Carnegie Mellon University
Department of Computer Science
15–212: Fundamental Principles of Computer Science II:
Advanced Programming Techniques
Sections using Standard ML (15–212–X)

`http://foxnet.cs.cmu.edu/15-212-X/`

Frank Pfenning

Fall 1996

1 Organization

Instructor

	Frank Pfenning
Office:	Wean Hall 8127
Extension:	x8-6343
E-mail:	`fp@cs.cmu.edu`
URL:	`http://www.cs.cmu.edu/~fp`
Office Hour:	Th 1:30–2:30 and by appointment

Teaching Assistants

	Iliano Cervesato	Carsten Schürmann
Office:	WeH 8121	WeH 8121
Extension:	x8-3069	x8-3069
E-mail:	`iliano@cs.cmu.edu`	`carsten@cs.cmu.edu`
Office Hours:	Tu 11:00–12:00, We 2:30–3:30	

Secretary

	Rosemary Hornyak
Office:	Wean Hall 8106
Extension:	x8-3853
E-mail:	`rosemary@cs.cmu.edu`

Meeting Times

Lecture: Tuesday and Thursday, 12:00 – 1:20, WeH 5403.

Recitations: Wednesday of each week at the following times:

Section:	C	G
Time:	10:30	2:30
Place:	SC 200	SC 200
Instructors:	Cervesato & Schürmann	

2 Textbooks

The required textbook for this course is *ML for the Working Programmer*, **2nd edition**, by L.C. Paulson, Cambridge University Press, 1996. Note that most of the lecture material does not appear in the textbook. Occasionally, supplementary notes will be handed out and made available electronically.

The following supplementary texts have been placed on reserve in the E&S Library:

1. *Elements of ML Programming*, by Jeffrey D. Ullman, Prentice-Hall, 1994.
2. *Elements of Functional Programming* by Chris Reade, Addison-Wesley, 1989.
3. *Introduction to Standard ML* by Robert Harper.

These supplementary texts are slightly out of date with respect to the version of Standard ML used in this class. Some programs in these texts will not work as given—please refer to the local guide to using Standard ML at CMU for more information.

3 Objectives

The objective of the CS 211/212 course sequence is to provide the student with a firm foundation in the fundamental principles of computer science. CS 211 is concerned with the fundamentals of data structures and algorithms, including the analysis of algorithms and their implementation in the C programming language. CS 212 is concerned with high-level programming techniques. You will learn how to decompose problems, how to compose the solutions into complete programs, and how to reason about programs to ensure that they are correct.

The course will be taught using the Standard ML programming language, a relatively new language supporting higher-order functions, exceptions, polymorphism, data abstraction, and modularity. Using Standard ML as a vehicle we will cover the following topics:

- Induction and recursion, loop invariants.
- Symbolic computation.
- Data abstraction, representation invariants.
- Higher-order functions, continuations.

- Assignment, mutable data structures.
- Modularity and the structure of large programs.
- Streams and demand-driven programming.
- Exceptions, advanced control.
- Grammars and parsing.
- State encapsulation, objects.
- Computability.

4 Class Schedule

The schedule of lectures and associated readings for the class is given in Table 1. This schedule is subject to change during the semester.

5 Course Requirements

Participation in this course consists of the following activities:

- Attending and participating in lectures and recitations.
- Reading the textbook and handouts.
- Carrying out homework exercises.
- Taking examinations.

Each of these activities will have a bearing on your final grade in the course.

Attendance at lecture and recitation is *strongly* encouraged. You are responsible for all material presented in lecture and in recitation. The lectures are *not* based directly on the textbook. You should *not* expect that all lecture material will be given to you in written form, nor should you expect that the material presented in lecture will be drawn from the text. The recitations are a supplement to the lectures consisting of both new material not presented in lecture and review of both lectures and homework problems. Active participation in recitation is strongly encouraged, and will increase your chances for getting a good grade.

