

# CMSC 15100: Introduction to Computer Science I

The University of Chicago, Autumn 2016

Adam Shaw and Matthew Wachs

<http://www.classes.cs.uchicago.edu/archive/2016/fall/15100-1>

---

**Welcome!** In CMSC 15100 (informally *CS151*), we introduce a selection of major computer science topics through instruction in computer programming and related analytical techniques.

CS151 is designed for students intending to major or minor in the subject, although others are welcome.

The specific goals of the course are these:

- to understand solving computational problems in terms of identifying, and, when necessary, designing, relevant abstractions,
- to process data structures in several ways, most importantly by the technique of structural recursion,
- to learn to recognize and exploit common computational patterns through code organization and higher-order programming,
- to learn to use simple and polymorphic types as a powerful approximation of correctness in computer programs, and
- to analyze the efficiency of certain algorithms.

In pursuing these goals, students will become acquainted with a selection of classic data structures and algorithms. Broader, more technical treatments of these topics, in particular algorithm analysis, are presented in later undergraduate courses.

We use the Racket programming language in our studies. Racket is a dialect of Scheme, a language with a long history in the field of computer science generally and college-level instruction specifically. This year, for the third time, we will use the Typed Racket variant of Racket, for reasons we will articulate during the quarter.

Having completed this course, students will be able to use computer programming as a robust and efficient method for analytical problem solving and creative endeavors, and develop a sense of the relationship between computer programming and computer science. Students will discover, in future work, that the experience gained in this course applies to programming generally, in any programming language; that is, CS151 must not be thought of as a course in Scheme

programming. Furthermore, students will have gained experience with certain best practices in the discipline, including practical ones.

### **Piazza: Online Support**

Register with *piazza*. Piazza is an online question-and-answer system that we use for that purpose as well as distribution of some course materials. You will receive an email about piazza registration, with instructions, at your *uchicago* email address at the start of the quarter, so make sure you check that email address by September 26.

### **Instructors**

Adam Shaw, email: [ams@cs.uchicago.edu](mailto:ams@cs.uchicago.edu), office: Eckhart 128.

Matthew Wachs, email: [mwachs@cs.uchicago.edu](mailto:mwachs@cs.uchicago.edu), office: Eckhart 127.

### **Graduate Teaching Assistants**

You teaching assistants this quarter are Yan Liu, Hannah Morgan, Nedelina Teneva, and Liwen Zhang.

You will interact with your TAs in labs and through piazza, but you will not need to email them directly. Please see the next section.

### **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 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.

A few piazza rules:

- Do not post any more than a snippet of code to piazza. Any post that says “Here’s my hundred lines of code – what’s wrong with it?” will be deleted immediately.
- Please do not post anonymously to piazza. Piazza posts are better, more thoughtfully written, and more courteous when the author is identified. Furthermore, we are committed to building a community where questions are asked openly and enjoy respect from all constituencies — fellow stu-

dents and teachers alike. We reserve the right to delete anonymous posts from piazza.

**Lectures** There are three sections of CS151 this quarter, at the following times and places.

- Section 1: MWF 9:30–10:20, Stuart 102, Adam Shaw.
- Section 2: MWF 11:30–12:20, Ryerson 276, Adam Shaw.
- Section 3: MWF 1:30–2:20, Kent 107, Matthew Wachs.

The first lecture is on Monday, September 26; the last is on Wednesday, November 30.

**Devices** We do not allow the use of electronic devices during lectures. They are simply too distracting. This includes laptops, smartphones, tablets, and anything that beeps, blinks, records, locomotes, *etc.* The lone exception to this policy is that students whose handwriting issues necessitate their use of a device for note taking will be permitted to use a plain text editor on a laptop or tablet 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, read “*see-sil*”). The CSIL is located on the first floor of Crerar Library. Attendance at the lab session for which you are registered is mandatory.

We offer twelve weekly lab sections at six different meeting times. If you need to switch your lab time, there will be a way to do so online early in the quarter; details to follow. You will work on one of the department’s Macintosh computers 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 2:30pm–3:50pm; Tues 4pm–5:20pm; Tues 5:30pm–6:50pm;  
Wed 2:30pm–3:50pm; Wed 4:00pm–5:20pm; Wed 5:30pm–6:50pm.

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

**Office Hours** The office hours schedule will be announced on the web once the quarter starts. In addition to the office hours we provide, the College Core Tutor Program employs computer science tutors Sunday through Thursday nights from 7pm–11pm, starting in the second week of the quarter. The College Core

tutors are specifically charged with support of CS151 students, and are often alums of the course themselves.

