

# CS 4110

## Programming Languages & Logics

### Lecture 1 Course Overview

24 ~~26~~ August ~~2014~~ 2016



# Programming Languages

One of the oldest fields in Computer Science...

- $\lambda$ -calculus – Church (1936)
- FORTRAN – Backus (1957)
- LISP – McCarthy (1958)
- ALGOL 60 – Backus, Naur, Perlis, & others (1960)
- Pascal – Wirth (1970)
- C – Ritchie (1972)
- Smalltalk – Kay & others (1972)
- ML – Milner and others (1978)
- C++ – Stroustrup (1982)
- Haskell – Hudak, Peyton Jones, Wadler, & others (1989)
- Java – Gosling (1995)
- C# – Microsoft (2001)
- Scala – Odersky (2003)
- F# – Syme (2005)

# Programming Languages

---

...and one of the most vibrant areas today!

PL intersects with many other areas of computing

## Current trends

- Domain-specific languages
- Static analysis and types
- Language-based security
- Verification and model checking
- Concurrency

Both theoretically and practically “meaty”

# Syllabus

# Course Staff

---

## Instructor

~~Nate Foster~~ Adrian Sampson

Office: Gates ~~432~~ 411A

Hours: ~~Mon 4-5pm and Friday 11am-12pm~~ TBA

## Teaching Assistant

~~Fran Mota~~

Office: Hours: TBA

## Web Page

[~~2014fa~~ 2016fa](http://www.cs.cornell.edu/Courses/cs4110/)

## Discussion

<http://www.piazza.com>

# Course Goals

---

- Techniques for modeling programs\* mathematically
  - ▶ Operational, axiomatic, and denotational semantics
  - ▶ Examples with advanced features
  - ▶ Reasoning principles (induction, co-induction)
- Explore applications of these techniques
  - ▶ Optimization
  - ▶ Type systems
  - ▶ Verification
- Gain experience implementing languages
  - ▶ Interpreters
  - ▶ Program transformations
  - ▶ Analysis tools
- PhD students: cover material for PL qualifying exam
- Have fun :-)

\*and whole languages!

# Prerequisites

---

## Mathematical Maturity

- Much of this class will involve formal reasoning
- Set theory, formal proofs, induction
- Most challenging topic: denotational semantics

## Programming Experience

- Comfortable using a functional language
- For undergrads: CS 3110 or equivalent

Interest (having fun is a goal! :-)

If you don't meet these prerequisites, get in touch

# CS 4110 (Fall 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



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			

Your schedule with probably  
look something like this



# CS 4110 (Fall 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



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		Chapter 2	HW1 out
27 August	Inductive semantics			
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>		HWB 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 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



Cornell University  
Department of  
Computer Science

Home

Syllabus

Schedule

Resources

Date	Topic	Notes	Reading	Assignments
22 August				
24 August	Mathematical Preliminaries & Operational Semantics			HW1 out
27 August	Inductive			
29 August	Large-step semantics	PDF		
31 August	BMP	PDF		HW2 out
3 September	No class (Labor Day)			
5 September	BMP summaries	PDF		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	PDF		
12 September	Axiomatic semantics	PDF		
14 September	Hoare logic	PDF		HW4 out
17 September	$\lambda$ -calculus	PDF		
19 September	More $\lambda$ -calculus	PDF		
21 September	$\lambda$ -calculus encodings	PDF		HW5 out
24 September	Recursion	PDF		
26 September	Definitional translation	PDF		
28 September	Review	PDF		
1 October	Preliminary Exam I			
3 October	Continuations	PDF		
5 October	More continuations	PDF		HW6 out
8 October	No class (Fall Break)			
10 October	Types	PDF		

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

# CS 4110 (Fall 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



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	Chapter 2	HW1 out
27 August		Inductive		
29 August		Large-step semantics	PDF	
31 August		BMP	PDF	
3 September	Denotational & Axiomatic Semantics	No class (Labor Day)		
5 September		BMP summaries	PDF	
7 September				
10 September		Denotational semantics	PDF	
12 September		Axiomatic semantics	PDF	
14 September	Hoare logic	PDF		
17 September	$\lambda$ -calculus			HW4 out
19 September		More $\lambda$ -calculus	PDF	
21 September		$\lambda$ -calculus encoding	PDF	
24 September		Recursion	PDF	
26 September		Definitional translation	PDF	
28 September		Review	PDF	
1 October		Preliminary Exam I		
3 October		Continuations	PDF	
5 October	More continuations	PDF		
8 October	No class (Fall Break)			
10 October	Types	PDF		

