CMSC 22610	**Implementation**	**Handout 1**
Winter 2004	**of**	**January 5**
	Computer Languages	

Course information

Instructor: John Reppy
Hinds 033 (By appointment only)
702-5534

Lectures: MWF 1:30-2:20
Ry. 251

Mailing list: `cmsc22610@cs.uchicago.edu`
`mailman.cs.uchicago.edu/mailman/listinfo/cmsc22610`

Home page: `www.cs.uchicago.edu/archive/2004/winter/22610-1`

Overview

A vast majority of computer programs must deal with textual input of some form or another. This input can range from simple configuration languages to data description languages (*e.g.*, XML) to scripting languages to full-blown programming languages. In this course, we will cover the tools and techniques used to process the full range of computer languages (*i.e.*, languages that specify programs and data on computers). Topics include scanning and parsing, tree representations of structured input, simple typechecking, translation between intermediate forms, interpretation, simple code generation, and some run-time system issues. There will be homework assignments and programming projects. We will use Standard ML (SML) for the majority of the programming projects, but some C programming will also be required. Students should have taken CMSC 15400 (Introduction to Computer Systems) and be familiar with C programming.

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

Texts

The main text for the course is

> *Modern Compiler Implementation in ML*
> by Andrew W. Appel

Cambridge University Press, 1998

We strongly recommend that you also obtain a copy of

ML for the Working Programmer (2nd Edition)
by L.C. Paulson
Cambridge University Press, 1996

Course project

The course project is to implement a small scripting language called *Mini-Lua* (a subset of the Lua language). The project will be divided into four milestones.

Important dates

January 23 First project milestone (lexer)

February 4 Second project milestone (parser)

February 11 Midterm exam (in class)

February 20 Third project milestone (analyser)

March 12 Final project due.

March 17 Final exam.

Assignments and Grading

There will be both written homework assignments and programming projects. In addition, there will be a midterm exam in class on Tuesday, November 18th. Grades will be assigned based on roughly the following weights:

Homework	20%
Midterm exam	20%
Project	40%
Final exam	20%

Paper copies of the assignments will be distributed in lecture and electronic copies will be made available for the course web page. Homework should be handed in at the beginning of class the day they are due. Programming projects will be automatically collected from your course CVS repository. In general, late homework and programming assignments will not be accepted, although valid excuses delivered before the assignment is due will be considered.

Academic Honesty[1]

The University of Chicago is a scholarly academic community. You need to both understand and internalize the ethics of our community. A good place to start is with the Cadet's Honor Code of the US Military Academy: "A Cadet will not lie, cheat, or steal, or tolerate those who do." It is important to understand that the notion of property that matters most to academics is ideas, and that to pass someone else's ideas off as your own is to lie, cheat, and steal.

The University has a formal policy on Academic Honesty, which is somewhat more verbose than West Point's. Even so, you should read and understand it.

We believe that student interactions are an important and useful means to mastery of the material. We recommend that you discuss the material in this class with other students, and that includes the homework assignments. So what is the boundary between acceptable collaboration and academic misconduct? First, while it is acceptable to *discuss* homework, it is not acceptable to turn in someone else's work as your own. When the time comes to write down your answer, you should write it down yourself from your own memory. Moreover, you should cite any material discussions, or written sources, for example,

```
Note:  I discussed this exercise with Jane Smith.
```

The University's policy, for its relative length, says less than it should regarding the culpability of those who know of misconduct by others, but do not report it. An all too common case has been where one student has decided to "help" another student by giving them a copy of their assignment, only to have that other student copy it and turn it in. In such cases, we view both students as culpable and pursue disciplinary sanctions against both.

For the student collaborations, it can be a slippery slope that leads from sanctioned collaboration to outright misconduct. But for all the slipperyness, there is a clear line: present only your ideas as yours and attribute all others.

If you have *any* questions about what is or is not proper academic conduct, please ask your instructors.

[1] In keeping with the spirit of this section, credit must be given to Stuart Kurtz for text.