General Information and Syllabus

Based on a document by Julie Zelenski, Cynthia Lee, and others
Last updated June 24th, 2023

Instructors: Joel Ramirez (joel101@stanford.edu) & Adam Keppler (akeppler@stanford.edu).
Office Hours on **office hours calendar (/class/archive/cs/cs107/cs107.1238/help#calendar)**
All course staff: See **front page (/class/archive/cs/cs107/cs107.1238/index#course-staff)**

Course Essentials

- The **class website** for CS107 is located at **https://cs107.stanford.edu (https://cs107.stanford.edu)**. Please regularly check the course website as we will post important announcements there, as well as the course schedule, lecture materials, handouts, assignments, and more.
- We will have lecture on **Mondays, Wednesdays and Fridays, 3:00M - 4:15PM** in NVIDIA Auditorium.
- We will be using **Canvas** to store lecture recordings and access Zoom links for remote office hours, and to record lecture participation. Access our **Canvas course (https://canvas.stanford.edu/courses/175785)**.
- We will be using **Poll Everywhere** to record lecture participation and do in-class polls. Access Poll Everywhere with your Stanford email **here (https://pollev.stanford.edu)**.
- The **discussion forum on Ed (/class/archive/cs/cs107/cs107.1238/help#discussion-forum)** is the place for students to ask questions and discuss course topics with peers and staff. Visit **office hours (/class/archive/cs/cs107/cs107.1238/help#office-hours)** if you have more specific debugging or conceptual questions.
- The `myth` servers are our remote workspaces for completing programming assignments. Read more in our **getting started guide (/class/archive/cs/cs107/cs107.1238/getting-started.html)**

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, memory, 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.

Prerequisites

The prerequisite for CS107 is CS106B (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.

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.

Guiding Principles

We have designed the course to the best of our ability to provide flexibility. There are campus resources, such as **accommodations (https://oae.stanford.edu/students)**, **undergraduate advising directors (https://advising.stanford.edu/appointments)**, **well-being coaches (https://vaden.stanford.edu/well-being/coaching)**, **counselors (https://vaden.stanford.edu/home)**, **academic coaches (http://learningconnection.stanford.edu/academicskills)**, **Hume Center writing tutors (https://undergrad.stanford.edu/tutoring-support/hume-center)**, and the **FLI opportunity fund (https://fli.stanford.edu/students/current-students/financial-support)** for broader needs you might have. For information about specific policies and procedures for the quarter, **Student Affairs (https://reapproaching.stanford.edu/)** has the most up-to-date information for students. If there are additional ways we can support you in the course, please feel encouraged to reach out to us. Without requesting or expecting details of your situation, we will do everything we can to ensure your course learning is productive and enjoyable.

Course Accommodations

If you are ever sick, in COVID-19 isolation, encounter an emergency, or other exceptional circumstance, we have a variety of accommodation mechanisms for each course component, and we encourage you to reach out to the instructor, Head TA or TAs to let us know how we can help! The accommodation mechanisms include:

- late days on assignments (see **assignment late policy**)
- attending a makeup lab or getting an excused lab absence (see **labs**)
- lecture excused absences (see **lectures**)
- accommodations for exams due to illness, COVID-19 isolation, or other emergency (see **exams**)
- students who are ill, in COVID-19 isolation, have an emergency, or other exceptional circumstances may temporarily attend all office hours remotely with Head TA permission (see **Getting Help**)

If you feel ill or are sick, for the wellbeing of yourself and others please stay home, take care of yourself, take advantage of these accommodations, and reach out to us - we never want you to feel that you must attend class if you are not feeling well! Similarly, if you have an emergency or exceptional circumstance, please take advantage of these accommodations and reach out to us so that we can help.

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 of C programs

- strong understanding of the 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
- working effectively in a Unix development environment
- writing code that correctly ports to other architectures

and have **exposure** to

- a working understanding of the basics of computer architecture
- understanding compilers and disassemblers
- understanding the semantics of assembly with respect to stack layout

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 will really pay off - as you master the concept and advance your programming skills to the next level, you will be able to apply those skills to future projects! 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, discussion forum, 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!
- **Access and Accommodations**: Stanford is committed to providing equal educational opportunities for disabled students. Disabled students are a valued and essential part of the Stanford community. We welcome you to our class. If you experience disability, please register with the Office of Accessible Education (OAE). Professional staff will evaluate your needs, support appropriate and reasonable accommodations, and prepare an Academic Accommodation Letter for faculty. To get started, or to re-initiate services, please visit oae.stanford.edu. If you already have an Academic Accommodation Letter, we invite you to share your letter with us. Academic Accommodation Letters should be shared at the earliest possible opportunity so we may partner with you and OAE to identify any barriers to access and inclusion that might be encountered in your experience of this course.
- **Name and Pronouns**: We want you to be you in our class. You are always welcome to put your preferred name on all classwork and exams (just make sure to let us know, so we ensure you receive credit). 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 TAs, Head TA or the instructor, anonymously if you prefer (there is an **anonymous feedback link (/class/archive/cs/cs107/cs107.1238/index.html#feedback)** on the main page of the course website), if any aspect of our course or community can be improved.

