CMSC 151: Introduction to Computer Science I

The University of Chicago, Fall 2012, Adam Shaw

`http://www.classes.cs.uchicago.edu/archive/2012/fall/15100-1`

Welcome! In CS151, we introduce a selection of major computer science topics by way of computer programming and various analytical techniques. The course is designed for students intending to major or minor in the subject.

The specific goals of the course are these:

- to design data structures to solve specific computational problems,
- to process those data structures in several ways, most importantly by means of structural recursion,
- to learn to recognize and exploit common computational patterns,
- to become conversant with parametric polymorphism and higher-order programming, and
- to analyze the efficiency of computer programs.

In making progress towards these goals, students will become acquainted with a selection of classic data structures and algorithms. We use the Racket programming language as our platform.

Having completed the course, students will know how to use computer programming as a flexible, reliable, efficient and comprehensive method for analytical problem solving and creative endeavors. Students will discover, in future work, that the experience gained in this course applies to programming generally, in any programming language. Furthermore, they will have a wider acquaintance with best practices in the discipline.

Things You Must Do More or Less Immediately

- **Request a CS account no later than Monday, October 1.** Having a CS account confers upon you the ability to use CS department machines, supplies you a home directory securely accessible from anywhere on campus, and indeed from anywhere on the Internet, and various other perquisites. You request an account by filling out the web form at the following location:

  `https://www.cs.uchicago.edu/info/services/account_request`

- Register with piazza. You will receive an email, with instructions, at your uchicago email address about this, so make sure you check that email on October 1.

Instructor Adam Shaw, email: `ams@`[1], office: Ryerson 157.

Graduate Teaching Assistants Negar Mirsattari, Anna Olson, Sneha Popley, Nick Seltzer (labs); Nikita Mishra, Nedelina Teneva (homework).

Lectures All lectures are in Ryerson 251. There are two identical sections.

- Section 1: MWF 9:30–10:20.
- Section 2: MWF 11:30–12:20.

The first meeting is on Monday, October 1; the last meeting is on Wednesday, December 5. I do not allow the use of electronic devices during my lectures.

Lab Sessions Students must register for and attend lab sessions each week. Lab sessions are held in the Computer Science Instructional Laboratory (informally known as the MacLab), on level A of the Regenstein Library. Attendance at the lab session for which you are registered is mandatory.

We offer twelve lab sections at eight different times. During the Wednesday slots, two labs occur side by side at the same location, so those times accommodate two sections each. You will work on either a department Macintosh or Linux computer during your lab session, depending on which section you are in. You must use the department's computer during lab and may not use your own laptop.

The lab times are as follows:

> Tues 9am–10:20am; Tues 12pm–1:20pm; Tues 3pm–4:20pm; Tues 4:30pm–5:50pm; Wed 12:30pm–1:50pm; Wed 2pm–3:20pm; Wed 3:30pm–4:50pm; Wed 5pm–6:20pm.

There will be no lab sessions during the eighth week (Thanksgiving), and no lab sessions the last week (before reading period).

Office Hours To be announced on the course website.

Text *How to Design Programs*, Felleisen *et al.*, ISBN 0-262-06218-6. The textbook is available on campus at the Seminary Co-op Bookstore[2]; you can of course find new and used copies at your favorite online bookstore as well. Before buying a copy, note that the full text of the book is available online at `http://www.htdp.org` free of charge.

Software *DrRacket*, available at `http://racket-lang.org`, and *subversion* (details TBA).

[1] `cs.uchicago.edu`
[2] 5757 S. University Ave., `http://www.semcoop.com`.

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

Homework and Projects There will be homework assignments, roughly weekly. These will be augmented by longer project-style assignments spanning several weeks.

Exams There will be a 50 minute midterm exam in class, and a two hour final exam. For students in section 1, the final exam is Monday, December 10, 8am–10am; for section 2, it is Friday, December 14, 10:30am–12:30pm.

The final grade will be computed according to the following formula: homework and projects 30%, labs 20%, midterm exam 20%, final exam 30%. I will grade on a curve, so what constitutes an A, B, *etc.* will be determined by the best scores in the group.

Lateness Deadlines in this course are rigid. Since you submit your work electronically, deadlines are enforced to the minute. Late work will not be counted, with the following exception. You have one 24-hour extension on any lab or homework assignment (except the first), no questions asked. Note the 24-hour extension may not be used on the first homework or lab exercise. We will let you know the details of how to request an extension during the quarter.

(We will also accept late work in the case of special circumstances, when those circumstances are truly extraordinary. Having a lot of other work to do is definitely not an extraordinary circumstance.)

Schedule of Topics by Week (subject to change)

Week	**Topics**
1	expressions, functions
2	structures, variants, lexical scope
3	linked lists, structural recursion
4	higher-order programming
5	trees
6	accumulators, mutual recursion
7	efficiency, sorting
8	state
9	hash tables, maps
10	special topics TBD

Advice Writing code that does what it is supposed to do can be joyful, even exhilarating. By contrast, wrangling with broken code engenders precisely the opposite sensations. Work methodically. Start your work well ahead of time. Beyond a certain point, it is not profitable to be stumped. Ask us for assistance. We will help you get going again if you find yourself unable to make progress.

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. My 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.
3. Never submit work identical to another student's.
4. Document all collaboration.
5. Cite your sources.

We are absolutely serious about enforcing academic honesty. If you break any of these rules, you will face dire 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 should document it per the second rule), but you may not share your work directly, and when it comes time to sit down and type, you must do the work yourself. If you ever have any questions or concerns about honesty issues, raise them with your instructor, early.

2012 September 26 9:51am This is revision 0 of this document.