

**CMSC 22610**  
**Winter 2009**

**Implementation of**  
**Computer Languages - I**

**Handout 1**  
**January 6, 2009**

## **Course Information**

### **General Information**

**Instructor:** Matthew Fluet  
**E-mail:** [fluet@cs.uchicago.edu](mailto:fluet@cs.uchicago.edu)  
**Office hours:** W 10:00am – noon (or by apt.); TTI-C #521 (6045 S. Kenwood)

**TA:** Adam Shaw  
**E-mail:** [ams@cs.uchicago.edu](mailto:ams@cs.uchicago.edu)  
**Office hours:** F 10:00am – noon; CS Linux Lab (Regenstein Library)

**TA:** Yingqi Xiao  
**E-mail:** [xiaoyq@cs.uchicago.edu](mailto:xiaoyq@cs.uchicago.edu)  
**Office hours:** W 3:00pm – 5:00pm; Ryerson 177-c2

**Lectures:** TuTh 1:30pm – 2:50pm; Ryerson 251

**Home page:** [www.classes.cs.uchicago.edu/archive/2009/winter/22610-1](http://www.classes.cs.uchicago.edu/archive/2009/winter/22610-1)

**Mailing list:** [cmssc22610@mailman.cs.uchicago.edu](mailto:cmssc22610@mailman.cs.uchicago.edu)  
[mailman.cs.uchicago.edu/mailman/listinfo/cmssc22610](http://mailman.cs.uchicago.edu/mailman/listinfo/cmssc22610)

**GForge server:** [cs22610.cs.uchicago.edu](http://cs22610.cs.uchicago.edu)

### **Overview**

The course covers principles and techniques for implementing computer languages, such as programming languages, query languages, specification languages and domain-specific languages. Topics include lexical analysis, parsing, tree representations of programs (both parse trees and abstract syntax trees), types and type checking, interpreters, abstract machines, and run-time systems. This is a project-based course involving the implementation of a small language using Standard ML.

This course is the first in a sequence of two courses that cover the implementation of computer languages. The second course (CMSC 22620; Spring 2009) covers more advanced issues related specifically to the translation of general purpose programming languages.

## Text Books

Required:

**Title:** Modern Compiler Implementation in ML  
**Author:** Andrew Appel  
**Publisher:** Cambridge University Press  
**ISBN:** 978-0521607643  
**Errata:** [www.cs.princeton.edu/~appel/modern/ml/errata.html](http://www.cs.princeton.edu/~appel/modern/ml/errata.html)

Strongly recommended:

**Title:** ML for the Working Programmer  
**Author:** Lawrence Paulson  
**Publisher:** Cambridge University Press  
**ISBN:** 978-0521565431

Optional:

**Title:** The Standard ML Basis Library  
**Editors:** Emden Gasner and John Reppy  
**Publisher:** Cambridge University Press  
**ISBN:** 978-0521794787

## Assignments, Exams, and Grades

There will be both homework assignments and programming projects. Homework assignments should be submitted at the beginning of class (1:30pm) on the due date. Programming projects will be automatically collected from the course GForge server at 10:00pm on the due date. Late homework assignments and programming projects will not be accepted, unless special circumstances have been discussed with the course staff *in advance* of the due date and/or other arrangements have been made.

There will be both a midterm exam (in class) and a final exam; see below for dates.

Grades will be assigned based on the following grading scheme:

<b>Homework:</b>	20%
<b>Midterm exam:</b>	20%
<b>Project:</b>	40%
<b>Final exam:</b>	20%

## Course Project

The project for the course is to implement a small functional programming language, called *LangF*. (Students who have taken CMSC 22100 should recognize the language as an enrichment of *System F*, the polymorphic  $\lambda$ -calculus.) The project will be divided into four parts, each requiring a significant programming effort. The implementation will be undertaken using the Standard ML programming language and submission of the project milestones will be managed using the course GForge server. Programming projects will be individual efforts (no group submissions).

## Important Dates

- January 23:** First project milestone (scanner)
- February 9:** Second project milestone (parser) [Moved from February 6.]
- February 12:** Midterm exam (in class: 1:30pm – 2:50pm; Ryerson 251)
- February 27:** Third project milestone (type-checker)
- March 15:** Fourth project milestone (code-generator)
- March 17:** Final exam (1:30pm – 3:30pm; Ryerson 251)

## Academic Honesty

As with all courses, the University of Chicago's policy on Academic Honesty and Plagiarism applies.

In this course, all submitted work must be your own work (i.e., written up or programmed by you alone) and must include acknowledgments of any collaborators or sources (other than course text books or handouts) used to produce your submission.

You are encouraged to discuss course material with other students. Discussion of homework assignments and programming projects is also allowed, but sharing solutions or code is not allowed.

## Disclaimer

I reserve the right to make any changes to the syllabus as I deem necessary throughout the course. Minor changes, such as programming project due dates, will be announced orally during class and posted on the course mailing list and home page. Major changes, such as grading percentages, will additionally be provided in writing.