Quick Facts

Essentials

Course Code:	CS 70
Course Title:	Data Structures & Program Development
Website:	http://www.cs.hmc.edu/cs70/
or	http://www.cs.hmc.edu/courses/2006/fall/cs70/
Wiki:	http://www.cs.hmc.edu/cs70/wiki
or	http://www.cs.hmc.edu/twiki/bin/view/CS70Fall2006/
Help Email:	cs70help@cs.hmc.edu
Professor:	Melissa O'Neill ‹oneill@cs.hmc.edu›, Olin 1243, x79661
Prerequisites:	CS 60
Credit Hours:	3
Class Times:	Tues/Thu 2:45–4:00 pm TG 103

Overview

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 and a solid working understanding of C++. You will learn

- How to write readable programs in C++ (including approaches to keeping code elegant and simple and 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.

Detailed learning objectives for the class are given in the *What You Lean in CS 70* handout.

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 an-

nouncements posted to the class mailing list(s) and with the contents of the class website.

Website

The class website is available at

http://www.cs.hmc.edu/cs70/

This website is also reachable from the CS department's home page (via the Course Schedule link). 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.

Wiki

CS 70 uses a Wiki site that allows *you* to post material about CS 70, available at

http://www.cs.hmc.edu/cs70/wiki

If you would like to create a discussion area for any topic in CS 70, you can with the Wiki site. The most important area of the Wiki is the CourseNotes area, where we post the content from our lecture slides and you add all the material that was not on those slides (including oral material and material written on the board). Included on the Wiki is a roster for listing the a "designated note taker" for each class. These note takers have primary responsibility for the notes for a given class, but you should check their notes regularly and make sure that they are complete and free of errors.

You should register on the CS departmental Wiki site as soon as you can, and add your name to the lecture-notes roster as soon as possible.

To reduce the risk of this site being archived by web spiders, there are *no links* to this website from the main course site. You can, of course, bookmark the site yourself (but please don't put a link on a publicly accessible page).

Mailing Lists

The class mailing list is cs-70-l@hmc.edu. If you are registered in the course, you should be on this mailing list. If you find that you are not receiving CS 70 email messages, contact CIS, or alternatively take matters into your own hands and manually join the appropriate list by sending mail from your preferred account to listkeeper@hmc.edu, with a message body containing subscribe cs-70-l.

Most class-related questions should be sent to the help alias, cs70help@cs.hmc.edu.

Computing Resources

You undertake your homework assignments using the computer resources provided by the CS department, or you may use your own machine, if you can set it up ap-

propriately. All the necessary software for CS 70 is set up on knuth, which is a fast Linux-based CPU server.

To use knuth, you need a CS account. 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 to reactivate your account.

You can only reach knuth 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.

Using Your Own Computer

For many students, especially students running Microsoft Windows, it is easiest *not* to use your own machine, but to log into knuth. If you nevertheless do wish to develop your assignment code your own computer, you will need a standard Unix-like development environment with gcc, make, perl and so forth. MacOS X and Linux are suitable environments, provided that you have installed the necessary developer tools. You may also be able to work under Cygwin environment for Microsoft Windows.

CS 70 uses Subversion (a.k.a. svn) for collaboration and assignment submission, so you will also need to find and install at least version 1.3.2.

All homework assignments will be tested on knuth. If you use your own machine, it is your responsibility to check that your code also works on knuth.

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 CS 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.

Coursework & Grading

Your final grade in CS 70 will be calculated by combining your grades each assessed course component in the proportions shown in Table 1. (Sometimes it can be necessary to make small adjustments to these proportions during the semester. If any adjustments are necessary, the changes will be announced in class and posted on the course website.)

55%	Assignments
15%	Midterm
20%	Final
5%	Class Participation
5%	Wiki Participation

Table 1: Course Components

Assignments

There will be approximately ten 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 are posted on the class website. Assignments will normally be due late Monday evening.

Readings & Quizzes

I may occasionally set reading assignments. I may also occasionally instigate quick quizzes on material you are supposed to have read.

Exams

There are two exams, a midterm and a final. These exams are three-hour open-book take-home exams. In general, exams test do not test recall of facts, but your ability to *apply* what you know.

Your Responsibilities

You have a number of responsibilities in this class. It is important that you understand them so that you can avoid costly mistakes such as loss of credit on an assignment or accidental violation of the honor code.

Pair Programming

CS 70 uses the increasingly prevalent *pair-programming* methodology for all homework assignments. At the start of the semester the class divides into pairs who work together throughout the semester. All homework will be done as a pair, with a single, joint, assignment turned in by one member of the pair.

In the pair-programming model, you *must* spend the bulk of your time working together as a team, with one person at the keyboard and the other at their side watching and making suggestions. All work should be a joint effort. 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.

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, basic C++ programming issues, and the like.

If your work is influenced by materials you have read, or discussions with other students, you should document that influence. When coding, you should describe those contributions in your comments.

