

CS 4110

# Programming Languages & Logics

Lecture 38  
Typed Assembly Language

30 November 2012

# Schedule

---

## Monday

- Typed Assembly Language

## Wednesday

- Polymorphism
- Stack Types

## Today

- Compilation
- Course Review

# Certified Compilation

---

An *automatic* way to generate certifiable low-level code

Type safety implies memory, security properties

# Certified Compilation

---

An *automatic* way to generate certifiable low-level code

Type safety implies memory, security properties

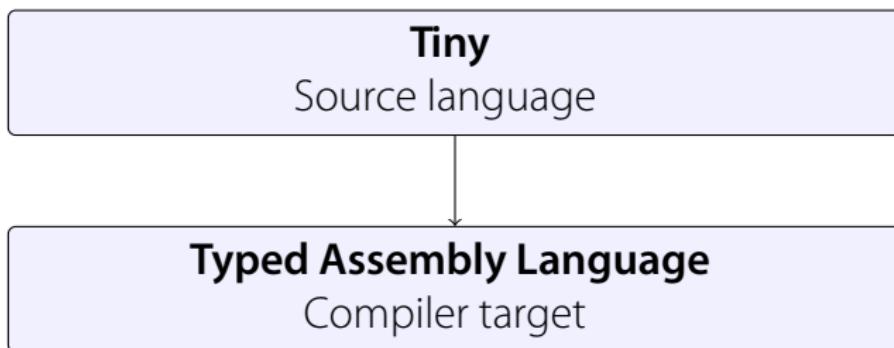
**Tiny**

Source language

# Certified Compilation

An *automatic* way to generate certifiable low-level code

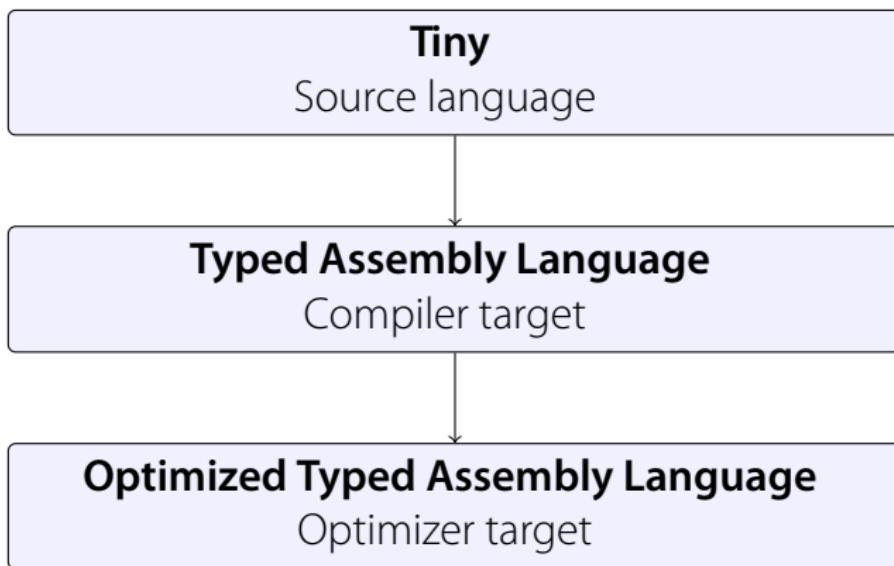
Type safety implies memory, security properties



# Certified Compilation

An *automatic* way to generate certifiable low-level code

Type safety implies memory, security properties



# The Tiny Language

---

## Features

- Integer expressions
- Conditionals
- Recursive functions
- Function pointers (but no closures)
- A strong, static type system

# The Tiny Language

## Features

- Integer expressions
- Conditionals
- Recursive functions
- Function pointers (but no closures)
- A strong, static type system

## Example program

```
letrec fun fact (n:int) : int =  
    if n = 0 then 1 else n * fact (n - 1)  
in fact (42)
```

# Tiny Type System

---

## Expressions

$$\Phi \vdash x : \Phi(x)$$

# Tiny Type System

---

## Expressions

$$\Phi \vdash x : \Phi(x)$$

$$\Phi \vdash f : \Phi(f)$$

# Tiny Type System

---

## Expressions

$$\Phi \vdash x : \Phi(x)$$
$$\Phi \vdash f : \Phi(f)$$
$$\Phi \vdash n : \text{int}$$

# Tiny Type System

## Expressions

$$\Phi \vdash x : \Phi(x)$$

$$\Phi \vdash f : \Phi(f)$$

$$\Phi \vdash n : \text{int}$$

$$\frac{\Phi \vdash e_1 : \text{int} \quad \Phi \vdash e_2 : \text{int}}{\Phi \vdash e_1 + e_2 : \text{int}}$$