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

# CS 4110 (Fall 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



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	Chapter 2	HW1 out
27 August		Inductive		
29 August		Large-step semantics	PDF	
31 August		BMP	PDF	
3 September	No class (Labor Day)			
5 September	Denotational & Axiomatic Semantics	BMP summaries		
7 September				
10 September		Denotational semantics	PDF	
12 September		Axiomatic semantics	PDF	
14 September		Hoare logic	PDF	HW4 out
17 September	$\lambda$ -calculus	$\lambda$ -calculus	PDF	
19 September		More $\lambda$ -calculus	PDF	
21 September		$\lambda$ -calculus encoding	PDF	HW5 out
24 September		Recursion	PDF	
26 September		Definitional translation	PDF	
28 September	Review	PDF		
1 October	Preliminary Exam I			
3 October	Continuations	PDF		
6 October	More continuations	PDF		HW6 out
8 October	No class (Fall Break)			
10 October	Types	PDF		

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

# CS 4110 (Fall 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



Cornell University  
Department of  
Computer Science

Home

Syllabus

Schedule

Resources

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

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

**CS 4110 (Fall 2014)**  
 Programming Languages and Logics  
 MWF 9:05-9:55  
 Gates G01



Cornell University  
 Department of  
 Computer Science

**Home Syllabus Schedule Resources**

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

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

**Type Systems & Program Analyses**

$\lambda$ -calculus

# CS 4110 (Fall 2014)

Programming Languages and Logics

MWF 9:05-9:55

Gates G01



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		Chapter 2	HW1 out
27 August	Inductive			
29 August	Large-step semantics	PDF		
31 August	BMP	PDF		HW2 out
3 September	No class (Labor Day)			
5 September	BMP exercises	PDF		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	PDF		
12 September	Axiomatic semantics	PDF		
14 September	Hoare logic	PDF		HW4 out
17 September	$\lambda$ -calculus	PDF		
19 September	More $\lambda$ -calculus	PDF		
21 September	$\lambda$ -calculus encoding	PDF		HW5 out
24 September	Recursion	PDF		
26 September	Definitional translation	PDF		
28 September	Review	PDF		
1 October	Preliminary Exam I			
3 October	Continuations	PDF		
6 October	More continuations	PDF		HW6 out
8 October	No class (Fall Break)			
10 October	Types	PDF		

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

**CS 4110 (Fall 2014)**  
 Programming Languages and Logics  
 MWF 9:05-9:55  
 Gates G01



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		Chapter 2	HW1 out
27 August	Inductive semantics			
29 August	Large-step semantics	PDF		
31 August	BMP	PDF		HW2 out
3 September	No class (Labor Day)			
5 September	BMP exercises	PDF		
7 September	Denotational & Axiomatic Semantics			
10 September	Denotational semantics	PDF		
12 September	Axiomatic semantics	PDF		
14 September	Hoare logic	PDF		HW4 out
17 September	$\lambda$ -calculus	PDF		
19 September	More $\lambda$ -calculus	PDF		
21 September	$\lambda$ -calculus encoding	PDF		HW5 out
24 September	Recursion	PDF		
26 September	Definitional translation	PDF		
28 September	Review	PDF		
1 October	Preliminary Exam I			
3 October	Continuations	PDF		
6 October	More continuations	PDF		HW6 out
8 October	No class (Fall Break)			
10 October	Types	PDF		

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

Type Systems & Program Analyses

Preliminary Exam II

Advanced Topics

$\lambda$ -calculus

Preliminary Exam I

Fall Break



**CS 4110 (Fall 2014)**  
 Programming Languages and Logics  
 MWF 9:05-9:55  
 Gates G01



Cornell University  
 Department of  
 Computer Science

Home Syllabus Schedule Resources

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

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



# Course Work

---

## Participation (5%)

- Lectures
- Office hours
- Online discussions

## Homework (40%)

- 10 assignments, roughly one per week
- Mix of theory and practice
- Can work with *one* partner
- No late submissions
- Two slip days and lowest score discarded

## Preliminary Exams (15% each)

- ~~October 6th~~      Not scheduled by the registrar-
- ~~November 14th~~    probably in-class

## Final Exam (25%)

- Date TBD

# Academic Integrity

---

Some simple requests:

1. You are here as members of an academic community. Conduct yourself with integrity.
2. Problem sets must be completed with your partner, and only your partner. You must *not* consult other students, alums, friends, Google, GitHub, StackExchange, Course Hero, etc.!
3. If you aren't sure what is allowed and what isn't, please ask.

