

MPCS 50101: Concepts of Programming

The University of Chicago, Summer 2017, Adam Shaw

<http://people.cs.uchicago.edu/~adamshaw/mpcs50101-2017>

Welcome! In MPCS 50101, we provide an immersive introduction to computer programming for students who may not have had any prior programming experience. Although we use the Java programming language, the course emphasizes fundamental concepts in computer programming, in favor of the quirks of Java. The main distinctive feature of Java that finds its way into our curriculum is its emphasis on object-oriented software design. Therefore we characterize this course as a broad introduction to general-purpose computer programming and algorithmic reasoning in an object-oriented context.

The specific goals of the course are these:

- to learn to construct algorithms to solve common problems,
- to learn to express algorithms precisely as computer programs,
- to learn to refine and improve programs by an iterative process, including by identifying errors carefully and fixing them,
- to learn to identify when primitive data types are appropriate to a task, and when compound data is needed,
- to develop a notion (as opposed to a theory; that comes later) of computational efficiency,
- to learn about data structures for collections to the greatest possible extent, including but not limited to arrays, lists, stacks, queues, maps, sets, trees, and hash tables.

As additional goals, students will begin to learn to view a computer as a programmer does, becoming acquainted with common tools like the command-line terminal and emacs, and will also learn some of the language computer programmers and computer scientists use to talk about problems and their solutions. In this course alone, mastery should not be expected, although we intend to put you on the path to mastery as best we can in the available time.

It is expected that students will go on to write programs in other languages and in other paradigms later in their careers. Correspondingly, the course is designed to provide broad rather than narrow preparation.

Instructor

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

Something You Must Do More or Less Immediately

- Register with *piazza*. 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, so make sure you check that email address by June 19. Piazza will be the main source of the most up-to-date information about this course.

Graduate Teaching Assistants

We have (two or three) graduate student teaching assistants joining me to instruct you this summer: [to be named later]. The TAs will be available for office hours outside of class, at times to be designated later.

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.

Class Meetings We will meet on Mondays and Wednesdays from 6pm–8pm from Monday June 19 to Wednesday August 9, with the exception of a holiday break on Monday July 3. Class meetings are either in Ryerson 251 or in Crerar library, as per the meeting schedule posted at the course website.

I discourage the use of electronic devices during lectures. They are distracting. This includes laptops, smartphones, and tablets. I strongly recommend you take notes by hand, because I believe (and research supports¹) that it improves the quality of your learning.

¹see “The Pen is Mightier than the Keyboard” by Mueller and Oppenheimer, *Psychological Science*, April 2014.

Schedule of Topics by Week (subject to change)

Week	Topics
1	programmers' tools, primitive data, operators, expressions
2	objects, classes, methods, interfaces, tests
3	conditionals, recursion, iteration, Strings, arrays, collections
4	images, enumerated types, class equality, I/O, exceptions, errors
5	midterm exam; searching, sorting, efficiency
6	lists, trees, generics
7	sets, maps
8	hash tables, final exam

Office Hours We will announce the full schedule of office hours once the course starts and we have had a chance to find out what times are convenient for the group.

Text *Introduction to Programming in Java*, Sedgewick, Wayne. The textbook is available on campus at the Seminary Co-op Bookstore²; 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*, *javac* and *git*. Windows users will need to download and install *Cygwin*.

Grading Coursework is comprised of homework exercises and exams. Your score will be computed according to the following formula: homework exercises 40%, midterm exam 30%, final exam 30%. The lowest homework score will be dropped for each student. We will scale the grades, so what precisely constitutes an A, B, *etc.* will be determined by the collective performance of the class.

Homework There will be weekly homework assignments. These will be assigned shortly after class and due before the next one (the exact time is to be announced).

Exams There will be a midterm exam for all students tentatively scheduled at the following time: Monday, July 25, 6pm. Plan accordingly. The final exam will be on Wednesday, August 9 at 6pm. Both exams will be held in Ryerson 251.

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, but the lowest homework grade will be dropped from the final reckoning (providing *de facto* amnesty against one missed deadline).

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

²5751 S. Woodlawn Avenue; <http://www.semcoop.com>.

Academic Honesty In this course, as in all your courses, you must adhere to honesty guidelines as set forth at <http://college.uchicago.edu/policies-regulations/academic-integrity-student-conduct>. The University'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.

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

2017 April 21. Revision 0 of this document.