# Tiny Type System

## Expressions

$$\Phi \vdash x : \Phi(x)$$

$$\Phi \vdash f : \Phi(f)$$

$$\Phi \vdash n : \text{int}$$

$$\frac{\Phi \vdash e_1 : \text{int} \quad \Phi \vdash e_2 : \text{int}}{\Phi \vdash e_1 + e_2 : \text{int}}$$

$$\frac{\Phi \vdash e_1 : \tau_1 \rightarrow \tau_2 \quad \Phi \vdash e_2 : \tau_1}{\Phi \vdash e_1 \ e_2 : \tau_2}$$

# Tiny Type System

## Expressions

$$\Phi \vdash x : \Phi(x)$$

$$\Phi \vdash f : \Phi(f)$$

$$\Phi \vdash n : \text{int}$$

$$\frac{\Phi \vdash e_1 : \text{int} \quad \Phi \vdash e_2 : \text{int}}{\Phi \vdash e_1 + e_2 : \text{int}}$$

$$\frac{\Phi \vdash e_1 : \tau_1 \rightarrow \tau_2 \quad \Phi \vdash e_2 : \tau_1}{\Phi \vdash e_1 \ e_2 : \tau_2}$$

$$\frac{\Phi \vdash e_1 : \text{int} \quad \Phi \vdash e_2 : \tau \quad \Phi \vdash e_3 : \tau}{\Phi \vdash \text{if } e_1 = 0 \text{ then } e_2 \text{ else } e_3 : \tau}$$

# Tiny Type System

## Expressions

$$\Phi \vdash x : \Phi(x)$$

$$\Phi \vdash f : \Phi(f)$$

$$\Phi \vdash n : \text{int}$$

$$\frac{\Phi \vdash e_1 : \text{int} \quad \Phi \vdash e_2 : \text{int}}{\Phi \vdash e_1 + e_2 : \text{int}}$$

$$\frac{\Phi \vdash e_1 : \tau_1 \rightarrow \tau_2 \quad \Phi \vdash e_2 : \tau_1}{\Phi \vdash e_1 \ e_2 : \tau_2}$$

$$\frac{\Phi \vdash e_1 : \text{int} \quad \Phi \vdash e_2 : \tau \quad \Phi \vdash e_3 : \tau}{\Phi \vdash \text{if } e_1 = 0 \text{ then } e_2 \text{ else } e_3 : \tau}$$

$$\frac{\Phi \vdash e_1 : \tau_1 \quad \Phi, x : \tau_1 \vdash e_2 : \tau_2}{\Phi \vdash \text{let } x = e_1 \text{ in } e_2 : \tau_1}$$

# Tiny Type System

---

## Declarations

$$\frac{\Phi, x : \tau_1 \vdash e : \tau_2}{\Phi \vdash \text{fun } f(x : \tau_1) : \tau_2 = e : (f : \tau_1 \rightarrow \tau_2)}$$

# Tiny Type System

## Declarations

$$\frac{\Phi, x : \tau_1 \vdash e : \tau_2}{\Phi \vdash \text{fun } f(x : \tau_1) : \tau_2 = e : (f : \tau_1 \rightarrow \tau_2)}$$

## Programs

$$\frac{\Phi = f_1 : \tau_{11} \rightarrow \tau_{12}, \dots, f_k : \tau_{k1} \rightarrow \tau_{k2} \quad \Phi \vdash d_i : (f_i : \tau_{i1} \rightarrow \tau_{i2}) \quad \Phi \vdash e : \text{int}}{\vdash \text{letrec } d_1 \cdots d_k \text{ in } e}$$

# Tiny Type System

## Declarations

$$\frac{\Phi, x : \tau_1 \vdash e : \tau_2}{\Phi \vdash \text{fun } f(x : \tau_1) : \tau_2 = e : (f : \tau_1 \rightarrow \tau_2)}$$

## Programs

$$\frac{\Phi = f_1 : \tau_{11} \rightarrow \tau_{12}, \dots, f_k : \tau_{k1} \rightarrow \tau_{k2} \quad \Phi \vdash d_i : (f_i : \tau_{i1} \rightarrow \tau_{i2}) \quad \Phi \vdash e : \text{int}}{\vdash \text{letrec } d_1 \cdots d_k \text{ in } e}$$

Exercise: Chat that *fact* is well typed

# Type-preserving Compilation

- Most compilers consist of a series of transformations
- In our compiler, each step will propagate types
- A transformation consists of two sub-translations:
  - ▶ From source types to target types
  - ▶ From source terms to target terms
- After each step, can typecheck the intermediate output code to detect errors in the compiler
- Key correctness property will be that transformations preserve types (as well as semantics)

