# CS 421 – Programming Languages and Compilers

Welcome!

Today's class:

- Staff
- Why 421?
- Class structure and policies
- Intro to OCaml
- Assignment of hwk I (due Friday)

#### Staff

Professor: Sam Kamin, 4237 SC, 3-8069

Office hrs: TBD

TAs: Dongyun Jin (TBD)

Baris Aktemur (TBD)

William Mansky (TBD)

Web page:

http://www.cs.uiuc.edu/class/cs421/

Newsgroup: news.cs.uiuc.edu - class.cs421

## Why 421?

#### Why learn about compilers?

- Complete picture of how programs go from keyboard to execution
- Understand translation from high-level language to machine language
- Learn to build compilers and other programs that process structured input
- Learn interesting algorithms

## Why 421?

Why learn about languages?

- Increase ability to learn new languages
- Learn correct terminology for describing languages
- Become better programmers by seeing different perspectives on programming

## Class structure

- Lectures Wednesday and Friday. Notes posted after class. Strongly advise taking notes in class.
- Weekly assignments
- Programming in OCaml
- Grades
- Cheating
- Lateness

#### Class structure

#### Topics:

- First half: Compilers
  - First 2 1/2 classes: OCaml
- Second half: Languages

Full details on web page

- Functional programming language
  - One of the two leading language paradigms (the other is object-oriented)
  - Defined mainly by no assignment statements
    - ► Heavy use of dynamically-allocated data structures and recursion
- Everything we will do in the first half of the class could also be done in Java, but:
  - OCaml notationally much more concise
  - Using OCaml now will prepare you for more advanced uses of OCaml in second half

Interactive system

Load files: #use "filename";;

Define variables and functions

# Let 
$$g = y + x$$
; Tava: int  $f(Int y)$ ;  
 $f(0) \Rightarrow 13$  return  $y$ 

Functions of multiple arguments:

- Arithmetic and comparison operators: usual.
- ▶ Boolean operators: &&, ||, not =  $\int_{-\infty}^{\infty} x_{o} t^{-1} dt$
- Conditional expressions

Duse of parentheses Java:

Only for precedence

$$(f \times) + 3 \qquad f(x+3) \qquad (f \times) + 3$$

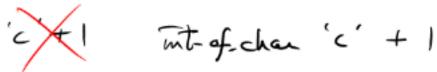
$$f(x+3)$$

Prece dence Function appli.

Strings:"....", ^ for concatenation. (String module contains length, get. etc.)

# Open String;; # length "...-";;

Characters: 'c' (int\_of\_char, char\_of\_int)



Printing to console: print\_int, print\_string

Sequencing: use ;.

Comments: (\* ... \*)

Licke "old-style" ( comments: /\* \*/

except they rest: (\* ... (\* ... (\* ... 4) ... \*)

/\* ... /\* --- \*/ ... \*/

Recursive function definitions

let rec f X = ----- f (x-1) ---

## Homework 1 – due Friday

Define a few simple functions.