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)
- C# – Microsoft (2001)
- F# – Syme (2005)
Programming Languages

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

PL intersects with many other areas

Current trends

- Domain-specific languages
- Static analysis and types
- Language-based security
- Formal verification
- Concurrency

Both theoretically and practically “meaty”
Syllabus
Course Goals

- Learn techniques for modeling programs*
  - Formal semantics (operational, axiomatic, denotational)
  - Modeling to advanced language features
  - Develop reasoning principles (induction, co-induction)
- Explore applications of these techniques
  - Optimization
  - Static analysis
  - Verification
- PhD students: cover material for PL qualifying exam
- Have fun :-)
Prerequisites

Programming Experience
- e.g., C, Java, Prolog, OCaml, Haskell, Scheme/Racket
- Comfortable with a functional language
- For undergrads: CS 3110 or 4110 or equivalent

Mathematical Maturity
- e.g., set theory, rigorous proofs, induction
- Much of this class will involve formal reasoning
- Hardest topic: denotational semantics

Interest (having fun is a goal! :-)

If you don’t meet these prerequisites, get in touch.
Course Work

Participation (5%)  
- Lectures, recitations, and office hours  
- Email list discussions

Homework (30%)  
- 6 assignments, roughly every other week  
- Mostly theoretical, some programming  
- *Strongly* encouraged to work with a partner  
- Two “slip” days: automatic 48-hour extension

Preliminary Exam (25%)  
- March 22nd + take-home problems.

Final Exam (40%)  
- Date and time TBA  
- Cumulative, with focus on the material from 2\(^{nd}\) half
Two simple requests:

1. Most of you are here training to become members of the research community. Conduct yourself with integrity.

2. If you aren’t sure what is allowed and what isn’t, please ask!
I will provide reasonable accommodations to students who have a documented disability (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.
Course Staff

Instructor
Nate Foster
Office: Gates 432
Hours: Mondays 4am-5pm

Teaching Assistant
Eric Perdew
Hours: TBA

Web Page
http://www.cs.cornell.edu/Courses/cs6110/2016sp

(office hours start next week)