# Special Needs and Wellness

---

- I will provide reasonable accommodations to students with documented disabilities (e.g., physical, learning, psychiatric, vision, hearing, or systemic).
- If you are experiencing undue personal or academic stress at any time during the semester (or if you notice that a fellow student is), contact me, Engineering Advising, or Gannett.

# Language Specification

# Language Specification

---

Formal Semantics: what do programs mean?

## Three Approaches

- Operational
  - ▶ Models program by its execution on abstract machine
  - ▶ Useful for implementing compilers and interpreters
- Axiomatic
  - ▶ Models program by the logical formulas it obeys
  - ▶ Useful for proving program correctness
- Denotational
  - ▶ Models program literally as mathematical objects
  - ▶ Useful for theoretical foundations

# Language Specification

---

Formal Semantics: what do programs mean?

## Three Approaches

- Operational
  - ▶ Models program by its execution on abstract machine
  - ▶ Useful for implementing compilers and interpreters
- Axiomatic
  - ▶ Models program by the logical formulas it obeys
  - ▶ Useful for proving program correctness
- Denotational
  - ▶ Models program literally as mathematical objects
  - ▶ Useful for theoretical foundations

Question: few languages have a formal semantics. Why?



# Formal Semantics

---

## Too Hard?

- Modeling a real-world language is hard
- Notation can get very dense
- Sometimes requires developing new mathematics
- Not yet cost-effective for everyday use

## Overly General?

- Explains the behavior of a program on *every* input
- Most programmers are content knowing the behavior of their program on *this* input (or these inputs)

Okay, so who needs semantics?

# A Tricky Example

---

Question #1: is the following Java program legal?

Question #2: if yes, what does it do?

```
class A { static int a = B.b + 1; }  
class B { static int b = A.a + 1; }
```

# Who Needs Semantics?

---

## Unambiguous Description

- Anyone who wants to design a new feature
- Basis for most formal arguments
- Standard tool in PL research

## Exhaustive Reasoning

- Sometimes have to know behavior on all inputs
- Compilers and interpreters
- Static analysis tools
- Program transformation tools
- Critical software

# Language Design

# Design Desiderata

---

Question: What makes a good programming language?

# Design Desiderata

---

Question: What makes a good programming language?

One answer: "a good language is one people use"

# Design Desiderata

---

Question: What makes a good programming language?

One answer: "a good language is one people use"

Wrong! Are COBOL and JavaScript the best languages?

# Design Desiderata

---

**Question:** What makes a good programming language?

**One answer:** “a good language is one people use”

**Wrong!** Are COBOL and JavaScript the best languages?

**Some good features:**

- Simplicity (clean, orthogonal constructs)
- Readability (elegant syntax)
- Safety (guarantees that programs won't “go wrong”)
- Support for programming in the large (modularity)
- Efficiency (good execution model and tools)



# Design Challenges

---

Unfortunately these goals almost always conflict.

- Types provide strong guarantees but restrict expressiveness.
- Safety checks eliminate errors but have a cost—either at compile time or run time.
- Some verification tools are so complicated, you essentially need a PhD to use them!

# Design Challenges

---

Unfortunately these goals almost always conflict.

- Types provide strong guarantees but restrict expressiveness.
- Safety checks eliminate errors but have a cost—either at compile time or run time.
- Some verification tools are so complicated, you essentially need a PhD to use them!

A lot of research in programming languages is about discovering ways to gain without (too much) pain.

# Story: Unexpected Interactions

---

A real story illustrating the perils of language design

Cast of characters includes famous computer scientists

## Timeline:

- 1982: ML is a functional language with type inference, polymorphism (generics), and monomorphic references (pointers)
- 1985: Standard ML innovates by adding polymorphic references → unsoundness
- 1995: The “innovation” fixed

# ML Type System

**Polymorphism:** allows code to be used at different types

**Examples:**

- $\text{List.length} : \forall \alpha. \alpha \text{ list} \rightarrow \text{int}$
- $\text{List.hd} : \forall \alpha. \alpha \text{ list} \rightarrow \alpha$

**Type Inference:**  $e \rightsquigarrow \tau$

- e.g., let  $\text{id}(x) = x \rightsquigarrow \forall \alpha. \alpha \rightarrow \alpha$
- Generalize types not constrained by the program
- Instantiate types at use  $\text{id}(\text{true}) \rightsquigarrow \text{bool}$

# ML References

---

By default, values in ML are immutable.

