

CS421: Programming Languages and Compilers

Topic 1: Course Introduction¹

Sameer Sundresh
sundresh@uiuc.edu

University of Illinois at Urbana-Champaign

May 29, 2007

¹Based on slides by Mattox Beckman, as updated by Vikram Adve, Gul Agha, Elsa Gunter, and Mark Hills.

Introduction

Welcome to CS 421!

- ▶ **Programming is the practice of automation.**
- ▶ The field of *programming languages* is the study of **how to automate programming tasks**.
 - ▶ *Translate a high-level design into an executable program.*
 - ▶ *Show that a program does or doesn't do what it's supposed to.*
- ▶ This class covers topics that are both
 - ▶ Fundamental to **theoretical** Computer Science, and
 - ▶ Eminently **practical** for software developers.

Lecture Outline

- ▶ Introduction (*5 minutes*)
- ▶ Course Logistics (*10 minutes*)
 - ▶ Course staff contact info
 - ▶ Electronic resources
 - ▶ Grading and homeworks
- ▶ Overview of Programming Languages (*10 minutes*)
 - ▶ What is a programming language?
 - ▶ Kinds of languages
 - ▶ What kinds of software tools are out there?
 - ▶ A bit of history
 - ▶ Reading assignment
- ▶ Introduction to OCaml (next set of slides) (*50 minutes*)

Course staff contact info

Sameer Sundresh (Instructor)

- ▶ Office: 0207 SC
- ▶ Office Hours: 12:45-2:45pm, Friday
- ▶ Email: sundresh@uiuc.edu

Chris Osborn (TA)

- ▶ Office: 0207 SC
- ▶ Office Hours: 12:45-2:45pm, Monday
- ▶ Email: cosborn3@uiuc.edu

Online hours (see website for details)

- ▶ e-mail & newsgroups: any time
- ▶ Jabber IM Wednesday 6-8pm CDT & during office hours

Electronic resources

- Course Web Page** <http://www.cs.uiuc.edu/class/su07/cs421/>.
Check announcements daily.
Includes lectures slides + **supplementary notes**.
Also includes useful resources and course policies.
- Newsgroup** `class.cs421`
Read newsgroup daily also.
- Languages** OCaml, a dialect of ML.
See <http://caml.inria.fr/ocaml>
- Computer Lab** EWS labs: <http://www.ews.uiuc.edu/>
NOTE: Contact TA if you have no EWS account.

Grading and homeworks

- ▶ Homework assignments (MPs): 50% of grade, submitted electronically
 - ▶ Weekly assignments, assigned/due on Tuesdays
 - ▶ Machine Problems (MPs) in OCaml
 - ▶ Some assignments include a written/theoretical part
 - ▶ Individual work (ok to clarify with others)
- ▶ Midterm: 20%, in class July 2nd
- ▶ Final: 30%, August 4th, 1-4pm

4th Credit hour project

- ▶ For students registered for 4 hours
- ▶ Can work in groups of up to 3 people
- ▶ Potential projects detailed on course web page
- ▶ Fairly flexible – you should pick something you are interested in, to make it more fun
- ▶ Project proposals are due by **June 14**, but can be in earlier if you want to get a head start
- ▶ 1/2 page (per project) status reports due every 2 weeks via e-mail

What is a programming language?

- ▶ A notation for software designs (**syntax**)
- ▶ A definition of what this notation means/how programs behave (**semantics**)
- ▶ How are languages defined?
 - ▶ Sometimes they're defined precisely, using math as a meta-language (e.g., "*The Definition of Standard ML*").
 - ▶ More often they're defined loosely, which leads to ambiguities and complexity (e.g., C++).
- ▶ How do you make a *good* programming language definition?

Kinds of languages

- ▶ Assembly (x86, MIPS, PowerPC, ...)
- ▶ Imperative (C, Pascal, ...)
- ▶ Object-Oriented (C++, Java, Smalltalk, ...)
- ▶ Functional (**OCaml**, Scheme, Haskell, ...)
 - ▶ *Functions* as first-class objects
 - ▶ Programs generally behave as functions: same inputs \Rightarrow same outputs
- ▶ Declarative Logic Programming (Prolog, Maude, ...)
 1. Define a logical system (equations, inference rules, etc.)
 2. Pose a query
 3. Programming system solves for a result

What kinds of software tools are out there?

- ▶ Covered in this class:
 - ▶ Compilers
 - ▶ Interpreters
 - ▶ Type checkers
 - ▶ Code generators
- ▶ Also:
 - ▶ Debuggers
 - ▶ Static analysis tools
 - ▶ Model checkers
 - ▶ Theorem provers

A bit of history

THEORY		lambda calculus		parsing		type systems		verification		CS 421
combinatory logic						semantics				
1920s	1930s	1940s	1950s	1960s	1970s	1980s	1990s	2000s		
PRACTICE		computers		LISP	Simula	ML	Web	CS 421		
				FORTRAN	C	C++	Java			

Note: this is an approximate timeline, many things are left out.

Reading assignment

- ▶ Look this stuff up online (Wikipedia, etc.)
- ▶ Read the supplementary notes on the course website:
<http://www.cs.uiuc.edu/class/su07/cs421/lectures/>.