Quick Facts

Essentials

Course Code: CS 70
Course Title: Data Structures & Program Development
Website: http://www.cs.hmc.edu/cs70/
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: Mon/Wed 11:00 AM–12:15 PM TG201 (Section 2)
Mon/Wed 1:15 PM–2:30 PM TG201 (Section 1)

Overview

This course builds on the foundations in computer science you received in CS 60, and helps develop your programming and problem-analysis skills. It also provides a grounding in fundamental data structures and a solid working understanding of C++. You will learn

- How to write readable programs in C++ (approaches to keeping code elegant and simple; 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 objectives for the class are given in the What You Learn 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 announcements posted to the class mailing list(s) and with the contents of the class website.
WEBSITE & WIKI

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 us when you have questions), homework assignments will be posted on the class website.

The CS 70 website is a “Wiki”. The pages are editable, allowing you to post material about CS 70. If you would like to create a discussion area for any topic in CS 70, you can do so. The most important area of the Wiki is the CourseNotes area, where I post the content from my 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 that associates every class with two people for who will act as our “designated note takers.” These note takers have primary responsibility for the notes for a given class, but you should check their work regularly. Since you can refer to these notes during exams, it is to your advantage to make them complete and free of errors.

As soon as possible,

1. Register for the Wiki via http://www.cs.hmc.edu/wiki/TWiki/TWikiRegistration
2. Add your name to the lecture-notes roster.

EMAIL

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

The class mailing list is cs-70-l@hmc.edu. If you are registered in the course, you should be on this mailing list. Mail to this list normally goes to your college @hmc.edu account.

Homework grades and other material meant specifically for you is often sent to your CS email (@cs.hmc.edu) address. It is your responsibility to check your email on a CS machine such as knuth regularly or to forward your CS email to an account that you do check regularly. (See http://www.cs.hmc.edu/wiki/QREF/Mail#How_do_I_forward_my_mail.)
Coursework & Grading

Assignments

All assignments come with a ”grading rubric” explaining exactly what we will be looking for, and a table summarizing how your grade will be computed. I highly recommend you read the rubric before starting an assignment, as it explains what we are looking for and may hint at possible pitfalls.

There will be approximately ten homework assignments during the term. Assignments will usually be distributed in the Wednesday lecture and be due on the following Wednesday night. A followup self-assessment is due Friday afternoon.

The self-assessment gives you a chance to objectively critique your own work; the instructions are on the Wiki. Particularly good or bad (or missing) self-assessments can affect your homework score, most often by a third of a letter grade (e.g., B ↔ B+ ↔ A−).

Exams

There are two exams, a midterm and a final. These exams are three-hour open-book take-home exams. The exams test more than recall of raw facts; they test your ability to apply what you know.

Course Grade

Your final grade in CS 70 will be calculated from the component grades as shown in Table 1.¹

<table>
<thead>
<tr>
<th>Weight</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>55%</td>
<td>Assignments</td>
</tr>
<tr>
<td>15%</td>
<td>Midterm</td>
</tr>
<tr>
<td>20%</td>
<td>Final</td>
</tr>
<tr>
<td>5%</td>
<td>Class Participation</td>
</tr>
<tr>
<td>5%</td>
<td>Wiki Participation</td>
</tr>
</tbody>
</table>

Table 1: Grading Formula

Your Responsibilities

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.

¹These weights apply in the normal case, but different rules may apply in extreme situations. You cannot, for example, pass the class, if you score 0% on the final, and 100% everywhere else.
**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 Outside Your Pair**

We would love for you to discuss the lecture and reading topics with any or all of your classmates; this can range anywhere from informal chats in the hallway to formal study groups organized on the course wiki.

You can even discuss high-level features of assignments and the ideas involved, 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 basic issues related to completing the assignments—how to use UNIX, basic C++ issues, and the like.

Two good rules of thumb for dealing with homework assignments are the “in your head rule” and “understanding, not rote learning” rules. When students help each other, they should leave with understanding in their heads, not physical or electronic artifacts. Thus you are not allowed to meet as a group and leave with notes on paper, nor can you help someone fix a bug and then leave without first reverting the bug to its unfixed state. Similarly, it is not okay to leave with an answer you don’t understand (in the worst case, where such a situation seems unavoidable because you have found out the answer but still don’t understand it and have to leave, it is vital for you to say so, since not doing so is dishonest). Also, if you are helping someone, remember that giving away answers can be detrimental to someone’s understanding—often it is better to ask leading questions instead. Working together as peers and coming to an understanding together is, of course, even better.

**Reminder: 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. Specifically:

- 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.

- You should not do anything that a reasonable student peer would describe as “subverting the clear intent of the assignment,” unless you have asked for and received permission to do so. Finding open-sourced code that you can use to solve an assigned problem, for example, would typically be subverting the intent of the assignment because your shortcut means that you do not learn what the assignment aims to teach.
• If any assigned material is substantially similar to material you have done before please contact me. In particular, in an situation where I am expecting you to do new work, you may not reuse or refer back to substantially similar work you have previously done. It is just as bad to copy from your past self as from someone else.

• 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 us through another student or by email.

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 us before the due date. It may be that there is some help we 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.

*Extenuating Circumstances — Unforeseeable Situations 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 foreseen, 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.
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

Late Penalty Formula

If, against the preceding advice, you do submit work late, your score will be scaled using
the multiplier returned by the function shown in Figure 1. This function (graphed in Fig-
ure 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 us immediately, even if you think that being sick will not affect your ability to complete your assignments. You should also notify us 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 someone. 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 us 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 us and the graders, maximizing your chances of getting a quick answer to your question.

I believe that you learn best when you discover answers for yourself, so I will often respond to questions with other questions, 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 the 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 us directly, rather than via the cs70help alias.

TEXTBOOKS

REQUIRED TEXTS

There are two required textbooks for this course:


The following book is recommended, but not required: