CS107
M Chang & J Zelenski

Handout #1
April 2, 2018

CS107 — Course Overview

The class

CS107 is the third course in Stanford's introductory programming sequence. Our CS106 courses provide a solid foundation in programming methodology and abstractions and CS107 follows on to develop programming maturity and expand breadth and depth of experience with computer systems. The course will work from the C programming language down to the CPU to de-mystify the machine. With an 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, performance optimization, and memory organization and management.

The class has two lectures a week and a weekly lab for hands-on learning and experimentation. There will be significant programming assignments and you can expect to work hard and be challenged. Your effort will pay off— by mastering the machine and raising your programming skills to the next level, you will have powerful mojo to bring to any future project!

The student

The prerequisite is successful completion of CS106B/X (or equivalent) and eagerness to advance to the next level. You should be an accomplished programmer who has practical C/C++ skills in using recursion, pointers and dynamic memory management, classic data structures (lists, trees, graphs, 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. You're welcome to talk with us if you need further help determining the right placement for you.

The word on the street from a CS107 alum:

> "You know how Chinese restaurants have an asterisk next to some items on the menu? The asterisk warns the customer that the item is really hot and spicy. I think these asterisks are great, because sometimes the names can be ambiguous and it's a drag to get a really spicy plate when you didn't expect it. I think the Course Bulletin should have this feature. Some of the classes (ME101, CS107, etc.) would have an asterisk, and at the bottom of the page would be a warning the class is pretty spicy, so you know what you're getting into!"

Web site

If there is only one thing to take away from this handout, it's the URL for our course web site

http://cs107.stanford.edu

This is your go-to place for all things CS107— announcements, course materials, syllabus, assigned readings, office hour schedule, discussion forum, and more. This rest of this handout serves to hit a few highlights; refer our web site for more detail.

Instructors:

Michael Chang
`mchang91@stanford.edu`
Office: Gates 203

Julie Zelenski
`zelenski@stanford.edu`
Office: Gates 188

Lectures: MF 12:30-1:50pm in NVidia Auditorium

Lab: Various times TWTh in Gates B08. The hands-on lab component is our favorite part of CS107. All students participate in a weekly 2-hour lab section, working in pairs on guided exercises. During the first week, we will post the lab schedule on the course web site and you will be able to signup for a lab from the available options. Lab signup is **first-come first-served**; best to move quickly once signups open if you don't have much schedule flexibility. Local SCPD students are encouraged to attend a regular on-campus lab if possible. This provides the best participatory experience and we think it's worth your effort to join us in person. We will prepare a webcast summary for SCPD students who are following along remotely.

Readings: The required text is *Computer Systems: A Programmer's Perspective*, Bryant & O'Hallaron (**3rd edition**). The bookstore stocks a custom reader with just the chapters used for CS107. The B&O text is also used by CS110 (different set of chapters); you can instead consider buying the entire textbook if planning ahead. We also 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 of K&R is available via Stanford's subscription to Safari Books Online.

Software: Our home base is the shared cluster of unix workstations (`myth`). All software and tools used for labs and assignments is pre-installed on these hosts.

Exams: The midterm will be in-class Friday May 11th. The final will be Wednesday June 13th 12:15-3:15 pm. Before committing to the course, verify that you can attend both the midterm and final exams as there will be **no alternate/make-up exams**. SCPD students from the Bay Area will come to campus for the exams. Remote students take exams on-site.

Grading: Course grades will be determined by 50% coursework (programming assignments and labs) and 50% exams (midterm and final). To pass the course, both the coursework and exam aggregates must each be passing work. Restated, if the composite of your exams is failing, then you will fail the class in spite of an astounding performance on coursework or vice versa.

Getting help: We want to enable everyone to succeed in this course and offer several different paths to help. We run a discussion forum on Piazza for all public Q&A. Email to `cs107@cs.stanford.edu` can be used for issues of a private nature. The course staff holds a large number of office hours scheduled throughout the week. The course web site houses useful documents such as how-to guides for the tools (gdb, Valgrind, and so on) and advice pages. Please take advantage of these resources!

Honor Code: The Stanford Honor Code is an undertaking of the students and faculty to individually and collectively participate in upholding the highest standards for academic integrity. Please **review the full collaboration policy on our site** (`http://cs107.stanford.edu/collaboration.html`) for a thorough explanation of how it applies in this course. Your programs should be your own original, independent effort and must not be based on, guided by, or jointly developed with the work of others.

The vast majority of you are here to learn and will do honest work for an honest grade. We celebrate and honor your commitment. Because it's important that all cases of academic dishonesty are identified for the sake of those playing by the rules, 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 cases of concern to the Office of Community Standards. While we certainly don't want to create a Big Brother environment, we do need to be clear how far we'll go to make sure the consistently honest feel their honesty is valued.

Disabilities: Students who need an academic accommodation based on the impact of a disability should initiate a request with the Student Disability Resource Center (SDRC) in the Office of Accessible Education. SDRC staff will evaluate the request with required documentation, recommend reasonable accommodations, and prepare an Accommodation Letter dated in the current quarter. Students should contact the SDRC as soon as possible since timely notice is needed to coordinate accommodations. SDRC 563 Salvatierra Walk (phone: 723-1066).

Syllabus: A rough outline to give you an idea of what is planned—expect minor adjustments as we go. Programming assignments will be assigned ~weekly.

Week	Topics
1	Admin, UNIX environment, bits/bitwise ops
2	Integer representation, computer arithmetic, C pointers/arrays
3	C-strings, C stdlib, dynamic allocation
4	C generics, void *, function pointers
5	Floating point representation, intro to assembly
6	x86-64: addressing, ALU ops **Midterm**: **Fri May 11th in-class**
7	x86-64: control, function calls, runtime stack
8	Address space, dynamic memory management
9	Performance, case study
10	Wrap/review
11	**Final**: **Wed June 13th 12:15-3:15pm**

Class quote: “*The motto of enlightenment is: Sapere aude! Have courage to use your own understanding!”* (Immanuel Kant, 1784). The transformation in understanding and confidence we can achieve together in 10 weeks is nothing short of amazing. Bring your curiosity and perseverance and join us the noble pursuit of enlightenment in the way of computer systems! ☺

This tag cannot be removed under penalty of law except by the consumer. Do not eat this paper. Recycle it.