CMSC 152: Introduction to Computer Science 2

The University of Chicago, Summer 2014, Lamont Samuels

`http://www.classes.cs.uchicago.edu/archive/2014/summer/15200-91/`

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

The specific goals of the course are these:

- to comprehend the basic principles of algorithmic problem solving.
- to use basic control constructs and data types to solve problems.
- to learn to refine and improve programs by an iterative process, using the C programming language, including by identifying errors carefully and fixing them,
- to develop a beginning notion of computational efficiency

Tasks to do ImmediatelyCS Account

- **If you do not have a CS account, request a CS account no later than Wednesday, July 30.** Having a CS account allows you 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`

Instructor:

Lamont Samuels, email: `lamonts@cs.uchicago.edu`, office: Ryerson 176.
Office Hours: Tuesday 1PM-3PM and Friday 11AM-12PM in my office

Mark Maskeri's Office Hours: Wednesday 11:30AM-1PM in CSIL

Teaching Assistants:

Mark Maskeri, Tristan Rasmussen and Sammi Wrescher

Rather than contacting the TAs with questions by email, please use the collective question-and- answer system at

`http://piazza.com`.

CS Account:

- **If you do not have a CS account, request a CS account no later than Wednesday, July 30.** Having a CS account allows you 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:

Lectures: All lectures are in Ryerson 251. Lecture meets Monday, Wednesday, and Friday from 1:30-3:30pm. The lectures will be split into two parts with an intermediate break between. Please take the time to get up grab a drink of water and stretch your legs. Two hours can be a very long time sitting still.

The first meeting is on Monday, July 28; the last meeting is on Friday, August 29.

Labs: Labs are **mandatory**. Weekly lab sessions are from 4PM-6PM on Wednesdays. Labs are held in the Computer Science Instruction Lab (CSIL), in the John Crerar Library.

Text: There are two recommended texts for C. They are both useful but not required.

- C Programming, A Modern Approach, by K.N. King.

- The C Programming Language by Brian Kernighan and Dennis Ritchie.

These books are available for purchase from the Seminary Co-op Bookstore.

Your are not strictly required to purchase the books, as the lecture notes should be enough to get you past the homework and the exam. However, we strongly recommend that you buy the books if you intend to continue working with C, as it can be an invaluable reference (plus the source of many interesting exercises during the course itself).

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

Evaluation Coursework is split into homework exercises, labs, and exams. You score will be computed according to the following formula: homework exercises 35%, quizzes 25%, final exam 30% and labs 10%. We will scale the grades, so what precisely constitutes an A, B, *etc.* will be determined by the collective performance of the class.

Final Exam The final exam will be on Friday, August 28. Both exams will be held in Ryerson 251 (our regular classroom) at our normal time (1:30PM-3:30PM).

Late Work Late work will not be accepted. Our time frame is compressed and you must keep up. Exceptions to this policy will be granted only in the case of extraordinary circumstances.

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.

Acknowledgements: A large amount of the material I use in this course was generously supplied to me from previous instructors (and in particular Adam Shaw, and Matthew Rocklin). Thank you for graciously helping me with this course.