Quick Facts

1 Class Overview

Course Code:	CS 70
Course Title:	Data Structures and Program Development
Website:	http://www.cs.hmc.edu/courses/2003/spring/cs70/
Help Email:	cs70help@cs.hmc.edu
Professor:	Melissa O'Neill ‹oneill@cs.hmc.edu›, Olin 1243, x79661
Prerequisite:	CS 60 (or equivalent)
Credit Hours:	3

Class Meetings:

Section	Days	Time	Place
1	Mo/We	11:00–12:15 AM	BKB 134
2	Mo/We	2:45–4:00 PM	TG 101

1.1 Scope

This class covers the following topics:

- Major features of the C++ language, including pointers, classes, templates, and operator overloading
- Programming style
- Abstract data types and data structures, including arrays, lists, stacks, queues, dequeues, and dynamic dictionaries
- Efficient data structures for these data types, including arrays, lists, trees, self-balancing trees, hash tables, and B-trees
- Memory management
- Practical aspects of using data structures in programs
- Analysis of data structures, including worst-case, average-case, and amortized analysis

1.2 Goals

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 providings a grounding in fundamental data structures.

In addition, you should learn

- How to follow good C++ style conventions (including how to properly comment and document your code so that other people can understand it easily)

- How, and when, to use a range of common data structures (lists, arrays, stacks, queues, trees, hash tables, balanced trees, etc.)
- How to manage memory explicitly, including how and when to use pointers
- How to use basic Unix software development tools (makefiles, header files, etc.)

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

2 Electronic Access

Much of the communication for this course will be handled electronically through the class website and mailing list(s). You are responsible for being familiar with announcements posted to the class mailing list(s) and with the contents of the class website.

2.1 Website

The class website is available at

http://www.cs.hmc.edu/courses/2003/spring/cs70/

This website is also reachable from the CS department's home page (via the Course Schedule link). (http://www.cs.hmc.edu/~oneill/). As well as providing useful general information (such as how to find me when you have questions), homework assignments will be posted on the class website.

2.2 Mailing Lists

The class mailing lists are

- cs-70-1-l@hmc.edu for section 1
- cs-70-2-l@hmc.edu for section 2

If you are a preregistered Harvey Mudd student, you should already be on the correct list. If you are an off-campus student or you registered late, you may need to join the appropriate list by sending mail from your preferred account to listkeeper@hmc.edu, with a message body containing subscribe cs-70-*2*-l *Jane Doe* (using your own name and section, of course).

Most class-related questions should be sent to the help alias, cs70help@cs.hmc.edu. Mail sent to that address will be forwarded to me and all the graders, so that whoever sees it first can provide a prompt answer.

2.3 Computer Accounts

All homework assignments will be submitted on *turing*, the department's Sun Enterprise 3000. You must have an account on *turing* to complete the work for this class. If you do not already have an account, you should fill out an account-request form (available from the CS system administrator in Beckman B101).

If you have an account that is no longer active, see the system administrator in B101 to reactivate your account.

You can only reach *turing* from machines in the Beckman terminal room or by using `ssh`—you will need an `ssh` client on your personal computer. See `http://www.cs.hmc.edu/tech_docs/qref/ssh.html` for more information on obtaining, installing, configuring, and using `ssh` clients.

2.4 Email Accounts

While you can receive mailing-list mail at any address you choose, homework grades and other material meant specifically for you will be sent to your *turing* email address. It is your responsibility to check your email on *turing* regularly or to have a `~/.forward` file that forwards your *turing* email to an account that you do check regularly.

3 Grading and Homework Assignments

Most of your grade will be based on your completion of homework assignments. Furthermore, much of the material on the tests will be based on things you learn while doing your homework. Thus it is critical for you to do your homework assignments. You will be better off dropping the class than allowing your homework to slide.

There will be about 11 homework assignments during the term. You will have one week to complete most of them, but you will be given two weeks for some, due to their difficulty or other factors such as school breaks. Assignments will be posted on the class website. Assignments will normally be due in stages, with deadlines on Tuesday, Wednesday and Thursday evening at 10:00 PM. The *Homework Policies* handout covers general information on completing and submitting your assignments, as well as information about how your work will be graded.

In addition to the assignments, there will be two exams: a midterm and a final. End-of-term grades will be based on the following formula:

55%	Homework assignments
15%	Mid-term exam
20%	Final exam
5%	Class participation
5%	Class quizzes & exercises

Remember that you must submit *some* material 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 day, the lowest non-NC grade from your submitted homework assignments will be dropped during the calculation of your final grade for the course.