6 Homework

Homework assignments are a critical part of the course work. A full understanding of the material in the textbooks and lectures can only be gained by applying them to solve problems. There will be both written assignments and programming assignments. All assignments will be handed out, and will be

Date	*Topic*	*Supplementary Reading*
27 Aug	Evaluation & Typing	2.1–2.6, 2.11–2.12
29 Aug	Binding, Scope, Functions	2.14–2.16, 2.20–2.22
3 Sep	Recursion & Induction	6.1–6.3
5 Sep	Products, Datatypes, Patterns	2.7–2.8, 4.1–4.4
10 Sep	Tail Recursion, Structural Induction	6.4–6.5
11 Sep	*Assignment 1 due*	
12 Sep	Data Structures	3.1–3.6, 4.10–4.13
17 Sep	Representation Invariants	4.14–4.16
19 Sep	Regular Expressions	*none*
24 Sep	Higher-Order Functions, Continuations	5.1–5.11
25 Sep	*Assignment 2 due*	
26 Sep	Exceptions	4.5–4.9
1 Oct	*Review*	
3 Oct	*Midterm*	
8 Oct	Modularity: Signatures and Structures	2.20–2.22
9 Oct	*Assignment 3 due*	
10 Oct	Data Abstraction	7.1–7.5
15 Oct	Reasoning with Abstract Types	*none*
17 Oct	Modularity: Functors	7.8–7.10
22 Oct	Modularity: Hierarchies	7.11–7.15
23 Oct	*Assignment 4 due*	
24 Oct	Games	5.17–5.20
29 Oct	Streams, Demand-Driven Computation	5.12–5.16
31 Oct	Mutation and State	8.1–8.3
5 Nov	Encapsulating State, Objects	8.4–8.6
6 Nov	*Assignment 5 due*	
7 Nov	Simulation	*none*
12 Nov	Grammars, Abstract Syntax	*none*
14 Nov	Parsing	9.1–9.3
19 Nov	Interpreters	9.10–9.13
20 Nov	*Assignment 6 due*	
21 Nov	Computability	*none*
26 Nov	Computability	*none*
28 Nov	*No Class* (Thanksgiving)	
3 Dec	System Design, Other Languages	*none*
5 Dec	*Review*	

Table 1: Schedule of Lectures and Readings

made available on line, during the recitation period on Wednesday every other week.

Completed homework assignments are due at recitation. Programming assignments are to be handed in using the electronic submission procedure described below. Programming assignments are due at *10:00AM* on the due date; written assignments are to be handed in at the start of recitation.

Late homework will be accepted up to the *start* of lecture on Thursday, with a 25% penalty assessed for lateness. No homework assignments will be accepted after the start of the Thursday lecture. Requests for extensions will be considered only on personal application to the instructor; *absolutely no requests for extension will be accepted via telephone or electronic mail.*

Some homework assignments will include extra credit. Any extra credit earned will be recorded separately from your nominal grade, and will be used in determining your final letter grade for the course.

Graded assignments will be returned at recitation one week after they are submitted. Requests for re-grades are handled by the teaching assistant responsible for the assignment, and should be made directly to him or her.

Programming assignments will be graded based on style and correctness; a working program is *not* sufficient for full credit. Make sure your code is properly annotated with invariants and comments.

7 Examinations

There will be one mid-term and one final examination in this course. The midterm will be given during class on *Oct 3* and will be a closed-book examination. The final examination will be open book and last for three hours.

Missed examinations count as zero credit. Except in the case of dire medical or family emergencies, no make-up examinations will be administered.

8 Final Grade

The raw score for the course is determined by a weighted average of each student's scores on homework assignments and exminations according to the following formula:

Homeworks	50%
Midterm	20%
Final	30%

The final letter grade for the course will be determined by evaluating each student's performance relative to that of the other students in the class. The raw scores are plotted as a histogram which is then used to determine the letter grade. Individuals whose raw scores lie near the borderline between letter grades may have their grade adjusted upwards or downwards based on such factors as attendance at lecture and recitation, earned extra credit, participation in recitation, and any special circumstances.

A grade of "incomplete" will be assigned only in the most extraordinary circumstances.

9 Cheating and Collaboration

Any work submitted as a homework assignment or examination must be entirely your own and may not be derived from the work of others, whether a published source, another student, or any other person. It is cheating to copy, examine, or alter anyone else's homework assignment or computer program, or to use a computer program to transcribe or otherwise modify or copy anyone else's files. It is cheating to submit solutions derived from any source, including textbooks, journal articles, or course materials from any other course.

It is your responsibility to protect your programs, homework assignments, and examinations from illicit inspection or copying. You are expected to use the standard file system protection mechanisms to render your course materials unreadable to anyone other than yourself. Failure to do so may be regarded as evidence of improper collusion on homework assignments. Any attempt to circumvent computer security protections by browsing through another person's files or by taking advantage of flaws in the security system will be taken as *prima facie* evidence of cheating, and will be dealt with accordingly.