But we can easily extend the language with imperative features.

Add **reference types** of the form  $\tau \text{ ref}$

Add **expressions** of the form

$\text{ref } e : \tau \text{ ref}$	where $e : \tau$	(allocate)
$!e : \tau$	where $e : \tau \text{ ref}$	(dereference)
$e_1 := e_2 : \text{unit}$	where $e_1 : \tau \text{ ref}$ and $e_2 : \tau$	(assign)

Works as you'd expect (like pointers in C).

# Polymorphism + References

Consider the following program

Code	Type Analysis
let id = (fun x -> x)	

# Polymorphism + References

Consider the following program

Code

```
let id = (fun x -> x)
```

```
let p = ref id
```

Type Analysis

# Polymorphism + References

Consider the following program

Code

```
let id = (fun x -> x)
```

```
let p = ref id
```

```
let inc = (fun n -> n+1)
```

Type Analysis



# Polymorphism + References

Consider the following program

Code

```
let id = (fun x -> x)
let p = ref id
let inc = (fun n -> n+1)
p := inc;
(!p) true
```

Type Analysis

# Polymorphism + References

Consider the following program

Code

```
let id = (fun x -> x)
```

```
let p = ref id
```

```
let inc = (fun n -> n+1)
```

```
p := inc;
```

```
(!p) true
```

Type Analysis

```
id :  $\alpha \rightarrow \alpha$ 
```

# Polymorphism + References

Consider the following program

Code

```
let id = (fun x -> x)
```

```
let p = ref id
```

```
let inc = (fun n -> n+1)
```

```
p := inc;
```

```
(!p) true
```

Type Analysis

```
id :  $\alpha \rightarrow \alpha$ 
```

```
p :  $(\alpha \rightarrow \alpha)$  ref
```

# Polymorphism + References

Consider the following program

Code	Type Analysis
let id = (fun x -> x)	id : $\alpha \rightarrow \alpha$
let p = ref id	p : $(\alpha \rightarrow \alpha)$ ref
let inc = (fun n -> n+1)	inc : int $\rightarrow$ int
p := inc;	
(!p) true	

# Polymorphism + References

Consider the following program

Code	Type Analysis
let id = (fun x -> x)	$id : \alpha \rightarrow \alpha$
let p = ref id	$p : (\alpha \rightarrow \alpha) \text{ ref}$
let inc = (fun n -> n+1)	$inc : \text{int} \rightarrow \text{int}$
p := inc;	OK since $p : (\text{int} \rightarrow \text{int}) \text{ ref}$
(!p) true	OK since $p : (\text{bool} \rightarrow \text{bool}) \text{ ref}$

# Polymorphism + References

---

## Problem

- Type system is not sound
- Well-typed program  $\rightarrow^*$  type error!

# Polymorphism + References

---

## Problem

- Type system is not sound
- Well-typed program  $\rightarrow^*$  type error!

## Proposed Solutions

1. “Weak” type variables
  - ▶ Can only be instantiated in restricted ways
  - ▶ But type exposes functional vs. imperative
  - ▶ Difficult to use

# Polymorphism + References

## Problem

- Type system is not sound
- Well-typed program  $\rightarrow^*$  type error!

## Proposed Solutions

### 1. "Weak" type variables

- ▶ Can only be instantiated in restricted ways
- ▶ But type exposes functional vs. imperative
- ▶ Difficult to use

### 2. Value restriction

- ▶ Only generalize types of values
- ▶ Most ML programs already obey it
- ▶ Simple proof of type soundness



# Lessons Learned

---

- Features often interact in unexpected ways
- The design space is huge
- Good designs are sparse and don't happen by accident
- Simplicity is rare:  $n$  features  $\rightarrow n^2$  interactions
- Most PL researchers work with small languages (e.g.,  $\lambda$ -calculus) to study core issues in isolation
- But must pay attention to whole languages too