy to discuss them during office hours, or before/after any lecture!

Office Of Accessible Education Accommodations: We are happy to provide accommodations recommended by the OAE to ensure you can be comfortable in the course. Students who may need an academic accommodation based on the impact of a disability must initiate the request with the Office of Accessible Education (OAE). Professional staff will evaluate the request with required documentation, recommend reasonable accommodations, and prepare an Accommodation Letter for faculty. Unless the student has a temporary disability, accommodation letters are issued for the entire academic year. Students should

contact the OAE as soon as possible since timely notice is needed to coordinate accommodations. The OAE is located at 563 Salvatierra Walk (phone: 723-1066, URL: <https://oae.stanford.edu>).

Preferred Name and Gender Pronouns: We want you to be *you* in our class. You are always welcome to write your preferred name on all classwork and exams. If you have a name and/or pronoun that doesn't match our registrar-provided class roster, please let us know and we will ensure that we use it in our class.

Course Expenses: If obtaining any material or resource for use in our class presents a financial hardship for you, please let us know and we will work with you to accommodate.

Feedback: Please do not hesitate to reach out to the course staff or the instructor, anonymously if you prefer, if any aspect of our course or community can be improved.

Units

If you are a matriculated Stanford graduate student, you may enroll in CS107 for 3-5 units based on your schedule. Otherwise, you are required to enroll in CS107 for 5 units. Taking the course for reduced units does not alter the course requirements.

Class Website

The website for CS107 is located at <https://cs107.stanford.edu>.

Please check the course website regularly as we will post important announcements there, as well as the course schedule, lecture materials, handouts, assignments, and more. This handout serves to provide important highlights, but be sure to check the website for more details.

Textbook

Bryant & O'Hallaron. *Computer Systems: A Programmer's Perspective (3rd Edition)*

The bookstore has a less-expensive custom version of this textbook for our course that includes only the chapters we will cover; you can also use the regular full 3rd edition, which will come in handy if you plan to continue on to CS110 (which uses the rest of the book). You will need the 3rd edition of the textbook, which has substantial updates from IA32 to x86-64. There will be assigned readings from this textbook that are important preparation for lecture and lab. We also strongly recommend you have a "C language goto" in whatever form works best for you: textbook, tutorial, reference sheet, website, etc. As one suggestion, *The C Programming Language* by Kernighan and Ritchie is the classic text and a digital copy is available to Stanford students via [Safari Books Online](#).

Grading

Final grades for the course will be determined using the following weights:

15%	Lab Participation
35%	Assignments
17%	Midterm Exam (2/15, 12:30-2:20PM)
33%	Final Exam (3/22, 12:15-3:15PM)

To pass the course, both the coursework (assignments and labs) and exam (midterm and final) aggregates must each be passing work. Restated, if the composite of your exams is failing, then you will fail the class even with outstanding performance on coursework, or vice versa.

Programming Assignments

There will be seven programming assignments throughout the quarter, all of which are to be done **individually**. Assignments are due at **11:59PM sharp** on the dates specified on the course website, and are written and submitted electronically using the **myth** cluster of Unix workstations in the Gates Computer Science Building. These machines are accessible remotely from other machines such as your personal computer, and they are pre-installed with all software and tools used for labs and assignments.

The course staff grades each assignment on *functionality* (is the program's behavior correct from an external perspective?) and *style* ("code review" - is the code clean, well-written and elegant?), and returns your grades to you electronically via email and via the course website. Functionality is measured by how successfully the program executes on a comprehensive set of test cases. We create our test suite by working from the original program specification to identify a list of expected behaviors and write a test case for each. We use the autotester to run a submission on each test and award points for each successful result. Thus, the resulting functionality score is a direct reflection of how much observably correct behavior your program exhibited. This process is completely automated; the grader does not search your code to find bugs and deduct for them, nor do they appraise the code and award points for tasks that are attempted or close to correct.

Style is measured via several quality metrics (such as building cleanly, cleanly running under Valgrind, etc.) as well as a code review by the course staff. Your style score is mapped to the following scale; from past experience, most grades will be **ok**.

- + An outstanding job; reflects code that is notably clean, elegant and readable, and could be used as course example code for good style.
- ok A good job; reflects code that demonstrates solid effort and is fairly successful at meeting expectations, but also has opportunities for improvement.
- Has larger problems, but shows some effort and understanding. There were either large concerns, or a multitude of smaller concerns, in the submission.
- - Shows little effort and does not represent passing work. The work is incomplete or quite non-functional.

