CS110 Course Information

Instructor: Jerry Cain

Email: jerry@cs.stanford.edu

Phone: 415-205-2242 (between 7:00am and 9:00pm)

Office: Gates 192

Office hours: Tuesdays, Thursdays, 1:00pm - 4:00pm (starting January 13th)

Lectures: MWF 10:00am – 10:50am, Building 420, Room 040

Units: 5 units. Only matriculated graduate students with RA- or CA-ships can register for fewer than five. The course requirements are the same regardless of the number of units you take the class for.

Course Assistants: Rebekah Bartlett, Chang Xing (Raven) Jiang, Daniel Jih, Sneha Kudli, Kai-Yuan (Kai) Neo, Arpan Shah, and David Wang. CS110 CA's attend lectures, hold office hours, evaluate program submissions, and grade exams. Be glad they're here, because virtually all of them have either taken or CA'ed CS110 before, and they know the material so well they already know what your questions are going to be.

Prerequisites: The formal prerequisites for the course are CS106B, CS107 and CS103. Practically speaking, you need to be familiar with the C and C++ programming language, Unix/Linux, `make`, `Makefiles`, `gcc/g++`, `valgrind`, `gdb` and have some experience with basic computer architecture (x86 as it's taught in CS107, or exposure to some other ISA with the confidence and ability to pick up x86 as I reference it).

We'll initially code in C, but we'll eventually migrate to a modern dialect of C++. You should understand pointers, dynamic memory allocation (`malloc/realloc/free`), and C strings to start. You should also understand C++ classes, methods, references, templates, and C++'s `new` and `delete` operators. There are some features of the C++11 version of the language that you're not expected to know, but you should have enough programming maturity to pick those features up and search the web for reference materials as needed.

Readings:

- The first required textbook is Computer Systems: A Programmer's Perspective by Bryant and O'Hallaron, 2nd edition. Both CS107 and CS110 teach from a subset of the B&O textbook, so I've arranged for a custom version with just the chapters we need to be sold at the Stanford Bookstore. Of course, if you want to purchase the entire textbook, you're free to do that as well, though you'll need to buy from Amazon or some other online retailer.
- The second required textbook is Principles of Computer System Design: An Introduction by Jerome H. Saltzer and M. Frans Kaashoek. Stanford has university-wide digital access to the textbook, and because it's free, I assume most of you would just prefer to go with the free online version. Of course, you're welcome to purchase a hardcopy if you'd like. It's available for purchase on Amazon.

Website: `http://cs110.stanford.edu` is your favorite new website. There you'll find all reading assignments, lecture slides, homework assignments, and the full list of office hours. If you have any suggestions on how to make the course website even better, feel free to send Jerry an email.

Software: The shared UNIX workstations (`myth` machines in Gates B08 and Gates B30, available via `ssh`) provide all of the development tools needed for lecture examples and the six or so assignments (although I may occasionally reference the more powerful `corn` and `barley` machines to clarify the impact that more cores and larger caches have on execution).

I recognize we live in a laptop world, and that you'd prefer to code on your own machines and eventually port it over to the `myth`s (where you submit your work and we grade your assignments). However, I urge you to code, test, and debug directly on the `myth`s via `ssh`. The `myth`s are outfitted with a hip version of `g++` that supports the advanced C++11 features we'll eventually be relying on for better `string` support, type-safe template data structures, and object-oriented concurrency libraries.

Class email: There's a class mailing list I'll use for important announcements that just can't wait until lecture. All CS110 students are automatically subscribed to the `cs110-win1415-students@lists.stanford.edu` mailing list. The list server is in touch with Axess and automatically includes everyone enrolled. Make it a point to register for CS110 if you haven't already, as I tend to make use of the mailing list more during the first two weeks of the course than any other time.

Forums: We are using Piazza for class discussion forums (and a direct link is available on the CS110 website). When you have a question that might be of interest to other stu-

dents, please post it here for a speedy response. Please note that you should never include snippets of code directly from your own homework submissions, since that's sharing assignment code with others and a huge no-no.

Staff email: When you have a question for the course staff that shouldn't be shared with students, please send that question to `cs110@cs.stanford.edu`, where we maintain a shared question queue. We prefer this to emailing the staff members individually, as it gets you the quickest response. It also allows us to track which questions are being asked repeatedly so we can post an edited version of your question to the forums for everyone's benefit.

Grading: You can take the course for a letter grade, or as **CR/NC**. The course grading is divided between several programming assignments, a midterm, and a final exam. The grade breakdown is:

- Programming assignments: 45%
- Midterm Exam: 20%
- Final Exam: 35%

To receive a passing grade, you must pass **both** of my exams. Restated, if you fail either exam, that exam effectively counts 100% instead of 25%. (I will be clear what a passing grade is for the midterm so you can withdraw from the course before the deadline.)

