

CS 2010 Syllabus
University of Utah, School of Computing
August 20, 2003

Course Staff and Schedule

Instructor Matthew Flatt
Office: 3224 MEB
Phone: 587-9091
Email: mflatt@cs.utah.edu

Lectures MWF 12:55–1:45, EMCB 101

Lab 2 Th 7:30–8:20, 3225 MEB
3 Th 8:35–9:25, 3225 MEB
4 Th 9:40–10:30 3225 MEB
5 Th 10:45–11:35, 3225 MEB
6 Th 11:50–12:40, 3225 MEB
7 Th 12:55–1:45, 3225 MEB
8 F 8:35–9:25, 3225 MEB
9 F 9:40–10:30 3225 MEB
10 F 10:45–11:35, 3225 MEB

Web Page <http://www.cs.utah.edu/classes/cs2010/schedule.html>

Teaching Assistants TBA (see web page)

Consulting Hours TBA (see web page)

Course Overview

The focus of the course will be on teaching you how to begin with a problem statement and then systematically design a computer program that solves the problem. The idea is to give you the proper knowledge, intuition, and habits on which you can base a lifetime of learning in computer science.

The required text is *How to Design Programs: An Introduction to Programming and Computing* by Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, and Shriram Krishnamurthi. From the preface:

[This book's] main focus is *the design process* that leads from problem statements to well-organized solutions; it deemphasizes the study of programming language details, algorithmic minutiae, and specific application domains.

For most of the course, you will be using a small programming language that resembles the programming language Scheme. This small language is far simpler than languages such as Java and C++, which are often used for introductory classes. As a result, we will be able to spend most of our time learning the program design process instead of learning arcane programming language details.

You will be using a programming environment, DrScheme, that was created to go along with the text. DrScheme is designed to make life easier for beginners. Other commonly used programming environments, such as Microsoft's Visual Studio or the Free Software Foundation's Emacs/g++, are designed to make life easier for experts.

In other words, you will be using a text, a programming language, and a programming environment that are designed with the needs of beginners in mind. You will probably not be able to find a summer job writing programs using DrScheme. You will, however, be extremely well-positioned to learn languages such as Java and C++ (or even Scheme) in subsequent courses.

Pragmatics

We will meet for lecture on Mondays, Wednesdays, and Fridays for 50 minutes in EMCB 101. In lecture, I will use a projected laptop computer, sometimes with slides and sometime using DrScheme interactively. Slides and DrScheme transcripts will become available on the course web page. For lectures with slides, I will make every effort to put the slides on the web page well before lecture.

You will also meet with a TA in small laboratory sections for one hour on Thursdays or Fridays in MEB 3225. This room is equipped with a computer for each student. In each section, your TA will guide you through a set of interactive programming exercises. The laboratory sections will be absolutely essential to your success in the course, and all students are required to attend each section.

The schedule web page is the central organizational tool for the course:

`http://www.cs.utah.edu/classes/cs2010/schedule.html`

The schedule page contains a tentative day-by-day schedule that will be revised as the semester progresses. Reading assignments are listed on the schedule page for each lecture day. Slides and DrScheme transcripts will be linked from the schedule page, as will programming assignments and solutions.

Most Wednesdays, I will post a programming assignment for the following week on the schedule page. It will be due on Wednesday (seven days later) at 11:59 pm by electronic handin. Homework submissions more than 8 hours late count as half credit when final grades are calculated, except for the first two such submissions that are within 48 hours late. No submissions more than 48 hours late will be graded (and the electronic handin program will reject such submissions). Homework is graded on the following scale: “check plus” for essentially perfect work worth 100%, “check” for flawed but acceptable work worth 80%, “check minus” for seriously flawed work worth 50%, and zero for unsubmitted or unacceptable work worth 0%.

The course staff (instructor and teaching assistants) will hold regular consulting hours each week, during which we will be available to help you with questions or problems. The consulting schedule has not yet been determined, and it will be posted on the web page.

There will be two midterms, held in place of lecture, and one two-hour final. The midterms will be on Monday, September 22 and Friday, November 7. The final will be on Monday, December 8, 1:00-3:00 PM.

Your final grade will be based on the two midterms (15% each), the final (22%), the weekly assignments (45%), and attendance at and participation in the lab sections (3%).

Programming Assignments

All of the programming that you do in this course will be supported by the DrScheme programming environment (even the Java part near the end of the semester). DrScheme is available for your use in the College of Engineering’s CADE Lab in EMCB 224 and 226. The computers in the CADE Lab are almost all Sun workstations running Solaris (a version of Unix). We will be distributing your CADE Lab login name and password in the first laboratory section. (If you have not already taken it, you can take CS 1010 this semester to learn how to use the CADE Lab workstations.)

You can obtain a free copy of DrScheme to install on your home computer. An installation link is available on the class web page. DrScheme runs under Windows, Mac OS X, and Unix/Linux. DrScheme also runs under Mac OS 9, but the electronic-handin part does not work on that platform.

Whether you choose to do your assignments at home or in the CADE Lab, you will need to submit your completed assignment through the Handin button in DrScheme. The Handin button will not be present the first time that you start DrScheme; you will need to download and install an extension. (The installed extension resides in your personal space on the filesystem.) Instructions for downloading and installing the Handin extension appear on the course web page. The first lab will also demonstrate the installation process.