3.1 Collaboration and the Honor Code

All students—even those from other colleges—are expected to understand and comply with Harvey Mudd College's Honor Code. If you haven't already done so, you must read, sign, and abide by the computer science department's interpretation of the Honor Code to participate in this course.

You are encouraged to discuss general features of assignments and the ideas involved with other students, including general approaches to the problems, bugs in the specification, how long you've spent working on a problem, and so forth. You may also help each other with issues related to completing the assignments—how to use Unix, C++ syntax, and the like.

You may also use example code—such as code from your textbooks or other reference materials—as a starting point for designing your own code.

However, except for explicitly assigned group projects, you should not discuss the details of how you designed and implemented your algorithms or solved mathematical problems. You must not exchange literal copies of material, whether that material consists of code, program output, or English-language text (e.g., documentation). You also may not copy material from published or online sources, with or without cosmetic changes (such as altering variable names) without explicit permission. If you do have permission to use externally written material, you must attribute it properly and clearly indicate which material is yours and which material is not yours.

If you aren't sure whether something you've done or plan to do is allowed, you should explicitly document what you did and—if at all possible—consult with the course staff, ideally *before* you take the questionable action. Similarly, document any extensive or particularly important help you obtain, even if that help seems legitimate. If the questionable material or extensive help is explicitly marked as such, you might lose points but avoid violating the Honor Code.

These principles apply to all methods and media of discussion or exchange (voice, writing, email, etc.).

3.2 Attendance

You are expected to attend every class. I will not be taking attendance, but many classes will have group exercises that will affect your final grade. If you miss a class, you must find out what happened and get notes from a fellow student.

3.3 Due Dates

Unless announced, there are no automatic extensions to homework (or other) deadlines. Late submissions will not be accepted. If you are unable to turn in an assignment by its due date, you may be able to get an extension if

1. There are extenuating circumstances
2. You tell me that you're having problems (either directly or via someone else, such as the Dean of Students) as soon as you know—*before* the assignment's due date
3. You negotiate appropriate arrangements with me to make up the work

I won't allow you to get very far behind in your work unless you are in a situation that could entitle you to an incomplete at the end of the term (e.g., a major medical problem).

Also, please note that even if you have met the requirements listed here, I may still deduct points.

3.4 Illness

If you get sick during the term, notify me immediately, even if you think that being sick will affect your ability to complete your assignments. You should also notify me any time that you're sick enough to miss *any* classes (not just CS 70) or find that your performance is below par for any reason.

3.5 Getting Help

If you need help with a problem, send email to the `cs70help@cs.hmc.edu` mailing list, which is read by me and the graders, maximizing your chances of getting a quick answer to a question.

If you have sensitive issues—such as personal issues or Honor Code violations—that you need help with, contact me directly.

4 Textbooks

Textbooks cover much of the material discussed in lectures. By reading your textbooks prior to class and being aware of what they cover, you can significantly reduce the amount of note taking you need to do in class.

4.1 Required Texts

There are two required textbooks for this course:

1. Mark Allen Weiss, *Data Structures and Problem Solving Using C++*, second edition. Addison-Wesley, 1999. ISBN 0-201-61250-X.

   Be sure to get the second edition of Weiss, not the first!

2. Bjarne Stroustrup, *The C++ Programming Language*, third edition. Addison-Wesley, 1997. ISBN 0-201-88954-4.

There will also be readings from Brian Kernighan and Rob Pike's *The Practice of Programming* (Addison-Wesley, 1999; ISBN 0-201-61586-X), which is on reserve in Sprague Library.

4.2 Additional Books

The following books contain supplementary material that may be of interest. You do not have to buy these books—copies will be available on reserve in the library.

- Timothy Budd, *C++ for Java Programmers*. Addison-Wesley, 1999. ISBN 0-201-61246-1. Useful if you know Java but not C++. Note that the class website includes links to the author's errata and numerous notes and corrections to this book (accessible only from within the Claremont Colleges).
- *ISO C++ Standard*. ISO/IEC 14882-1998. The official specification of the C++ programming language. Available from the ISO and ANSI websites.
- Steve Oualline, *Practical C++ Programming*. O'Reilly, 1995. ISBN 1-56592-139-9. A fairly readable C++ reference.
- Bjarne Stroustrup's *The Design and Evolution of C++*. Addison-Wesley, 1994. ISBN 0-201-54330-3. An interesting and sometimes enlightening history of how C++ has evolved.

4.3 Terminal Room Library

The Terminal Room, Beckman B102, has a small library of useful reference books. That library includes both of the required textbooks and most of the other books mentioned here.