Violations will be handled in accordance with the University Policy on Cheating and Plagiarism.

10 Computing Facilities

Course Directory

The home directory for the course is `/afs/andrew/scs/cs/15-212-X`. It contains the following sub-directories:

`studentdir`	electronic submission directory
`handouts`	course handouts
`assignments`	homework assignments
`solutions`	sample solutions
`lib/sml`	library of SML support code

Programming Language

All programs in this course will be written in SML for execution in the Andrew environment. You should familiarize yourself as soon as possible with the use of MLWorks, the ML implementation used in the course. A handout describing the use of MLWorks under Andrew is available in `handouts/sml-andrew.ps` and will be distributed in class. For a quick start, run

`/afs/andrew/scs/cs/15-212-X/mlworks/bin/mlworks`.

Note that MLWorks currently runs only on SparcStations under Solaris and no other machine architecture.

Electronic Hand-in

All programs are to be handed in using the electronic hand-in procedure. To do this, copy the appropriate file(s) into the following submission directory:

`/afs/andrew/scs/cs/15-212-X/studentdir/`*userid*`/ass`*n*

where *userid* is your Andrew user identification code and n is the assignment number. You must be authenticated with your Andrew user identification to have access to your submission directory.

The submission directory will be automatically scanned at *10:00AM* on the due date to retrieve whatever files are present; further access to the directory will be precluded. You may freely copy files into the submission directory at any time up to the deadline for the assignment; the copy present at the submission deadline will be regarded as your submission for that exercise. A second late sweep occurs at *12:00* noon on the Thursday after the due date for late assignments.

Bulletin Boards

There are two bulletin boards devoted to this class, each with a title of the form `academic.cs.15-212-X.`*topic*, where *topic* is one of the following:

`announce` This bulletin board is used for official announcements by the teaching staff to publicize clarifications, corrections, or changes to the homework assignments, and to make announcements of changes to the class schedule. Anyone can read this bulletin board, but only the teaching staff may post to it. *Read this bulletin board regularly.*

`discuss` This bulletin board is to be used by the students to discuss homework assignments, and may be used to pose questions to the teaching staff regarding the homeworks. Anyone can read and post to this bulletin board.

The bulletin boards are your first line of communication with the teaching staff, and should be used in preference to electronic mail to make inquiries about the homework exercises. Electronic mail to the teaching assistants or professor should be limited to urgent communications requiring private handling or immediate response. Start early on your homework assignments. Do *not* expect the teaching assistants to be available for questions or read the bulletin boards late on Tuesday night before the deadline! *Electronic mail requests for extensions to complete a homework exercise will not be considered.*

Worldwide Web

The home page for the course is accessible by the URL

`http://foxnet.cs.cmu.edu/15-212-X/`.

The home page contains links to the course bulletin boards and to the course directories, as well as to various other sources of information about SML.

11 Advice

Learn by Doing

The homework exercises provide the most important part of your educational experience. Write out rough forms of your solutions, think them over, and write them out again in final form. Don't be afraid to try different approaches to the same problem.

Think Before You Hack

The homework exercises are designed to stress conceptual issues, and cannot ordinarily be solved sitting at a computer. The grading criteria stress readability of your code; it is *not* sufficient for a good grade that your programs work correctly. Omission of comments or invariants will also lead to deduction of points, so make sure your code is properly documented.

Start Assignments Early

Homework assignments are due every other week and require a substantial amount of work. Start your assignments early so you can get help in time if you need it. Since the MLWorks environment runs only on SparcStations, cluster space is limited and you may not be able to get on a machine close to the deadline.

Attend Lecture

The bulk of the course material is presented in lecture. *Most of the lecture material does not appear in the textbook*, so students are *strongly advised* to attend every lecture.

Attend Recitations

The recitations are the best time for you to review material, ask questions, and get personal help mastering the course material. The teaching assistants are friendly and enthusiastic. Supplementary material to the lectures will be presented in recitation; you are expected to attend and will be responsible for anything covered in the sections.

Keep Up

The course material is cumulative in nature and it is important to keep up. Experience has shown that once a student falls behind the pace, he or she tends to stay behind and suffers severely in the final grade. Keep up the pace, attend recitations and office hours for help, and see the instructor if you run into serious difficulties before they get out of hand.