CMSC 15200: Introduction to Computer Science II

The University of Chicago, Winter 2017
Adam Shaw

`https://www.classes.cs.uchicago.edu/archive/2017/winter/15200-1`

Welcome! In CS152, we extend our introduction to major computer science topics through instruction in imperative computer programming and various analytical techniques.

CS152 is designed for students intending to major or minor in the subject, although others are welcome. To enroll in CS152, students must have previously taken one of the department's first quarter courses.

The specific goals of the course are these:

- to learn imperative programming techniques,

- to build an understanding of the memory usage of computer programs, including an awareness of the stack/heap distinction, fluency with pointers and dynamic memory allocation and deallocation,

- to design a variety of useful data structures including lists, trees, maps, and graphs, and

- to analyze the efficiency of certain algorithms.

These goals will be reached in part by reexamination of techniques learned in the previous quarter in a new light. Broader, more technical treatments of these topics, in particular algorithm analysis, are presented in later undergraduate courses.

We use the C programming language in our studies. C is the *lingua franca* of computer programmers, and one of the historic and ongoing successes in the design of practical programming languages.

Having learned a second programming language well, and having viewed a variety of common problems from two distinct vantage points, students who complete CS152 will begin to see past the superficial characteristics of computer programs to appreciate their deeper properties.

Online Help: Piazza

We do not use Chalk in CS152. We will use piazza instead. We know most of you are familiar with piazza already, but please ask your instructor for help getting started with it as needed.

Instructor

Adam Shaw, email: `ams@cs.uchicago.edu`, office: Eckhart 128.

Graduate Teaching Assistants

Jiajun Shen, Nedelina Teneva

Contacting Us

If you have questions about the course, and those questions are in a sense impersonal — that is, they are about course material or course logistics — we ask that you post those questions publicly on piazza, rather than contacting any of the staff members directly. This ensures you will receive the fastest, most consistent possible response from the staff. Since students usually have *common* questions, posting public questions is also very efficient for your classmates as well. As yet another advantage, it avoids duplication of work on the part of the staff.

In cases where you have a question that is about your own personal situation and not relevant to the class as a whole, you may ask a "private question" on piazza, which is invisible to your classmates, or send email to your instructor directly.

Lectures

- Section 1 (Shaw): MWF 9:30–11:20. Stuart 105

The first lecture is on Wednesday, January 4; the last is on Wednesday, March 8. (There are two other sections of CS152 this quarter; they are taught by Diana Franklin and have their own materials. Those lectures, exams and exercises are not interchangeable with section 1.)

We do not allow the use of electronic devices during lectures. They are simply too distracting. This includes laptops, smartphones, and tablets. The lone exception to this policy is for students whose handwriting issues necessitate their use of a device for note taking, who will be permitted to use a plain text editor on a laptop whose wireless capability is turned off. If you are such a student, let your instructor know.

Lab Sessions Students must register for and attend lab sessions each week. Lab sessions are held in the Computer Science Instructional Laboratory (also known as the CSIL); it is located on the first floor of Crerar Library. Attendance

at the lab session for which you are registered is mandatory.

We offer four weekly lab sections at four different meeting times. You will work on a department's Macintosh or Linux computer, depending on the day, during your lab session. You must use the department's computer during lab and may not use your own laptop.

The lab times are as follows:

Tues 3:00pm–4:20pm, Tues 4:30pm–5:50pm, Wed 2:30–3:50pm, Wed 4:00pm–5:20pm

Schedule of Topics by Week (subject to change)

Week	Topics
1	the shell, functions
2	conditionals, operators, recursion, iteration
3	arrays, pointers
4	allocation, memory management, I/O
5	lists, higher-order programming
6	unions, variants, bits and bytes
7	sorting, efficiency, trees
8	tree balancing, graphs
9	graphs, games
10	special topics: data structures and algorithms

Office Hours To be announced on the web once the quarter starts. In addition to the office hours we provide ourselves, the College Core Tutor Program (aka "Harper Tutors") employs computer science tutors Sunday through Thursday nights from 7pm–11pm, starting in the second week.

