

---

## CMSC 22100/32100: Programming Languages

### Course Information

M. Blume

September 30, 2008

---

## 1 Instructors

Matthias Blume (<http://www.tti-c.org/blume/>)

## 2 Where and When?

Classes will meet on Tuesdays and Thursdays from 1:30 to 2:50pm in Ryerson 276.

## 3 Course Website

Course website: <http://www.tti-c.org/blume/classes/aut2008/proglang/>.  
Check the website regularly for homework assignments and announcements.

## 4 Course Description

Programming languages are the most fundamental tools involved in the creation of software, and thus play a key role in computing. This is a foundational course exploring the principles and concepts underlying the design of programming languages. We use a formal approach based on operational semantics to give clear and precise descriptions of language concepts, such as flow of control, data structures and types, modularity and abstraction, and concurrency. The major paradigms of imperative, functional, and object-oriented paradigms will be covered, though the functional paradigm will be the main focus because of its simplicity. To support your theoretical understanding of the subject, we will use programming projects (implemented in Standard ML) to gain first-hand practical experience with language definitions and their realization on actual computer hardware.

## 5 Prerequisites

We will assume knowledge of basic mathematical concepts such as sets, and inductive proofs. If you are not readily familiar with these, then we recommend studying the first few chapters of Pierce's book. We will not assume any prior knowledge of the Standard ML language but you will be expected to learn the

language in a “demand-driven” fashion as the projects develop. While superficially this might seem like a distraction, we expect that studying Standard ML will directly support your appreciation of programming language fundamentals. (You may even find it independently rewarding.)

## 6 Teaching Assistant

Siwei Wang (<mailto:siweiw@cs.uchicago.edu>) is the teaching assistant for the course. He will hold office hours and grade homework assignments. In addition, Wonseok Chae (<mailto:wchae@tti-c.org>) has kindly volunteered to be an additional (unofficial) teaching assistant.

## 7 Text Book and Course Software

The text book is an earlier version of Robert Harper’s online text book:

Robert Harper. *Practical Foundations for Programming Languages*. (<http://www.tti-c.org/blume/classes/aut2008/proglang/text/offline.pdf>)

Graduate students will find supplementary reading material in Matthias Felleisen and Matthew Flatt’s text:

Matthias Felleisen and Matthew Flatt. *Programming Languages and Lambda Calculi*. (<http://www.cs.utah.edu/plt/publications/pllc.pdf>)

For additional reading, see:

Benjamin C. Pierce. *Types and Programming Languages*. (MIT Press, 2002, ISBN 0-262-16209-1)

For projects, we will use the Standard ML language and the SML/NJ implementation. For further information on SML, see Bob Harper’s Programming in Standard ML book, and the documentation for the SML’97 Basis Library (these are both available on the class web site).

You should install SML/NJ on your computer. You can either have the system administrator install it for you, or install it yourself. For installation instructions see the SML/NJ website. Make sure that you install a relatively recent version.

## 8 Homework, Exercises, Exams

There will be a number of homework assignments (about one a week) and a number of exercises. If you do your homework and exercises regularly, we expect that the exams will be reasonably stress free.

Homework assignments will be due at the specified day at 10pm. All written homework assignments need to typeset (using LaTeX) and submitted to me and the teaching assistants in a manner that will be specified later. For programming portions, you should submit a tarball of your sources.

At the end of some classes, we will assign an exercise. These need to be solved and handed in at the beginning of the next class.

Homework in which you do not achieve a minimum grade of B on the first try will automatically be re-assigned to you for a second time (in addition to whatever new homework is due that week).

## 9 Collaboration and Cheating

You are required to do your own homework. You can, however, discuss the problems and ideas for solutions with other students in the class. You must credit the people that you talk to by clearly including their names in the homework that you turn in.

You are not allowed to share code. In particular, you cannot do the programming assignments in groups. Similarly, you cannot share or copy final solutions to non-programming problems.

The rule of thumb is that you must be able to reproduce all the work that you turn in.

No collaboration is allowed in exams.

## 10 Grading

Homework and exercises will make up 55% of your final grade. The midterm amounts to 20%, and the final will make up 25% of the final grade.