Course Structure

Course Grades

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

- **50%** Assignments
- **15%** Lab Participation
- **10%** Lecture Participation
- **10%** Heap Allocator(Final Project)
- **15%** Midterm

Final course grades will be calculated by determining the raw overall score, and then breaking the score distribution into different letter grades. We do not decide these letter grade cutoff points until the end of the quarter, based on the course distribution - for that reason, we cannot guarantee in advance that a particular raw overall score will translate to a particular letter grade.

If you choose to take the course CR/NC, your final grade must be satisfactory or better--in other words, *you need a C- or better to get a CR.*

Lecture

Lectures are held in person on Mondays, Wednesdays and Fridays. Lecture attendance/participation is recorded via **Poll Everywhere (https://ctl.stanford.edu/use-learning-technology/live-polling)**. Students who submit answers to all questions during a lecture (regardless of response correctness) will receive credit for attending that lecture. If you are ill, notify the course staff to have the absence excused. Please do not come to class if you are sick. Please do not hesitate to reach out to the course staff or the instructor if any personal circumstances or issues arise!

Unexcused absences will count against your grade.

Labs

In addition to lecture, you must also sign up for a weekly hands-on in-person lab led by one of our CAs. During the lab, students will work in small groups on guided exercises. Labs encourage participatory collaborative learning - the goal is for everyone to come away with a clear understanding of the lab concepts which will complement the assignments.

Part of your course grade comes from arriving on time, attending, participating in and completing the work in your lab each week. Please meake sure that you are checked off by a member of the course staff before leaving the lab to ensure that your lab participation / attendance grade is recorded. Arriving late or leaving early without prior consent will result in not receiving lab credit.

If you have an extenuating circumstance (illness, COVID-19 isolation, emergency, etc.) and cannot make any labs during a given week, please email the course staff for accommodations. Otherwise, if you do have to miss attending the lab entirely, we strongly encourage you to work through the exercises for your own benefit, but there is no makeup credit for missed participation.

Assignments

There will be seven assignments throughout the quarter, all of which are to be done ***individually***. Assignments are due at ***11:59pm PT sharp*** on the dates specified 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 used for labs and assignments. Students edit, compile, and debug on the `myth` systems using a suite of open source development tools including `gcc`, `make`, `gdb`, and `valgrind`.

Assignments may consist of written portions and coding portions, and are graded and returned to you electronically via the course website. The course staff grades each programming portion on **functionality** (is the program's behavior correct from an external perspective?) and **style** ("code review" - is the code clean, well-written and elegant?). 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. There may also be some limited points awarded manually by TAs for other functionality aspects not covered in the automated test cases.

The total number of points for an assignment is noted in its writeup. The points indicate the weight of that assignment relative to the others; i.e. an assignment graded out of 100 points has twice the weight of one graded out of 50.

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. The **style guide (/class/archive/cs/cs107/cs107.1238/styleguide)** outlines further guidelines. 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 many significant issues and does not represent passing work.
- `0` No work submitted, or barely any changes from the starter assignment.

Additionally, for each assignment outstanding student submissions will be chosen to receive an award and presented at the beginning of a lecture. Awards will typically be made for outstanding style fundamentals, exceptional architecture or design, creativity of solution. The goal of the awards is to encourage you to submit your best work and allow the class to learn from your thoughtful implementations.

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 grades being released. For more information about the assignment grading process, please see our page on **how assignments are graded (assignment-grading.html)**.

Assignment Late Policy

Every student begins the quarter with **five free "late days,"** to be used only for assignments. Each late day allows you to submit an assignment up to one calendar day late without penalty. For example, if a due date is Wednesday at 11:59PM PT, using 1 late day allows you to submit until that Thursday at 11:59PM PT without penalty, and 2 late days allows you to submit until that Friday at 11:59PM PT without penalty. Late days may only be used in 24-hour increments. **We will not accept assignments more than 2 days late and no late days may be used for the last assignment.**

After the late days are exhausted, submissions that come in late will be assessed a late penalty per day late. Specifically, if you submit up to 1 day late, your score will be capped at 80% of possible points. This means that if you have a score above 80%, then your score will be lowered to 80%. If you have a score at or below 80%, your score will remain the same. Similarly, if you submit between 1 and 2 days late, your score will be capped at 60% of possible points.

