Nick Troccoli
CS107

Handout #1
September 14, 2020

CS107 — General Information

Lecturer: Nick Troccoli
E-mail: `troccoli@stanford.edu`
Helper Hours: *see course website*

Lectures: Monday and Friday, 11:30AM–12:50PM PDT via Zoom. Lecture videos are available via Canvas.

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.

CS 107A: CS107A is part of ACE (Additional Calculus for Engineers), a supplementary instruction program that includes weekly sections, office hours, and ACE-specific review sessions. CS107A is done in addition to all the normal requirements for CS107. CS107A is application-only; please see the FAQ page 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.

Guiding Principles for Teaching and Learning in a Time of Pandemic

(by Prof. Brandon Bayne, UNC – Chapel Hill; via CS103; with light adjustments)

1. Nobody signed up for this.
    - Not for the sickness, not for the social distancing, not for the sudden end of our collective lives together on campus
    - Not for an online class, not for teaching remotely, not for learning from home, not for mastering new technologies, not for varied access to learning materials
2. The humane option is the best option.
    - We are going to prioritize supporting each other as humans
    - We are going to prioritize simplicity
    - We are going to prioritize sharing resources and communicating clearly
3. We will foster intellectual nourishment, social connection, and personal accommodation.
    - Accessible asynchronous content for diverse access, time zones, and contexts
    - Optional synchronous lecture review sessions to learn together and combat isolation
    - Please do not hestitate to reach out to the course staff or the instructor if any personal circumstances or issues arise.
4. We will remain flexible and adjust to the situation.
    - Nobody knows where this is going and what we'll need to adapt
    - Everybody needs support and understanding in this unprecedented moment
    - We will constantly evaluate and listen to ensure the class is going as smoothly as possible.

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
- working effectively in a Unix development environment

and have **exposure** to

- a working understanding of the basics of computer architecture

Course Environment

(by Cynthia Lee, with light adjustments)

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 (helper 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 helper 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 put 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. Please see the note at the end of this handout for more information.

Feedback: Please do not hesitate to reach out to the course staff or the instructor, anonymously if you prefer (there is an anonymous feedback link on the main page of the course website), 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.

Canvas

All enrolled students should have access to CS107's page on Canvas (**`https://canvas.stanford.edu`**). Canvas will be used to access course lecture videos, Zoom links and lecture check-in quizzes.

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 the Stanford Libraries (see the "Available Online" section on that page).

Grading

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

45%	Assignments
10%	Mid-quarter Assessment
20%	Final Project
15%	Lab Participation
10%	Lecture Check-in Quizzes

If you choose to take the course **S/NC**, your final grade is computed the same as it would be for a letter grade; in other words, **you need a C- or better to get an S**.

Assignments

There will be roughly six assignments throughout the quarter, all of which are to be done **individually**. Assignments are due at **11:59PM PDT 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 used for labs and assignments.

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.

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 on the assignments page of the course website 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.

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.

Final Project

The course final project has you implement your own "heap allocator" – a fundamental component of many of the programs we will write this quarter. The final project will be graded using the same functionality/code review process as for assignments but acts as a capstone project that sums up topics from throughout the quarter. **You must do the final project in order to pass the class.**

Getting Help

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

Online 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, or general assignment questions. It should **not** be used for specific questions about your assignment code.

Helper Hours: Helper hours are offered throughout the week, and are where you can sign up in a queue and get help from the course staff. Helper hours are best for course topic questions, general assignment questions, or questions about your code.

Course Staff Email: You can email the course staff at `cs107@cs.stanford.edu`. Please do not email course staff members individually. Emailing the course staff is best for private matters such as grade questions or personal accommodations. It should **not** be used to ask questions about course content or your assignment code.

Instructor Email / Helper Hours: For other sensitive or confidential topics, please feel free to email the instructor directly or stop by their helper hours to talk privately.

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

Every student begins the quarter with **five free "late days."** 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 PST, using 1 late day allows you to submit until that Thursday at 11:59PM PST without penalty, and 2 late days allows you to submit until that Friday at 11:59PM PST without penalty. **We will not accept assignments more than 2 days late and no late days may be used for the final project**.

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. As a result, getting an extension for exceptional circumstances beyond the provided free late days must be approved by the instructor. **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 assignment deadline. Please do not hesitate to reach out to the course staff or the instructor if any personal circumstances or issues arise**!**

Labs

In addition to lecture, you must also sign up for a weekly 1-hour-30-minute hands-on lab led by one of our CAs live via videochat. In lab, students work in small groups on guided exercises. Part of your course grade comes from attending, participating in and completing the work in your lab each week. Labs will also have short *pre-lab* exercises to act as a warm-up to that week's activities.

You must submit your lab time preferences using a link posted on the main course webpage **starting on Tuesday, September 15th at 5PM PDT.** Note that signups **are not first-come-first-serve**; you may submit your preferences any time up to **Saturday, September 19th at 5PM PDT**. 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).

Lecture Check-In Quizzes

The main lecture material is pre-recorded as a collection of shorter videos that are posted on Canvas in advance of that actual lecture day. Each video will have an associated "lecture check-in quiz" – a very short understanding check that will permit multiple attempts. You will have until the date of the live lecture session to do these quizzes. The sum of lecture quizzes for each day is weighted equally (in other words, all quizzes combined for lecture 2 are weighted the same as all quizzes combined for lecture 3). Live lecture sessions will be much shorter than the full class period, and will be devoted to reviewing concepts further, answering questions, and doing additional exercises.

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.

Course Expenses

All students should retain receipts for books and other course-related expenses, as these may be qualified educational expenses for tax purposes. If you are an undergraduate receiving financial aid, you may be eligible for additional financial aid for required books and course materials if these expenses exceed the aid amount in your award letter. For more information, review your award letter or visit the Student Budget website (`https://financialaid.stanford.edu/undergrad/budget/index.html`).

Video Notice

Lecture sessions will be recorded on video and posted to the course Canvas site only. Lab sessions and Helper Hours sessions will not be recorded. If you have questions, please contact a member of the teaching team.

This document is copyright © Nick Troccoli, and based on similar handouts from Julie Zelenski, Michael Chang, Cynthia Lee, Chris Gregg, Alisha Adam, Rohit Talreja, Mehran Sahami, Chris Piech, Marty Stepp, Jerry Cain and others. Licensed under Creative Commons Attribution 2.5 License. All rights reserved.