**Schedule of Topics by Week** (subject to change)

Week	Topics
1	expressions, types, functions
2	conditionals, structures, patterns, unions
3	local variables, inductive structures: trees and lists
4	natural numbers; parametric polymorphism, functions as data
5	higher-order programming
6	anonymous functions; images and universes
7	efficiency, binary search trees
8	mutual recursion, sorting
9	state, vectors, mutation
10	project

**Optional Text** *How to Design Programs*, Felleisen *et al.*, ISBN 0-262-06218-6. Over the years and as our curriculum has evolved, we have departed from this text to such an extent that we no longer require you to buy a copy. Moreover, the full text of the book is available online at <http://www.htdp.org> free of charge. It is a useful reference and a good book in its own right, and, despite various differences with this course in its present form, has given our curriculum its shape and thematic content.

**Software** All the software we use in this course is available free of charge for all common platforms. We will mainly use *DrRacket*, available at <http://racket-lang.org>, and *subversion*. Macintosh and Linux users very likely have subversion on their machines already. Windows users will need to download and install *Cygwin*, and will be able to include subversion in their Cygwin installations. We will provide detailed installation instructions for various platforms once the quarter starts.

**Grading** Coursework is comprised of lab exercises (done at lab sessions, discussed above), homework assignments, projects, and exams. Each student's final grade will be computed according to the following formula: lab exercises 15%, homework 20%, project work 15%, exams 25% each. What precisely constitutes an A, B, *etc.* will be determined by the collective performance of the class. Having said that, earning 92% assures at least an A-; 82% a B-; 72% a C-; 62% assures a D. We reserve the right to adjust that scale, but we will not adjust any of those thresholds *upward*.

We know your grades are important to you. Nevertheless, point lawyering is not appreciated. When a grader has deducted points on one of your pieces of work, view it foremost as an act of mentorship and an opportunity for improvement.

**Homework** There will be weekly homework assignments. These will be assigned on Monday or Tuesday (usually) and will be due the following Monday (usually).

**Projects** There will be a longer multipart project during the latter part of the term. The final part of this project will be due during exam week. The theme of this year's project is chess. Please don't be put off if you are not a chess player: you won't need to be able to play chess at all well, you only need to know the rules of the game.

**Exams** There will be two exams for all students that will meet outside of class.

- midterm exam: Monday, October 24, 7pm–9pm, and
- final exam: Monday, November 28, 7pm–9pm.

Please plan accordingly, which is to say, make every effort not to plan anything else for either of those times. Both exams are planned to be held in Kent 107, although this location is not confirmed at the time of this writing.

**Exam Accommodations** If you are a student who has special exam-taking arrangements with Student Disabilities Services (SDS), you must contact SDS to arrange for a time, date and proctor for the exams in this course. We will do our part by delivering the exams to the SDS office as needed.

If you do not have any arrangements with SDS but believe you should, please contact them directly (<https://disabilities.uchicago.edu/>) or talk to your college adviser to open the conversation.

To be clear, we, your instructors, are not in a position to judge who needs what sort of special accommodations; we leave those judgments to professionals. SDS, along with the college administration, makes these determinations, and we abide them. Therefore, although you may ask one of us about making an accommodation for an exam, we will simply refer to your adviser or SDS in such cases.

**Late Work** Deadlines in this course are rigid. Since you submit your work electronically, deadlines are enforced to the minute. Late work will not earn credit. However, to allow for whatever inevitable deadline-related difficulties arise for individuals throughout the quarter, we will drop the lowest scoring homework exercise from our final calculation. Please note that no project work will be dropped, and no labs: only one homework.

(We will occasionally accept 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.
3. Never submit work identical to another student's.
4. Document all collaboration.
5. Cite your sources.

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.

**Advice** Writing code that does what it is supposed to do can be enlivening, joyful, even uplifting. By contrast, fighting for hours with broken code is the opposite: discouraging, frustrating, and generally miserable. We would like you to experience more of the former and less of the latter.<sup>1</sup> 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. We will help you get going again when you are stuck.

---

*2016 Sept 22 4:00pm.* This is revision 0 of this document.

---

<sup>1</sup>In particular, we never want you to come to us and say “I spent six hours fighting with subversion.” Do that under no circumstances whatsoever. This campus is positively crawling with people who know how to negotiate version control systems. Get help.