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 still avoid violating the Honor Code.

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

Attendance & Participation

You are expected to attend *every* class. I will not be formally taking attendance, but many classes will have group exercises that will affect your final grade (both directly, because I grade for class participation, and indirectly, because questions on the exams are often similar to the group-exercise questions). If you wish to miss a class for any reason, you should ask beforehand about the make-up work you will need to do. If you are sick, you can send word to me through another student.

You are expected to participate actively in each class. The only way to receive a high grade for class participation is to be an active participant in the class (duh!).

Due Dates & Late Policy

Late Work Is Strongly Discouraged

The penalties for submitting late almost always outweigh the benefits. With proper planning, there is almost always a way to avoid suffering the penalties that arise from turning work in late. For example, if you know of an upcoming commitment that (such as an exam in another course, or a family event you must attend) that could affect your ability to get an assignment done by its deadline, you may ask for an assignment *early*. Similarly, if, after starting an assignment, you think it is impossible to

accomplish the necessary work in the available time, consult me before the due date. It may be that there is some help I can offer you that would allow you to proceed more quickly, or it may be that everyone in the class is having the same problem, in which case we might adjust the assignment.

Unforeseeable Extenuating Circumstances Only

Extenuating circumstances (such as illness) are dealt with on a case-by-case basis. In general, you are only excused for situations you could not have forseen, and only if you explain the situation at your soonest opportunity (either directly or via someone else, such as the Dean of Students), *before* the due date.

Late Penalty Formula

If, against my advice above, you do submit work late, your score will be scaled using the multiplier returned by the C function shown in Figure 1. This function (graphed in Figure 2) is swift to penalize lateness—even one minute late results in a lateness multiplier of 0.964 (i.e., a 3.6% penalty). At nine minutes late, the multiplier is 0.9, a 10% penalty. But with time the rate slows down, the 0.8 point is reached at a little before the first hour, and the 0.6 multiplier is not reached until the work is about 6.5 hours late. At, and beyond, the 12-hour point, the multiplier is zero. Times are based on the time that the submission process finished, not when you began submitting. For assignments due at 11:59 PM, we count late minutes from midnight and round down to whole numbers of minutes. Thus, you have about two minutes of grace after the clock ticks over to reading 11:59 PM. (It is, of course, risky to cut things close, especially since the submit system takes time to perform its work.)

Illness

If you get sick during the term, notify me immediately, even if you think that being sick will not 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.

Getting Help

If you need help with a course-related problem, come and see me. Talking in person is often the fastest way to resolve a problem, especially if it's conceptual. If you speak to a grader and are not happy with (or convinced by) their answer, you should see it as their failure rather than yours, and seek me out for a better answer.

If you have a simple question, a clarification, or a simple request, you may prefer to use email. You should 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 your question.

```
double lateMult (double mins)
{
    const double MAX_LATE_PERIOD = 12 * 60;
    /* i.e. twelve hours (in minutes) */

    /* This formula is strange, but gives a good penalty curve. */
    double  lateness = mins / MAX_LATE_PERIOD;
    double  root     = sqrt(lateness);
    double  squared  = lateness * lateness;
    double  penalty  = 1.0 - (root * (1.0 - root) + squared * root);
    return (penalty < 0.0) ? 0.0 : penalty;
}
```

Figure 1: The Late Penalty Function

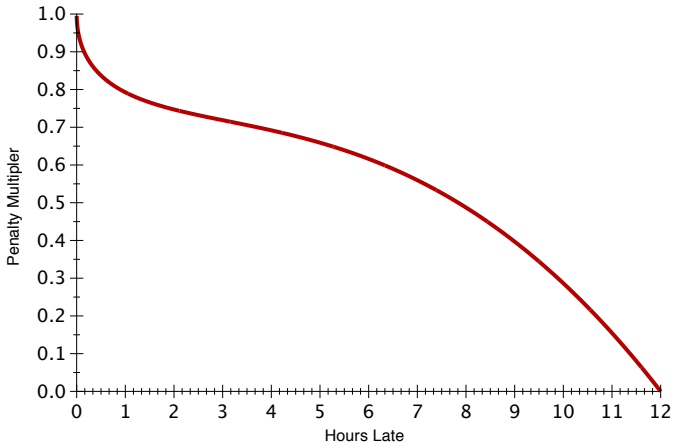

Figure 2: Graph of the Late Penalty Function

I believe that you learn best when you discover answers for yourself, so I will often respond to conceptual questions with questions of my own, designed to help you find the answer for yourself. If you wait to ask your question until you are feeling very frustrated (or until the last minute), you may find my answer maddening. For that reason, ask your questions *early*.

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

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.

Required Texts

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

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

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 Sprague.

- Brian Kernighan and Rob Pike, *The Practice of Programming*. Addison-Wesley, 1999. ISBN 0-201-61586-X.

- 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.

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.