# Type Translation

---

$\mathcal{T}[\cdot]$  maps Tiny types to TAL Types

# Type Translation

---

$\mathcal{T}[\cdot]$  maps Tiny types to TAL Types

- Integers:  $\mathcal{T}[\text{int}] = \text{int}$

# Type Translation

$\mathcal{T}[\cdot]$  maps Tiny types to TAL Types

- Integers:  $\mathcal{T}[\text{int}] = \text{int}$
- Functions:
  - ▶ Caller pushes argument and return address on stack
  - ▶ Callee pops the return address and argument and places result in  $r_a$

$$\mathcal{T}[\tau_1 \rightarrow \tau_2] = \forall \rho. \{sp : \mathcal{K}[\tau_2, \rho] :: \mathcal{T}[\tau_1] :: \rho\} \rightarrow \{\}$$

where

$$\mathcal{K}[\tau, \sigma] = \{sp : \sigma, r_a : \mathcal{T}[\tau]\} \rightarrow \{\}$$

$$\mathcal{K}[\tau] = \{r_a : \mathcal{T}[\tau]\} \rightarrow \{\}$$

# Expression Translation

---

$\mathcal{E}[\cdot]$  maps Tiny expressions (really typing derivations) to labeled code blocks

# Expression Translation

$\mathcal{E}[\cdot]$  maps Tiny expressions (really typing derivations) to labeled code blocks

- We use the stack extensively:
  - ▶ To evaluate an expression, evaluate subexpressions then push them onto the stack
  - ▶ Return final value in  $r_a$
  - ▶  $M$  maps expression variables to stack offsets
  - ▶  $I(M)$  increments the offset associated with each variable in  $M$
- Overall, uses just two registers,  $r_a$  and  $r_t$ , and the stack
- Shape of the translation is  $\mathcal{E}[e]_{M,\sigma} = J$ , where  $J$  is a sequence of typed, labeled blocks.
- Write  $L_f$  for the TAL label of function  $f$

# Expression Translation

## Expression variables

$$\mathcal{E}[\![x]\!]_{M,\sigma} = \text{sld } r_a, M(x)$$

# Expression Translation

Expression variables

$$\mathcal{E}[\![x]\!]_{M,\sigma} = \text{sld } r_a, M(x)$$

Function variables

$$\mathcal{E}[\![f]\!]_{M,\sigma} = \text{mov } r_a, L_f$$

# Expression Translation

Expression variables

$$\mathcal{E}[\![x]\!]_{M,\sigma} = \text{sld } r_a, M(x)$$

Function variables

$$\mathcal{E}[\![f]\!]_{M,\sigma} = \text{mov } r_a, L_f$$

Integers

$$\mathcal{E}[\![n]\!]_{M,\sigma} = \text{mov } r_a, n$$

# Expression Translation

Expression variables

$$\mathcal{E}[\![x]\!]_{M,\sigma} = \text{sld } r_a, M(x)$$

Function variables

$$\mathcal{E}[\![f]\!]_{M,\sigma} = \text{mov } r_a, L_f$$

Integers

$$\mathcal{E}[\![n]\!]_{M,\sigma} = \text{mov } r_a, n$$

Addition

$$\begin{aligned}\mathcal{E}[\![e_1 + e_2]\!]_{M,\sigma} &= \mathcal{E}[\![e_1]\!]_{M,\sigma} \\ &\quad \text{push } r_a \\ &\quad \mathcal{E}[\![e_2]\!]_{I(M),\text{int} :: \sigma} \\ &\quad \text{pop } r_t \\ &\quad \text{add } r_a, r_t, r_a\end{aligned}$$

# Expression Translation

## Application

$$\begin{aligned}\mathcal{E}[e_1 \ e_2]_{M,\sigma} &= \mathcal{E}[e_1]_{M,\sigma} \\ &\quad \text{push } r_a \\ \mathcal{E}[e_2]_{I(M), \mathcal{T}[\tau_1 \rightarrow \tau_2]} &:: \sigma \\ &\quad \text{pop } r_t \\ &\quad \text{push } r_a \\ &\quad \text{push } L_r[\rho] \\ &\quad \text{jmp } r_t[\sigma] \\ L_r : \forall \rho. \mathcal{K}[\tau_2, \sigma] &\end{aligned}$$

where  $\Phi \vdash e_1 : \tau_1 \rightarrow \tau_2$

and  $L_r$  fresh

# Expression Translation

## Conditional

$$\begin{aligned}\mathcal{E}[\text{if } e_1 = 0 \text{ then } e_2 \text{ else } e_3]_{M,\sigma} &= \mathcal{E}[e_1]_{M,\sigma} \\ &\quad \text{bneq } r_a, L_{\text{else}}[\rho] \\ &\quad \mathcal{E}[e_2]_{M,\sigma} \\ &\quad \text{jmp } L_{\text{end}}[\rho] \\ L_{\text{else}} &: \forall \rho. \{sp : \sigma\} \rightarrow \{\} \\ &\quad \mathcal{E}[e_3]_{M,\sigma} \\ &\quad \text{jmp } L_{\text{end}}[\rho] \\ L_{\text{end}} &: \forall \rho. \mathcal{K}[\tau, \sigma]\end{aligned}$$