- 0 No work submitted, or barely any changes from the starter assignment.

Using these categories means that the course staff can focus on the assignment's learning goals rather than spending time justifying each point. Our goal is to maximize the learning experience in doing the assignments, and we have found the "bucket" grading system to work much better for style feedback than assigning numeric grades from a pedagogical perspective over many quarters of experience.

Disputes about homework grading must be submitted to the course staff within 1 week of receiving your grade. For more information about the assignment grading process, please see the assignments page on the course website.

Getting Help

We want to enable everyone to succeed in this course and offer several paths to help:

Course policy questions: This handout and the course website are the primary resources for details on course policies. If you can't find an answer to your question, feel free to search or post on our public Piazza discussion forum.

Help with assignments or course topics: Feel free to search or post on Piazza or ask questions during lab or office hours. For private matters (e.g. something about your code, or a question about your grade), you can email the course staff at `cs107@cs.stanford.edu`.

Sensitive or confidential topics: Feel free to email the instructor privately or stop by their office hours.

The course website also houses useful documents such as how-to guides for the tools, and advice pages. Please take advantage of these resources!

Late Policy

Each assignment writeup will indicate the deadline for on-time submission. The date and time of the deadline are firm. Submitting even a minute past is considered late. **Submitting by the deadline earns an extra-credit on-time bonus, typically a ~5% perk.** The assignment writeup will specify the bonus if available.

If you miss the deadline, there may be a grace period for late submissions. The assignment writeup will specify if there is a grace period and for how long. You can make a late submission after the deadline and up to the end of the grace period. **These late submissions are not penalized, but do not earn the on-time bonus.** The grace period is typically 48 hours past the deadline, but may be abbreviated, or even none, on a per-assignment basis.

Submissions are not accepted beyond the grace period. The grace period is strictly enforced.

We recommend thinking of any grace period as a backup in case something unexpected comes up when aiming for the deadline. As a result, getting an extension beyond the provided grace period will generally not be granted. In *very special* circumstances (primarily extended medical problems or other emergencies), extensions may be granted beyond the provided grace period, but this is very rare. **Only the instructor will be able to approve extensions.** In particular, do not ask the CAs. All extension requests must be received in advance of the deadline, and you must make at least an initial submission within the grace period in any case, which can be replaced later if the request is granted.

Labs

In addition to lecture, you must also sign up for a weekly 1 hour 50 minute hands-on lab led by one of our CAs in Gates B08. In lab, students work in pairs on guided exercises. Part of your course grade comes from attending, participating in and completing the work in your lab each week. **You may miss one lab without penalty, although attending all labs will earn a small extra-credit bonus.**

You must sign up for a lab time using a link posted on the main course webpage **starting on Wednesday, January 9th at 6PM.** Note that signups **are first-come-first-serve**, so it's best to move quickly once signups open if you don't have much schedule flexibility. Labs begin **the second week of classes.** Note that you **must** sign up for labs via the course website (you do not sign up for labs on Axess).

Examinations

The midterm examination is scheduled for the full class period on **Friday, February 15th from 12:30PM-2:20PM.** The final examination is scheduled for **Friday, March 22nd from 12:15PM-3:15PM.** **There will be no alternate time for the midterm and final exams,** except for those with relevant official university athletics or Office of Accessible Education accommodations. Please make sure that you can attend the exams at the specified times before enrolling in the class. Also note that University policy prohibits students from enrolling in courses with conflicting lecture times or conflicting final exams.

All examinations are **closed-book** and **closed-electronic-device.** For the midterm exam, you will be allowed to bring in one double-sided 8.5x11" page of notes, and for the final exam, you will be allowed to bring in two double-sided 8.5x11" pages of notes. A reference sheet will be provided to you during each exam, as well as beforehand for studying, containing commonly-needed information. No other materials may be used.

The Honor Code

Academic conduct for students at Stanford is governed by the Honor Code. Part of the Honor Code is a pledge and expectation to participate in class without seeking inappropriate help on graded work such as assignments and exams. Please read the **separate Honor Code page** linked to on the main course website (<https://cs107.stanford.edu/collaboration.html>); you are responsible for knowing all of the details at this link, and for following the Honor Code in this course.

Violations of the Honor Code are taken very seriously; we reserve the right to use software tools to compare your submissions against those of all other current and past students, and will refer all suspected violations to the Office of Community Standards.