Midterm Exam: The two-hour midterm is scheduled for Thursday, February 12th at 7:00pm in a location to be determined. The midterm is **closed book, closed notes, and closed computer**, save for the fact that I'll allow you to prepare and refer to a single 8.5"-by-11" cheat sheet containing any information you can cram onto each of its two sides. I will include all relevant prototypes and type definitions (C functions, C++ classes, etc.) on the exam, and you're welcome to ask a staff member for a function or method prototype if I don't include it but you'd like to use it.

If you have a legitimate conflict with the 7:00pm - 9:00pm time slot, you're welcome to take the midterm earlier in the day during any two-hour window that fits entirely within the 10:00am - 5:00pm window. However, you must inform me via email that you need to take the exam earlier in the day, and you must do so by February 5th.

Final Exam: The three-hour final exam is scheduled for Thursday, March 19th at 8:30am. I'm not giving an alternate final, since this time slot is set aside for classes that meet MWF at 10:00am. CS110 isn't on SCPD this quarter, but if you're leveraging the SCPD system and taking a second class at the same time, then you are responsible for

ensuring the second class's final exam doesn't conflict with mine.

The final is also **closed book, closed notes, and closed computer**, but I'll allow you to bring and refer to the **four sides** of the **two** 8.5"-by-11" cheat sheets you prepare ahead of time. As with the midterm, I will include all relevant prototypes and type definitions (C functions, C++ classes, etc.,) and you're welcome to ask a staff member for any function or method prototype.

Late Policy: The class material builds on itself and getting behind is taboo, as it tends to impede progress on subsequent assignments. Late submissions make us grumpy, so any assignment turned in late will be assessed a penalty of 10% per day (24-hour period). Assignments will not be accepted more than **two** days after the original assignment due date. The CAs are as busy as you are, and it's unfair to require they grade a second wave of late assignments when I'm pressuring them to grade the on-time ones as quickly as possible. (Note: **You may not use any late days on the first assignment**, because we want to grade them as quickly as possible so you have a sense of how your submitted work is evaluated.)

All this said, we all have our emergencies. I suffer my own personal crises from time to time, so I expect you do too. Instead of demanding that you ask for special consideration, I will give each of you the privilege of granting yourself small extensions whenever crises arise. You get **three** free late days (24-hour periods) that you may use to extend the due dates of any assignment (except for the first) without penalty. You may use one or two of your free late days for any particular assignment. If an assignment is due on Wednesday night, then using one late day extends the deadline to Thursday night, and using two late days extends the deadline to Friday night. If an assignment is due on Friday night, then using one late day extends the deadline to Saturday night, and using two late days extends the deadline to Sunday night.

Understand that requests for additional late days are consistently denied unless all three grace days were used for the very best of reasons. If you feel your situation is so extenuating that any reasonable person would agree, then you should contact me to see about getting penalty-free extensions. (Once all of your grace late days are gone, you can still hand work in late, but the 10% penalty kicks in.)

Disabilities: Students who need an academic accommodation based on the impact of a disability must initiate the request with the Student Disability Resource Center (SDRC) located within the Office of Accessible Education (OAE). SDRC staff will evaluate the request with required documentation, recommend reasonable accommodations, and prepare an Accommodation Letter for faculty dated in the current quarter in which the re-

quest is being made. Students should contact the SDRC as soon as possible since timely notice is needed to coordinate accommodations. The OAE is located at 563 Salvatierra Walk, and their phone number is 650-723-1066.

Honor Code: Although you are encouraged to discuss ideas with others, your programs are to be completed independently and should represent original work. Whenever you obtain help (from current or previous CS110 students, the CA's, students in other classes, etc.) you should credit those who helped you directly in your program, e.g. in a program comment, type "The idea to use a `mutex`-guarded linked list of file descriptors came from a discussion with my CS110 CA, Rebekah Bartlett".

Any assistance that is not given proper citation is considered a violation of the Stanford Honor Code. To be even more specific, you are not allowed to collaborate on the coding of your programs, nor are you allowed to copy programs or even minute snippets of programs from other students, past or present. The following four activities are among the many I consider to be Honor Code violations:

1. Looking at another student's code.
2. Showing another student your code.
3. Discussing assignments in such detail that you duplicate a portion of someone else's code in your own program.
4. Uploading your code to a public repository (e.g. `github.com` or `bitbucket.com`) so that others can easily discover it via word of mouth or search engines. If you'd like to upload your code to a private repository, you can do so on `bitbucket` or some other hosting service that provides free-of-charge private hosting.

Unfortunately, our department sees more than its share of Honor Code violations. Because it's important that all cases of academic dishonesty be identified for the sake of those playing by the rules, we exercise our right to use software tools to compare your submissions against those of all other CS110 students, past and present.

I'm not trying to create some Big Brother environment. I'm just being clear about how far I'll go to make sure the consistently honest feel their honesty is valued. If the thought of copying code has never crossed your mind, you needn't worry, because I've never seen a false accusation make its way through Community Standards. If, however, you're ever tempted to submit unoriginal work—whether it's because you don't understand the material, or you do understand but you're short on time—then you need to remember this paragraph is here.