where  $\Phi \vdash e_2 : \tau$  and  $\Phi \vdash e_3 : \tau$

and  $L_{\text{else}}$  and  $L_{\text{end}}$  fresh

# Declaration Translation

## Function

$$\mathcal{F}[\![\text{fun } f(x : \tau_1) : \tau_2 = e]\!] = L_f : \mathcal{T}[\![\tau_1 \rightarrow \tau_2]\!]$$
$$\mathcal{E}[\![e_2]\!]_x := 2, \mathcal{K}[\![\tau_2, \rho]\!] :: \mathcal{T}[\![\tau_1]\!] :: \rho$$

pop  $r_t$

sfree 1

jmp  $r_t$

# Program Translation

## Program

$$\mathcal{P}[\![\text{letrec } d_1 \cdot d_k \text{ in } e]\!] = \mathcal{F}[\![d_1]\!]$$

⋮

$$\mathcal{F}[\![d_k]\!]$$

$$L_{main} : \forall \rho. \{sp : \mathcal{K}[\![\text{int}, \rho]\!] :: \rho\} \rightarrow \{\}$$
$$\mathcal{E}[\![e]\!]_{\cdot, \mathcal{K}[\![\text{int}, \rho]\!] :: \rho}$$

pop  $r_t$   
jmp  $r_t$

To run the program, push return address on stack and jump to  $L_{main}$

Result is returned in  $r_a$

# Factorial

```
Lfact: ∀ρ. {sp : K[int] :: int :: ρ}
    sld ra,2
    bneq ra,Lelse[ρ]
    mov ra,1
    jmp Lend
Lelse: ∀ρ. {sp : K[int] :: int :: ρ}
    sld ra,2
    push ra
    mov ra,Lfact
    push ra
    sld ra,4
    push ra
    mov ra,1
    pop rt
    sub ra,rt,ra
    pop rt
    push Lr[ρ]
    jmp rt[int :: K[int, ρ] :: int :: ρ]
```

# Factorial

$L_{fact} : \forall \rho. \{sp : \mathcal{K}[\text{int}] :: \text{int} :: \rho\}$

sld  $r_a, 2$

bneq  $r_a, L_{else}[\rho]$

mov  $r_a, 1$

jmp  $L_{end}$

$L_{else} : \forall \rho. \{sp : \mathcal{K}[\text{int}] :: \text{int} :: \rho\}$

sld  $r_a, 2$

push  $r_a$

mov  $r_a, L_{fact}$

push  $r_a$

sld  $r_a, 4$

push  $r_a$

mov  $r_a, 1$

pop  $r_t$

sub  $r_a, r_t, r_a$

pop  $r_t$

push  $L_r[\rho]$

jmp  $r_t[\text{int} :: \mathcal{K}[\text{int}, \rho] :: \text{int} :: \rho]$

$L_r : \forall \rho. \{sp : \mathcal{K}[\text{int}] :: \text{int} :: \rho, r_a : \text{int}\}$

pop  $r_t$

mul  $r_a, r_t, r_t$

jmp  $L_{end}[\rho]$

$L_{end} : \forall \rho. \{sp : \mathcal{K}[\text{int}] :: \text{int} :: \rho, r_a : \text{int}\}$

pop  $r_t$

sfree 1

jmp  $r_t$

# Compiler properties

## Theorem

*If  $\vdash P$  then there exists  $\Psi$  such that  $\vdash \mathcal{P}[\![P]\!] : \Psi$*

Proof by induction on  $P$ ...

**Main idea:** We don't have to trust the compiler because we can typecheck the assembly code emitted as output!

# Optimizations

---

Almost any compiler will produce better code than ours! (But how many compilers fit on four slides?)

The type system makes it possible to implement many standard optimizations to generate better code:

- Instruction selection and scheduling
- Register allocation
- Common subexpression elimination
- Redundant load/store elimination
- Strength reduction
- Loop-invariant removal
- Tail-calls
- and others...

Types do not interfere with the most common optimizations

# Tail-call Optimization

---

- A crucial optimization for functional languages
- Applies when the final operation in a function  $f$  is a call to another function  $g$
- Instead of having  $f$  push the return address onto the stack and engage in the normal calling sequence,  $f$  pops all of its temporary values, jumps directly to  $g$  and never returns!