Texts (required) *The C Programming Language (Second Edition)*, Kernighan, Ritchie. The textbook is available on campus at the Seminary Co-op Bookstore[1]; you can of course find new and used copies at your favorite online bookstore as well.

Software All the software we use in this course is available free of charge for all common platforms. We will mainly use *emacs*, *clang* and *subversion*. Windows users will need to download and install *Cygwin*, and will be able to include subversion in their Cygwin installations.

Grading Coursework is comprised of lab exercises (done at lab sessions, discussed above), homework assignments, projects, and exams.

---

[1] 5751 S. Woodlawn Avenue; http://www.semcoop.com.

Each student's final grade will be computed according to the following formula: homework and project work 30%, labs 20%, exams 25% each. We will scale the grades, so what precisely constitutes an A, B, *etc.* will be determined by the collective performance of the class. We announce in advance that a 92 guarantees at least an A- for the course; an 82 guarantees a B-; a 72 guarantees a C-; a 65 guarantees a D. The scale may shift downward from this somewhat, but the boundaries given here are fixed from the beginning.

Homework There will be weekly homework assignments. These will generally be due on Mondays.

Projects There will be a longer multipart project during the latter part of the term. It is important to write relatively large programs and not just little ones, because as the scale of the work grows, good code-writing habits become not just a matter of style and taste but positively essential.

Exams There will be two exams for all students (all three sections), tentatively scheduled at the following dates and times: Wednesday, February 1st, 7pm–9pm and Monday, March 6, 7pm–9pm. Please note that we have not chosen locations for these tests. Changes to this exam schedule will be announced conspicuously and well in advance on piazza.

We expect you to be available at both exam times unless you have another obligation that is inflexible and important. Both exams are technically *midterms* although the second one is effectively a final. There is no separate final exam during finals week.

Regrade Requests Graders, TAs, and instructors do make mistakes, so it is normal and expected that a few assignments will have grading errors. In order to manage the workload of regrade requests, however, *regrade requests must be submitted within 5 weekdays of the assignment being returned.*

Late Work Deadlines in this course are rigid. Since you submit your work electronically, deadlines are enforced to the minute. Late work will not be counted. In order to allow for some irregularities in your own schedule, we will drop the lowest homework grade (note: the dropped grade is specifically a homework grade, not a lab grade or project grade).

We will also late work in the case of special circumstances, when those circumstances are extraordinary.

Academic Honesty In this course, as in all your courses, you must adhere to college-wide honesty guidelines as set forth at `http://college.uchicago.edu/policies-regulations/academic-integrity-student-conduct`. The college's rules have the final say in all cases. Our own paraphrase is as follows:

1. Never copy work from any other source and submit it as your own.

2. Never allow your work to be copied or seen by another student.

3. Never submit work identical to another student's.

4. Never look at someone's working solution in order to solve your error.

5. Document all collaboration.

6. Cite your sources.

We are serious about enforcing academic honesty. If you break any of these rules, you will face tough consequences. Please note that sharing your work publicly (such as posting it to the web) definitely breaks the second rule. With respect to the third rule, you may discuss the general strategy of how to solve a particular problem with another student (in which case, you must document it per the fourth rule), but you may not share your work directly, and when it comes time to sit down and start typing, you must do the work by yourself. If you ever have any questions or concerns about honesty issues, raise them with your instructor, early.

Starting this quarter, we will enforce the following rule as well: any student who is determined to have violated academic honesty rules will not be allowed to withdraw from this course.

Advice Writing code that does what it is supposed to do can be joyful, even exhilarating. By contrast, fighting for hours with broken code is misery. We would like you to help you experience more of the former and less of the latter. Work methodically. Start your work well ahead of time. Beyond a certain point, it is not profitable to be stumped. If you have made no progress in some nontrivial chunk of time, say, one hour, it is time to stop and change your approach. Use one of our many support mechanisms to get some assistance (office hours, Piazza). We will help you get going again when you are stuck. We are familiar with common mistakes that stump students for hours but require only minutes for us to identify. Please take advantage of our experience.

2017 January 17 3:30pm. This is revision 0 of this document.

2017 January 25 5:30pm. This is revision 1, with corrected exam dates.