# CMSC 152: Introduction to Computer Science II

The University of Chicago, Winter 2015

Adam Shaw and Matthew Wachs

http://www.classes.cs.uchicago.edu/archive/2014/winter/15200-1

Welcome! In CS152, we introduce a selection of major computer science topics through instruction in computer programming and various analytical techniques. More specifically, we learn the craft of classical imperative programming using the C programming language.

CS152 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 learn to write algorithms and build data structures in an imperative, side-effect rich style,
- to become well-acquainted with the C programming language, which is a *lingua franca* in the computing professions,
- to learn the low-level implementations of a wide variety of data structures, including arrays, linked lists, hash tables, and trees,
- to use, design and create command-line tools,
- to learn to reason about memory usage and manage memory directly, and
- to analyze the efficiency of computer programs.

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.

## Piazza: Online Support

Piazza is an online question-and-answer system that we use for that purpose as well as distribution of course materials on occasion. You will receive an email about piazza registration, with instructions, at your uchicago email address at the start of the quarter.

### Instructors

Adam Shaw, email: ams@cs.uchicago.edu, office: Ryerson 157.

Matthew Wachs, email: mwachs@cs.uchicago.edu, office: Ryerson 175A.

### **Teaching Assistants and Graders**

Our graduate student teaching assistants this quarter are Connor Imes, Haopeng Liu, Nikita Mishra, Nick Seltzer, Severin Thaler, and Zhixuan Zhou. Connor, Nick, and Severin will be running labs.

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

Our graders this quarter are Jonathan Jin, Marc Kealhofer, Mark Maskeri, Benjamin Rohrer, Abraham Secular, and Alice Zhang.

#### 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 piaza, 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.

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. We reserve the right to delete anonymous posts from piazza.
- Do not post questions of the form "Will we lose points if...?" Ask about correctness, efficiency, elegance, but not points.

Lectures There are three sections of CS152 this quarter, at the following times and places.

- Section 1: MWF 9:30–10:20, Harper 140, Adam Shaw.
- Section 2: MWF 11:30–12:20, Ryerson 251, Adam Shaw.
- Section 3: MWF 1:30–2:20, Ryerson 251, Matthew Wachs.

The first lectures are on Monday, January 5; the last are on Wednesday, March 11.

We do not allow the use of electronic devices during lectures. The devices 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). There is a new CSIL facility as of this quarter; it 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 eight different meeting times. During the four Wednesday slots, two labs occur side by side in adjacent lab quadrants. 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 a department's Macintosh computer 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 12pm–1:20pm; Tues 1:30pm–2:50pm; Tues 3pm–4:20pm; Tues 4:30pm–5:50pm; 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** To be announced on the web once the quarter starts. In addition to the office hours we provide ourselves, the College Core Tutor Program employs computer science tutors Sunday through Thursday nights from 7pm–11pm, starting in the second week of the quarter.

**Text** The C Programming Language (Second Edition), Kernighan and Ritchie (this book is informally known as "K&R"). The textbook is available on campus

at the Seminary Co-op Bookstore<sup>1</sup>; you can of course find new and used copies at your favorite online bookstore as well.

**Software** We will mainly use *emacs*, *gcc* and *subversion*. Everyone should also go ahead and install *ssh* on their local machine(s) as needed. All C code written for this course must compile under gcc 4.6.3 without extensions (such as nested functions, a Boolean type, *etc.*). This version of gcc is the standard installation on the CS department machines at /usr/bin. Windows users are encouraged to download and install Cygwin (if they haven't already done so) along with these tools.

Schedule of Topics by Week (subject to change)

Week	Topics
1	the command line, binary numbers, types, conditionals
2	operators, functions, iteration, pointers
3	heap allocation, arrays, strings
4	structs, tagged unions, linked lists
5	more lists, trees
6	more trees, multidimensional arrays
7	hash tables, stacks, queues
8	graph representations, graph algorithms
9	more graph algorithms, in-place sorting, efficiency
10	tree balancing

**Grading** Coursework is comprised of lab exercises (done at lab sessions, discussed above), homework assignments, projects, and exams. The relative weighting of these in computing your grade is below.

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

**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.

Exams There will be two exams for all students that will meet in Kent 107.

- midterm exam: Wednesday, February 4, 7pm-9pm, and
- final exam: Wednesday, March 11, 7pm-9pm.

Please plan accordingly. Students with standing exam accommodations with the college must work with SDS to arrange for alternate times and places. Students without standing accommodations who have major conflicts must make

<sup>&</sup>lt;sup>1</sup>5751 S. Woodlawn Avenue; http://www.semcoop.com.

alternate arrangements with their instructor at least one week before the date of the exam in question.

Each student's final grade will be computed according to the following formula: lab exercises 15%, homework 20%, project work 15%, exams 25% each.

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 lab exercise from our final calculation, and we will drop the lowest scoring homework exercise as well. Please note that no project work will be dropped: only one lab and one homework.

(We will 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.

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.

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 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. We will help you get going again when you are stuck.