# Tail-call Optimization

## Unoptimized code

```
Lf: ∀ρ. {sp : K[τ, ρ] :: τf :: ρ, ra : τg} → {}  
...  
    salloc 2          % setup stack frame  
    sst Lr          % push return address  
    sst ra,2         % push argument  
    jmp Lg[τ :: τf :: ρ]  
Lr: ∀ρ. {sp : τ :: τf :: ρ, ra : τ} → {}  
    pop rt          % pop return address  
    sfree 1           % discard f's argument  
    jmp rt          % return
```

## Optimized code

```
Lf: ∀ρ. {sp : K[τ, ρ] :: τf :: ρ, ra : τg} → {}  
...  
    sst ra,2  
    jmp Lg[ρ]        % g returns to f's caller
```

# Course Review

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

## Home Syllabus Schedule Resources

Date	Topic	Notes	Reading	Assignments
22 August	Introduction	<a href="#">PDF</a>	Winskel 1	
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Inductive definitions and proofs	<a href="#">PDF</a>		
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational semantics	<a href="#">PDF</a>		HW3 out
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encodings	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			
10 October	Types	<a href="#">PDF</a>		

12 October	More types	<a href="#">PDF</a>		HW7 out
15 October	Record types	<a href="#">PDF</a>		
17 October	Subtyping	<a href="#">PDF</a>		
19 October	Polymorphism	<a href="#">PDF</a>		HW8 out
25 October	More polymorphism	<a href="#">PDF</a>		
27 October	Type inference	<a href="#">PDF</a>		
29 October	Propositions-as-types	<a href="#">PDF</a>		HW9 out
1 November	Existential types	<a href="#">PDF</a>		
3 November	Objects	<a href="#">PDF</a>		
5 November	Featherweight Java	<a href="#">PDF</a>		HW10 out
8 November	Featherweight Java types	<a href="#">PDF</a>		
10 November	Review	<a href="#">PDF</a>		
12 November	Preliminary Exam II			
15 November	Abstract interpretation	<a href="#">PDF</a>		
17 November	Concurrency	<a href="#">PDF</a>		
19 November	More concurrency	<a href="#">PDF</a>		HW11 out
22 November	Language-based security	<a href="#">PDF</a>		
24 November	Coq	<a href="#">PDF</a>		
26 November	No class (Thanksgiving)			
29 November	More Coq	<a href="#">PDF</a>		
1 December	Current trends in PL research	<a href="#">PDF</a>		
3 December	Review	<a href="#">PDF</a>		
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

Navigation				
Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries &			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational semantics	<a href="#">PDF</a>		HW3 out
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encodings	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			
10 October	Types	<a href="#">PDF</a>		
12 October	More types			
15 October	Record types		<a href="#">PDF</a>	
17 October	Subtyping		<a href="#">PDF</a>	
19 October	Polymorphism		<a href="#">PDF</a>	HW8 out
25 October	More polymorphism		<a href="#">PDF</a>	
27 October	Type inference		<a href="#">PDF</a>	
29 October	Propositions-as-types		<a href="#">PDF</a>	HW9 out
1 November	Existential types		<a href="#">PDF</a>	
3 November	Objects		<a href="#">PDF</a>	
5 November	Featherweight Java		<a href="#">PDF</a>	HW10 out
8 November	Featherweight Java types		<a href="#">PDF</a>	
10 November	Review		<a href="#">PDF</a>	
12 November	Preliminary Exam II			
15 November	Abstract interpretation		<a href="#">PDF</a>	
17 November	Concurrency		<a href="#">PDF</a>	
19 November	More concurrency		<a href="#">PDF</a>	HW11 out
22 November	Language-based security		<a href="#">PDF</a>	
24 November	Coq		<a href="#">PDF</a>	
26 November	No class (Thanksgiving)			
29 November	More Coq		<a href="#">PDF</a>	
1 December	Current trends in PL research		<a href="#">PDF</a>	
3 December	Review		<a href="#">PDF</a>	
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

Home Syllabus Schedule Resources				
Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries & Operational Semantics			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encodings	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			
10 October	Types	<a href="#">PDF</a>		
12 October	More types		<a href="#">PDF</a>	HW7 out
15 October	Record types		<a href="#">PDF</a>	
17 October	Subtyping		<a href="#">PDF</a>	
19 October	Polymorphism		<a href="#">PDF</a>	HW8 out
25 October	More polymorphism		<a href="#">PDF</a>	
27 October	Type inference		<a href="#">PDF</a>	
29 October	Propositions-as-types		<a href="#">PDF</a>	HW9 out
1 November	Existential types		<a href="#">PDF</a>	
3 November	Objects		<a href="#">PDF</a>	
5 November	Featherweight Java		<a href="#">PDF</a>	HW10 out
8 November	Featherweight Java types		<a href="#">PDF</a>	
10 November	Review		<a href="#">PDF</a>	
12 November	Preliminary Exam II			
15 November	Abstract interpretation		<a href="#">PDF</a>	
17 November	Concurrency		<a href="#">PDF</a>	
19 November	More concurrency		<a href="#">PDF</a>	HW11 out
22 November	Language-based security		<a href="#">PDF</a>	
24 November	Coq		<a href="#">PDF</a>	
26 November	No class (Thanksgiving)			
29 November	More Coq		<a href="#">PDF</a>	
1 December	Current trends in PL research		<a href="#">PDF</a>	
3 December	Review		<a href="#">PDF</a>	
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

