

Syllabus

Meeting Times

- Section 1: Monday and Wednesday, 1:15pm-2:30pm, TG 103.
- Section 2: Monday and Wednesday, 2:45pm-4:00pm, TG 103.

People

- Professor Chris Stone: `stone@cs.hmc.edu`
- Grutors: Jonathan Beall, David Buchfuhrer, Max Gibiansky, Ben Tribelhorn, and Matt Walsh

Communication

To us: Homework-related questions should be sent to `cs70help@cs.hmc.edu`, rather than directly to the professor or to specific grutors. Mail sent to `cs70help@cs` goes to the professor and all the grutors, and hence is more likely to be answered in a timely fashion.

To you: The course mailing list `cs-70-1@hmc.edu` will be used for announcements relevant to the entire class; it sends mail to your campus account (`@hmc.edu` or `@pomona.edu` or ...). Mail specifically to you (e.g., homework results) is normally sent to your CS account (`@cs.hmc.edu`). You are responsible for regularly reading mail sent to both addresses.

The web page <http://www.cs.hmc.edu/cs70> also contains material relevant to the course.

Books

There are two recommended texts for the course. Lectures and homeworks will assume you have done any assigned readings.

- *Data Structures and Problem Solving Using C++ (second edition)*, Mark Allen Weiss, Addison-Wesley, 1999. ISBN 0-201-61250-X
- *The C++ Programming Language, third edition*, Bjarne Stroustrup, Addison-Wesley, 1997. ISBN 0-201-88954-4. (The "Special Edition" of this book is also fine.)

Course Topics

This course builds on the foundations in computer science you received in CS 60, and helps you develop your programming and problem-analysis skills. It also provides a grounding in fundamental data structures. Specifically, it covers:

- The underlying programming model of the C++ language, and the way it affects how you must program;
- Basic principles for writing correct and readable code (in any language);
- Abstract data types including arrays, lists, stacks, queues, dequeues, and dynamic dictionaries;
- Efficient implementations for these data types, including arrays, lists, trees, self-balancing trees, hash tables, and B-trees;
- Analysis of data structure algorithms, including worst-case, average-case, and amortized analysis;
- Basic Unix software development tools (make, Subversion, ...);

You will also get practice writing software, including some fairly large programs, which should allow you to improve your coding/debugging skills and speed.

Grading

55%	Homework Assignments
20%	Mid-term Exam
20%	Final Exam
5%	Participation/In-Class Work

Homeworks

Assignments will go out frequently; typically longer (usually programming) assignments will be distributed on Wednesdays, and shorter (usually written) assignments will be distributed on Mondays. Assignments will be roughly weighted by the amount of work required.

Late assignments *will not* be accepted except in rare special cases where arrangements were made with the instructor well before the deadline, or in case of unforeseeable crises such illness. If you know that some other commitment will cause you to be away from campus, talk to the instructor.

All code will be tested on Wilkes or Knuth, the department's Linux servers. You may use one of these machines to develop your code (Knuth is faster). You can also develop your code elsewhere (e.g., a terminal-room Mac, off-campus systems, or your own computer), but if so, you should be sure to test your final code on a Linux server.

Assignments are graded both on how well they satisfy the exact specification of the problem and on how well they demonstrate good programming practices. (Outrageously ugly or inefficient code that produces only “something similar to” the desired output fails on both counts and will receive a very low score.) Unless otherwise specified, you must use the exact filenames specified in the assignment; if an assignment tells you to use the names `README` and `Answers.txt`, the names `ReadMe.txt` and `answers.txt` are not acceptable.

You must submit a reasonable attempt to receive a grade for a homework assignment; submitting blank files or not submitting anything will result in a grade of NC (no credit) that will count as a 0 when calculating your final grade. To protect you against the effects of a bad week, the lowest non-NC grade from your submitted homework assignments will be dropped during the calculation of your final grade for the course.

Pair Programming

CS 70 uses the pair-programming methodology for some programming assignments. Two people work together as a team, with one person typing and the other at their side observing, critiquing, and planning what to do next. The two roles should occasionally switch, so that time at the keyboard is equally divided; ideally, the screen is visible to both participants, and only the keyboard moves back and forth.

The resulting work should be a genuine joint effort, though commenting and very minor code changes can be done alone. You will be violating the honor code if you divide the work such that you work separately with one person doing one half of the work and the other person doing the other half.

Computing at Pomona

All students enrolled in CS 70 may use the machines in Beckman B102 and B105. However, Pomona College students may wish to work closer to home.

The CS program at Pomona has workstations where students can work either locally or with sessions on the HMC computers. For accounts and card key access to the Pomona lab, see the web page <http://www.dci.pomona.edu/> or contact the systems manager Mike Dickerson in Andrew 252a,

Pomona also has a site license for the SSH Secure Shell, for computers running Windows, Linux, or Solaris. Pomona students can use `ssh` on their own computers for doing coursework. Contact Mike Dickerson for information.