You should think of free late days as extensions you have been granted ahead of time and use them when you might have otherwise tried to ask for an extension. Beyond these 5 late days, we will grant additional extensions only in cases where exceptional circumstances necessitate more than a 2-day extension, or when you have used your 5 late days and further extenuating circumstances necessitate additional accommodations. As a result, getting an extension for exceptional circumstances beyond the provided free late days must be approved by the Head TA. **Only the Head TA will be able to approve extensions.** *All extension requests must be received in advance of the assignment deadline.* Please do not hesitate to reach out to the course staff or the instructor if any personal circumstances or issues arise!

If you have unused late days at the end of the quarter each of them will count as 1 point in the assignments section added to your existing point allocation as a reward for on-time submission.

Examinations

The midterm examination is in-person and will be administered **during class - see the course landing page for the date**. If you have an academic or University conflict with this time, and absolutely cannot make the regularly scheduled midterm, you must send a request by email to the Head TA and instructors to arrange an alternate exam time. Please include in your email all the possible times you are able to take the exam.

All examinations are administered on paper and are open-book/notes, but no electronics/internet. Any printed material is allowed, but no electronic devices are permitted. Our exams will remain unproctored.

If you become ill, must enter COVID-19 isolation, or have another emergency around the time of or during an exam, please contact the course staff as soon as possible for accommodations.

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 (/class/archive/cs/cs107/cs107.1238/collaboration)**; 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.

Course Resources

Textbook(s)

Required: 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. 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. For some of the readings that we assign, you can view a free digital PDF copy of the textbook on Canvas under the "Files" tab.

Strongly recommended: 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 for checkout via **Open Library (https://openlibrary.org/books/OL2030445M/The_C_Programming_Language)** (make a free account to "borrow" it digitally). Another option is Nick Parlante's Essential C reader PDF **available here (http://cslibrary.stanford.edu/101)**.

There is also a CS107 reader: **https://web.stanford.edu/~cgregg/cgi-bin/107-reader (https://web.stanford.edu/~cgregg/cgi-bin/107-reader)**, which covers all of the course topics in detail.

Our lecture readings pull from Bryant & O'Halloran, Kernighan & Ritchie, and Essential C.

Getting Help

We want to enable everyone to succeed in this course and offer several help resources. Read **more about getting help in CS107 (/class/archive/cs/cs107/cs107.1238/help)**.

- **Discussion Forum**: The online discussion forum lets you search, ask and answer questions posted by fellow classmates. The course staff will also monitor and periodically respond to posts. The discussion forum is best for course policy questions, general course topic questions, general assignment questions or small debugging questions. It should not be used for larger questions about your assignment code.
- **Office Hours**: Office hours are offered throughout the week, and are where you can sign up in a queue and get help from the course staff. Office hours are best for course topic questions, general assignment questions, or in depth questions about your code.
- **Instructor+Head TA Email / Office Hours**: For private matters such as grade questions or other sensitive or confidential topics, please feel free to email the Head TA directly or stop by their office hours to talk privately. You can also email or chat with the instructor for private/personal matters.
- The course website houses various useful documents, such as how-to guides for the tools, and advice pages. Please take advantage of these resources!

Other Information

COVID-19 Precautions

Please check here for the latest on campus health guidance: https://healthalerts.stanford.edu/ . Most importantly, **please do not come to class** (lecture, lab, office hours, etc) if you are sick, you can help prevent the next pandemic!

Planning for the Unexpected

While the world is returning to the new-normal, we are cognizant that unforeseen circumstances and events have defined the 2020s. While we don't foresee a need to massively adjust the course plan, we will remain flexible and appreciate your patience with unforeseen changes.

Lecture Video Notice

Video cameras located in the back of the room will capture the instructor presentations in this course. For your convenience, you can access these recordings by logging into the course Canvas site. These recordings might be reused in other Stanford courses, viewed by other Stanford students, faculty, or staff, or used for other education and research purposes. Note that while the cameras are positioned with the intention of recording only the instructor, occasionally a part of your image or voice might be incidentally captured particularly if you ask questions in class. If you have questions, please contact a member of the teaching team.