## Home   Syllabus   Schedule   Resources

Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries &			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
6 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			
10 October	Types	<a href="#">PDF</a>		

12 October	More types	<a href="#">PDF</a>		HW7 out
15 October	Record types	<a href="#">PDF</a>		
17 October	Subtyping	<a href="#">PDF</a>		
19 October	Polymorphism	<a href="#">PDF</a>		HW8 out
25 October	More polymorphism	<a href="#">PDF</a>		
27 October	Type inference	<a href="#">PDF</a>		
29 October	Propositions-as-types	<a href="#">PDF</a>		HW9 out
1 November	Existential types	<a href="#">PDF</a>		
3 November	Objects	<a href="#">PDF</a>		
5 November	Featherweight Java	<a href="#">PDF</a>		HW10 out
8 November	Featherweight Java types	<a href="#">PDF</a>		
10 November	Review	<a href="#">PDF</a>		
12 November	Preliminary Exam II			
15 November	Abstract interpretation	<a href="#">PDF</a>		
17 November	Concurrency	<a href="#">PDF</a>		
19 November	More concurrency	<a href="#">PDF</a>		HW11 out
22 November	Language-based security	<a href="#">PDF</a>		
24 November	Coq	<a href="#">PDF</a>		
26 November	No class (Thanksgiving)			
29 November	More Coq	<a href="#">PDF</a>		
1 December	Current trends in PL research	<a href="#">PDF</a>		
3 December	Review	<a href="#">PDF</a>		
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

Home Syllabus Schedule Resources				
Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries & Operational Semantics			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
6 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			
10 October	Types	<a href="#">PDF</a>		
12 October	More types		<a href="#">PDF</a>	HW7 out
15 October	Record types		<a href="#">PDF</a>	
17 October	Subtyping		<a href="#">PDF</a>	
19 October	Polymorphism		<a href="#">PDF</a>	HW8 out
25 October	More polymorphism		<a href="#">PDF</a>	
27 October	Type inference		<a href="#">PDF</a>	
29 October	Propositions-as-types		<a href="#">PDF</a>	
1 November	Existential types		<a href="#">PDF</a>	
3 November	Objects		<a href="#">PDF</a>	
5 November	Featherweight Java		<a href="#">PDF</a>	HW10 out
8 November	Featherweight Java types		<a href="#">PDF</a>	
10 November	Review		<a href="#">PDF</a>	
12 November	Preliminary Exam II			
15 November	Abstract interpretation		<a href="#">PDF</a>	
17 November	Concurrency		<a href="#">PDF</a>	
19 November	More concurrency		<a href="#">PDF</a>	HW11 out
22 November	Language-based security		<a href="#">PDF</a>	
24 November	Coq		<a href="#">PDF</a>	
26 November	No class (Thanksgiving)			
29 November	More Coq		<a href="#">PDF</a>	
1 December	Current trends in PL research		<a href="#">PDF</a>	
3 December	Review		<a href="#">PDF</a>	
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

Home Syllabus Schedule Resources				
Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries & Operational Semantics			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)	Fall Break		
10 October	Types	<a href="#">PDF</a>		
12 October	More types		<a href="#">PDF</a>	HW7 out
15 October	Record types		<a href="#">PDF</a>	
17 October	Subtyping		<a href="#">PDF</a>	
19 October	Polymorphism		<a href="#">PDF</a>	HW8 out
25 October	More polymorphism		<a href="#">PDF</a>	
27 October	Type inference		<a href="#">PDF</a>	
29 October	Propositions-as-types		<a href="#">PDF</a>	
1 November	Existential types		<a href="#">PDF</a>	
3 November	Objects		<a href="#">PDF</a>	
5 November	Featherweight Java		<a href="#">PDF</a>	HW10 out
8 November	Featherweight Java types		<a href="#">PDF</a>	
10 November	Review		<a href="#">PDF</a>	
12 November	Preliminary Exam II			
15 November	Abstract interpretation		<a href="#">PDF</a>	
17 November	Concurrency		<a href="#">PDF</a>	
19 November	More concurrency		<a href="#">PDF</a>	HW11 out
22 November	Language-based security		<a href="#">PDF</a>	
24 November	Coq		<a href="#">PDF</a>	
26 November	No class (Thanksgiving)			
29 November	More Coq		<a href="#">PDF</a>	
1 December	Current trends in PL research		<a href="#">PDF</a>	
3 December	Review		<a href="#">PDF</a>	
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics

MWF 9:00-9:50

Upson 111



Cornell University  
Department of  
Computer Science

## Home Syllabus

## Schedule

## Resources

Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries & Operational Semantics			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			Fall Break
10 October	Types	<a href="#">PDF</a>		