When you click the Handin button, you must supply a username and password to the handin server. Your username is the same as for your CADE account. Your initial password will be mailed to your CADE account. Use the Manage CS2010 Handin... item in DrScheme's File menu to change your password at any time. When Handin communicates with the handin server, it uses SSL, so the connection is secure. Nevertheless, you should not use your CADE password for your handin account, because the handin server is maintained by the course staff, not CADE.

When you submit an assignment, the handin server inspects a few parts of your submission and rejects it if something is obviously wrong. These checks are intended to prevent basic misunderstandings and improve the feedback that you will receive from human graders. Human graders will inspect every submission and assign grades.

How To Approach this Class

For most of you, this will be your first class in computer science. For many of you, this will be one of your first classes at the University of Utah. Here is some advice on how to approach this class.

- Skim the relevant chapter in the textbook before you come to lecture, and read more carefully after the lecture. The lecture will not assume that you have read the current text chapter in advance, and the lecture should help you understand the context and key points of the chapter. Nevertheless, the lecture does not fully replace reading, because the lecture will omit fine points that are discussed in the text.
- Concentrate in lecture. The concepts that are presented in lecture are what's important; you will be able to find the details in the book. If you write down everything that is said and then try to figure it out later, you are wasting your time coming to lecture. Instead, think about what is being said. Try to answer all the questions that are asked, even if you only in your head. Raise your hand and ask a question when you don't understand something. Try to understand everything. Don't give up!
- Participate in the labs. You'll be sitting at a computer in a room with a teaching assistant and a few other students. Take advantage of the computer by trying things out. That way, you'll discover the things that you don't understand in a setting where there are plenty of other people to help you out.
- Respect the assignments. Some students expect that if they have done the reading, concentrated in lecture, and participated in the labs, then the assignments will be straightforward. What these students don't understand is that the assignments are designed to challenge you by requiring that you apply the concepts you have learned to new situations. The assignments will be your most important learning experience in the course; they will rarely be straightforward. You should start each problem set days before it is due. This way, you will have time to take a break when you get stuck.

Beginners are often surprised by the amount of human effort that has to go into designing, writing, and testing a program. Complaints from students about the amount of time required by introductory computer programming courses are universal. You should expect to spend three hours outside of class for every hour that you spend in class. In other words, you should expect to spend 12 hours per week reading, studying, and designing programs. Many of you will spend even more time. Please keep this in mind when setting up your schedule for the semester!

Getting Help and Information

The class web page, <http://www.cs.utah.edu/classes/cs2010/>, contains a variety of important resources, including the course schedule, course staff consulting hours and e-mail addresses, and links pertaining

to the textbook and DrScheme.

When we need to get in touch with you, we will send e-mail to you at your account in the CADE Lab. You must either read your e-mail in the CADE Lab regularly, or arrange to have it forwarded to a different address that you read regularly. A link on the class web page explains how to forward mail from your CADE Lab account.

There is a class mailing list (`cs2010@cs.utah.edu`), a staff mailing list (`teach-cs2010@cs.utah.edu`), and a student chat mailing list (`cs2010-chat@cs.utah.edu`).

We will use the class mailing list to send urgent messages, such as corrections to problem sets or changes in due dates, to everyone in the class. *Only the course staff will be allowed to send mail to this list.* You *must* subscribe to this list. The course web page contains a link for subscribing.

You can use the staff mailing list to send questions to me and to the TAs. Before you send a question, however, please check the FAQ on the schedule page. If we feel that the answer to your question would be of interest to the class at large, we will add your question and our reply to the FAQ.

You can use the chat mailing list to communicate with other students in the class. You can subscribe or not as you see fit; a link on the class web page explains how. The course staff will *not* be monitoring this list on a regular basis. The course web page contains a link for subscribing to this list.

We encourage you to seek us out whenever you need help, advice, or encouragement. We will always be available during our regular office hours, and you can make appointments for other times. Simple questions can often be answered by phone or electronic mail. Our consulting schedule will be posted on the class web page as soon as it is finalized.

Cooperation vs. Cheating

Working with others on assignments is a good way to learn the material and we encourage it. However, there are limits to the degree of cooperation that we will permit.

When working on programming assignments, you must work only with others whose understanding of the material is approximately equal to yours. In this situation, working together to find a good approach for solving a programming problem is cooperation; listening while someone dictates a solution is cheating. You must limit collaboration to a high-level discussion of solution strategies, and stop short of actually writing down a group answer. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. If you base your solution on any other written solution, you are cheating.

When taking an exam, you must work completely independently of everyone else. Any collaboration here, of course, is cheating.

We do not distinguish between cheaters who copy others' work and cheaters who allow their work to be copied.

If you cheat, you will be given an E in the course and referred to the University Student Behavior Committee. If you have any questions about what constitutes cheating, please ask.

Students With Disabilities

Reasonable accommodation will gladly be provided to the known disabilities of students in the class. Please let the instructor know of such situations as soon as possible. If you wish to qualify for exemptions under the Americans With Disabilities Act (ADA), you should also notify the Center for Disabled Students Services, 160 Union Bldg.