Quick Facts

Essentials

Course Code:	CS 131	
Course Title:	Programming Languages	
Website:	http://www.cs.hmc.edu/cs131/	
or	http://www.cs.hmc.edu/courses/2005/fall/cs131/	
Wiki:	http://www.cs.hmc.edu/twiki/bin/view/CS131Fall2005/	
Help Email:	cs131help@cs.hmc.edu	
Professor:	Melissa O'Neill ‹oneill@cs.hmc.edu›, Olin 1243, x79661	
Prerequisites:	CS 81, CS 60, CS 70	
Credit Hours:	3	
Class Times:	Mon/Wed 11:00–12:15 AM PA 1285	(Section 1)
	Mon/Wed 1:15–2:30 PM PA 1285	(Section 2)
Lab Times:	Thu 7:00–7:50 PM BK B105	(Lab A)
	Thu 8:00–8:50 PM BK B105	(Lab B)

Overview

This course provides you with a systematic framework for thinking about programming languages. You already had an introduction to the programming languages area in CS 60, where you used four languages representing at least four different approaches: Java for object-oriented programming, Rex for functional programming, Prolog for logic programming, and ISCAL for low-level (assembly) programming. But our emphasis is different in this course.

CS 131 is organized around ideas that recur in many different languages (with less emphasis on particular choices of syntax). Understanding these ideas will help you better understand the languages you know, and help you to quickly master new languages—even languages that haven't been invented yet. The concepts you learn will also help you to devise new languages of your own (which happens more often than you may think). We will study both the formal topics that arise when specifying the behavior of programming languages, and the practical issues that arise in implementing them.

The assignments in the class give you practical experience in implementing key programming-language concepts, as well as doing significant work in Standard ML, a language that has some valuable differences from the languages you may be used to.

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

The class website is available at

http://www.cs.hmc.edu/cs131/

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

The class wiki is available at

http://www.cs.hmc.edu/twiki/bin/view/CS131Fall2005/

This site allows *you* to post material about CS 131, including course notes.

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-131-l@hmc.edu. If you were registered in the course on the first day of classes, you should already be on the list. If 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-131-l.

Most class-related questions should be sent to the help alias, cs131help@cs.hmc.edu.

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 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.

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.

Group Programming in Labs

Lab assignments should be undertaken in groups of two. As much as possible, we would prefer to have the same two people work together throughout the semester.

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.

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, SML syntax, 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 following SML function

```
fun latemult (mins) =
    let val lateness = real(mins) / real(12 * 60)
        val root     = Real.Math.sqrt lateness
        val squared  = lateness * lateness
    in  Real.max(1.0(root * (1.0root) + squared * root), 0.0)
    end;
```

This function (which is shown graphically in Figure 1) 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.)

Remember, unless you've very much underestimated the assignment, it is almost never a good idea to submit late. The added points you gain from the extra work won't compensate for the effect of the lateness multiplier.

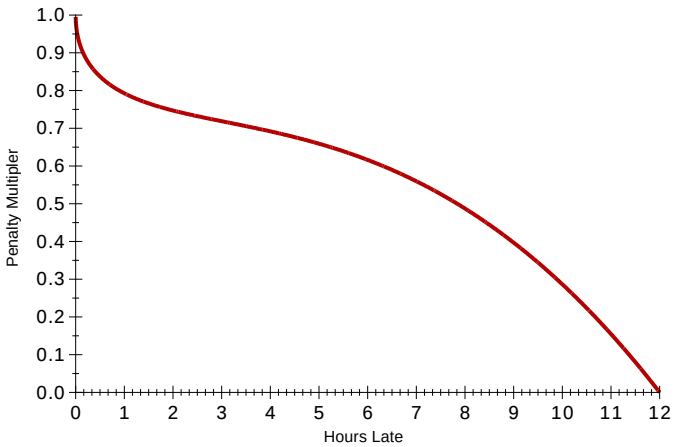

Figure 1: The Late Penalty Function

If you wish to finish an assignment but have your predeadline work graded, do not submit your later work using the regular submit system. Instead, send mail to `cs131help` and ask for special instructions for submitting your finished work for grader comments.

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 131) 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 or one of our graders. 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 cs131help@cs.hmc.edu mailing list, which is read by me and the graders, maximizing your chances of getting a quick answer to your question.

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

Textbooks

There is no *required* textbook for this class. All of the required material is presented in lectures, and given out in the form of assignments and handouts.

Optional Texts

There is one optional textbook for this course:

- Jeffrey D. Ullman, *Elements of ML Programming*, ML97 edition, Prentice Hall, 2001. ISBN 0-13-790387-1.

  On reserve in Sprague Library, QA76.73.M6 U45 1998.