12 October	More types	<a href="#">PDF</a>		HW7 out
15 October	Record types	<a href="#">PDF</a>		
17 October	Subtyping	<a href="#">PDF</a>		
19 October	Polymorphism	<a href="#">PDF</a>		
25 October	More polymorphism	<a href="#">PDF</a>		
27 October	Type Systems & Type Inference	<a href="#">PDF</a>		
29 October	Propositions-as-types	<a href="#">PDF</a>		HW9 out
1 November	Existential types	<a href="#">PDF</a>		
3 November	Objects	<a href="#">PDF</a>		
5 November	Feathernweight Java	<a href="#">PDF</a>		HW10 out
8 November	Feathernweight Java types	<a href="#">PDF</a>		
10 November	Review	<a href="#">PDF</a>		
12 November	Preliminary Exam II			
15 November	Abstract interpretation	<a href="#">PDF</a>		
17 November	Concurrency	<a href="#">PDF</a>		
19 November	More concurrency	<a href="#">PDF</a>		HW11 out
22 November	Language-based security	<a href="#">PDF</a>		
24 November	Coq	<a href="#">PDF</a>		
26 November	No class (Thanksgiving)			
29 November	More Coq	<a href="#">PDF</a>		
1 December	Current trends in PL research	<a href="#">PDF</a>		
3 December	Review	<a href="#">PDF</a>		
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

## Home Syllabus Schedule Resources

Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries & Operational Semantics			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMPL	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMPL properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuations	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)			Fall Break
10 October	Types	<a href="#">PDF</a>		

12 October	More types	<a href="#">PDF</a>	
15 October	Record types	<a href="#">PDF</a>	
17 October	Subtyping	<a href="#">PDF</a>	
19 October	Polymorphism	<a href="#">PDF</a>	
25 October	More polymorphism	<a href="#">PDF</a>	
27 October	Type Systems & Static Analysis	<a href="#">PDF</a>	
29 October	Propositions-as-types	<a href="#">PDF</a>	HW9 out
1 November	Existential types	<a href="#">PDF</a>	
3 November	Objects	<a href="#">PDF</a>	
5 November	Feathernweight Java	<a href="#">PDF</a>	HW10 out
8 November	Feathernweight Java types	<a href="#">PDF</a>	
10 November	Review	<a href="#">PDF</a>	
12 November	Preliminary Exam II		
15 November	Abstract interpretation	<a href="#">PDF</a>	
17 November	Concurrency	<a href="#">PDF</a>	
19 November	More concurrency	<a href="#">PDF</a>	HW11 out
22 November	Language-based security	<a href="#">PDF</a>	
24 November	Coq	<a href="#">PDF</a>	
26 November	No class (Thanksgiving)		
29 November	More Coq	<a href="#">PDF</a>	
1 December	Current trends in PL research	<a href="#">PDF</a>	
3 December	Review	<a href="#">PDF</a>	
13 December	Final Exam		

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

Home Syllabus Schedule Resources				
Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries &			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMPL	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMPL properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary	Preliminary Exam I		
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuation	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)	Fall Break		
10 October	Types	<a href="#">PDF</a>		
12 October	More types		<a href="#">PDF</a>	
15 October	Record types		<a href="#">PDF</a>	
17 October	Subtyping		<a href="#">PDF</a>	
19 October	Polymorphism		<a href="#">PDF</a>	
25 October	More polymorphism		Type Systems & Static Analysis	
27 October	Type inference		<a href="#">PDF</a>	
29 October	Propositions-as-types		<a href="#">PDF</a>	HW9 out
1 November	Existential types		<a href="#">PDF</a>	
3 November	Objects		<a href="#">PDF</a>	
5 November	Feathertweight Java		<a href="#">PDF</a>	HW10 out
8 November	Feathertweight Java types		<a href="#">PDF</a>	
10 November	Review		<a href="#">PDF</a>	
12 November	Preliminary Exam II			
15 November	Abstract interpretation		<a href="#">PDF</a>	
17 November	Concurrency		<a href="#">PDF</a>	
19 November	More concurrency		<a href="#">PDF</a>	
22 November	Language-knug security		<a href="#">PDF</a>	
24 November	Coq	Advanced Topics		
26 November	No class (Thanksgiving)			
29 November	More Coq		<a href="#">PDF</a>	
1 December	Current trends in PL research		<a href="#">PDF</a>	
3 December	Review		<a href="#">PDF</a>	
13 December	Final Exam			

