CMPS 203
Winter 2014

Nathan Whitehead
nwhitehe@ucsc.edu
Baskin Engineering 259A
Office hours: Tues 10-11 am, Thur 4-5 pm
Introductions

- **Instructor:**
  - *Nathan Whitehead*
  - nwhitehe@ucsc.edu
  - Baskin Engineering 259A
  - Office hours dedicated to CMPS 203: Tues 4-5 pm
  - General office hours: Tues 10-11 am, Thur 4-5 pm

- **Teaching Assistant:**
  - *Jeremy Gottlieb*
  - gottlieb@soe.ucsc.edu
  - Office hours: E2 316 Wed 1:30-2:30, also by email
Do you have what it takes to succeed?

- Programming language background
  - some programming experience
  - exposure to variety languages (e.g. C, Python, Java, ML, Lisp, Prolog)
  - CS 112 or equiv. comparative programming languages

- 'Mathematical sophistication'
  - not scared of math notation
  - can follow and do proofs
  - some familiarity with logic
Prerequisites

Talk to me if:
- you're not sure you meet the prerequisites
- you are a graduate student outside CS or CE
- you did not enroll but want to attend
Course Goals

- Learn more about functional lazy languages (Haskell)
- Learn about techniques for understanding and describing programming languages
  - operational semantics
  - relation to implementation
- Learn about formal verification (Coq)
  - general theorem proving
  - proving program correctness
- Learn how to construct verified interpreters
  - automatic program extraction
  - 'metaproperties'
Grand Plan

Lazy functional programming (Haskell) → Operational semantics → Interpreters (Haskell)

Theorem proving (Coq) → Program verification (Coq) → Verified interpreters (Coq + Haskell)

(we'll see how far we get on verified interpreters)
Reading

- Helpful books (optional)
  - *Types and Programming Languages*, Benjamin Pierce
  - *The Haskell School of Expression*, Paul Hudak
  - *The Formal Semantics of Programming Languages*, Glynn Winskel
  - *Interactive Theorem Proving and Program Development*, Yves Bertot and Pierre Castéran
  - *Certified Programming with Dependent Types*, Adam Chlipala
Course work

- In-class participation
- Homework
  - 4 assignments
  - announced and explained in class
  - typically due 1-2 weeks after assigned
  - use eCommons to submit
- Small final project
  - concentrated in second half
  - 20-40+ hours of work per person
- No exams
Grades are calculated:
  - Homework 50%
  - Project (report and presentation) 40%
  - Class participation 10%

Need to do well in all 3 areas
The Project

- More in-depth exploration of a topic of your own interest
- Example types:
  - Programming project (implement something)
  - Survey of published research (summarize papers with examples)
  - Short original research paper (investigate an idea)
- 2-5 page report, 5 slide/5 minute lightning talk
- Grad quality
  - citations
  - clearly expressed idea
  - put in context of field
Who are you?

domain

favorite PLs

year

FP experience

field

logic experience
Who are we?
Functional Programming?

No assignment.
No mutation.
No loops.

Dude wat
CODE WRITTEN IN HASKELL IS GUARANTEED TO HAVE NO SIDE EFFECTS.

...BECAUSE NO ONE WILL EVER RUN IT?
Who uses FP?

F#
Who uses FP?

MapReduce
Who uses FP?

Facebook

Erlang
Who uses FP?

Jane Street

OCaml