# CS 4110 (Fall 2010)

Programming Languages and Logics  
MWF 9:00-9:50  
Upson 111



Cornell University  
Department of  
Computer Science

Home Syllabus Schedule Resources				
Date	Topic	Notes	Reading	Assignments
22 August	Mathematical Preliminaries &			
24 August	Small-step semantics	<a href="#">PDF</a>	Winskel 2	HW1 out
27 August	Induction			
29 August	Large-step semantics	<a href="#">PDF</a>		
31 August	IMP	<a href="#">PDF</a>		HW2 out
3 September	No class (Labor Day)			
5 September	IMP properties	<a href="#">PDF</a>		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	<a href="#">PDF</a>		
12 September	Axiomatic semantics	<a href="#">PDF</a>		
14 September	Hoare logic	<a href="#">PDF</a>		HW4 out
17 September	$\lambda$ -calculus	<a href="#">PDF</a>		
19 September	More $\lambda$ -calculus	<a href="#">PDF</a>		
21 September	$\lambda$ -calculus encoding	<a href="#">PDF</a>		HW5 out
24 September	Recursion	<a href="#">PDF</a>		
26 September	Definitional translation	<a href="#">PDF</a>		
28 September	Review	<a href="#">PDF</a>		
1 October	Preliminary Exam I			
3 October	Continuations	<a href="#">PDF</a>		
5 October	More continuation	<a href="#">PDF</a>		HW6 out
8 October	No class (Fall Break)	Fall Break		
10 October	Types	<a href="#">PDF</a>		
12 October	More types		<a href="#">PDF</a>	HW7 out
15 October	Record types		<a href="#">PDF</a>	
17 October	Subtyping		<a href="#">PDF</a>	
19 October	Polymorphism		<a href="#">PDF</a>	
25 October	More polymorphism			
27 October	Type inference		<a href="#">PDF</a>	
29 October	Propositions-as-types		<a href="#">PDF</a>	HW9 out
1 November	Existential types		<a href="#">PDF</a>	
3 November	Objects		<a href="#">PDF</a>	
5 November	Feathernweight Java		<a href="#">PDF</a>	HW10 out
8 November	Feathernweight Java types		<a href="#">PDF</a>	
10 November	Review		<a href="#">PDF</a>	
12 November	Preliminary Exam II			
15 November	Abstract interpretation		<a href="#">PDF</a>	
17 November	Concurrency		<a href="#">PDF</a>	
19 November	More concurrency		<a href="#">PDF</a>	HW11 out
22 November	Language-knug security		<a href="#">PDF</a>	
24 November	Coq	Advanced Topics		
26 November	No class (Thanksgiving)			
29 November	More Coq		<a href="#">PDF</a>	
1 December	Current trends in PL research		<a href="#">PDF</a>	
3 December	Review		<a href="#">PDF</a>	
13 December	Final Exam	Final Exam		

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)
- Semantics (operational, axiomatic, denotational)

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)
- Semantics (operational, axiomatic, denotational)
- $\lambda$ -calculus (basics, encodings, extensions)

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)
- Semantics (operational, axiomatic, denotational)
- $\lambda$ -calculus (basics, encodings, extensions)
- Type systems (simple, extensions, properties)

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)
- Semantics (operational, axiomatic, denotational)
- $\lambda$ -calculus (basics, encodings, extensions)
- Type systems (simple, extensions, properties)
- Advanced topics (logic, security, concurrency, DSLs, TAL)

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)
- Semantics (operational, axiomatic, denotational)
- $\lambda$ -calculus (basics, encodings, extensions)
- Type systems (simple, extensions, properties)
- Advanced topics (logic, security, concurrency, DSLs, TAL)

Expect to solve problems just like the ones we've seen throughout the course...

# Final Topics

---

- Mathematical Preliminaries (inductive definitions)
- Semantics (operational, axiomatic, denotational)
- $\lambda$ -calculus (basics, encodings, extensions)
- Type systems (simple, extensions, properties)
- Advanced topics (logic, security, concurrency, DSLs, TAL)

Expect to solve problems just like the ones we've seen throughout the course...

...and to apply the skills you've acquired to new problems too!

# Final Logistics

---

- Date: Wednesday, December 12th
- Time: 7-9:30pm
- Where: Hollister 110
- Practice: Available from Upson 4146 starting Monday
- Review: Sunday, December 9th?

# Going further

---

- Apply to attend ACM SIGPLAN PLMW

# Going further

---

- Apply to attend ACM SIGPLAN PLMW
- CS 6110 – Advanced Programming Languages

# Going further

---

- Apply to attend ACM SIGPLAN PLMW
- CS 6110 – Advanced Programming Languages
- CS 5114/6114 – Network Programming Languages

# Going further

---

- Apply to attend ACM SIGPLAN PLMW
- CS 6110 – Advanced Programming Languages
- CS 5114/6114 – Network Programming Languages
- CS 7190 – Seminar in Programming Languages

# Going further

---

- Apply to attend ACM SIGPLAN PLMW
- CS 6110 – Advanced Programming Languages
- CS 5114/6114 – Network Programming Languages
- CS 7190 – Seminar in Programming Languages
- CS 4999 – Independent Research

# Going further

---

- Apply to attend ACM SIGPLAN PLMW
- CS 6110 – Advanced Programming Languages
- CS 5114/6114 – Network Programming Languages
- CS 7190 – Seminar in Programming Languages
- CS 4999 – Independent Research

Thank you, and stay in touch!