Last lecture!
Final exam info posted
Tips

- review definitions & concepts
- use code examples
- redo labs/exercises/assignments
- write elegant code
Big takeaways
So you just... learned about programming languages?
Big learning goals

1. Define, analyze, and modify syntactic features of a programming language.

2. Define and analyze semantic features of a programming language.

3. Write programs that operate on other programs.
A programming language is the (main) way a programmer communicates with a computer.
Sapir-Whorf hypothesis
the Blub paradox
As long as our hypothetical Blub programmer is looking down the power continuum, he knows he's looking down. Languages less powerful than Blub are obviously less powerful, because they're missing some feature he's used to. But when our hypothetical Blub programmer looks in the other direction, up the power continuum, he doesn't realize he's looking up. What he sees are merely weird languages. He probably considers them about equivalent in power to Blub, but with all this other hairy stuff thrown in as well. Blub is good enough for him, because he thinks in Blub.
functions as the basic unit of computation and abstraction
(+ (* 22 1.8) 32)
(+ (* -7 1.8) 32)
(define (add1ToAll lst)
  (if (empty? lst)
      '()
      (cons ((add1 (first lst))
              (add1ToAll (rest lst))))))

(define (sqrAll lst)
  (if (empty? lst)
      '()
      (cons ((sqr (first lst))
              (sqrAll (rest lst))))))
acc = seed
for x in lst:
    acc = f(x, acc)
(sort list less-than?)
sorted(list, key)
Collections.sort(list, Comparator)
environment
lexical scope
closures
extending language syntax
class

message passing

___dict___

macros to *bind names*
continuations
“non-deterministic” choice macros to alter control flow
domain-specific languages

make
HTML
SQL
GraphViz
type systems
expressing constraints
static type checking
static type inference
polymorphism: generic and ad-hoc
higher-order typeclasses

Functor
Monad
monads as abstraction of sequenced computations
Computations that can fail

Maybe a
Either String a
Computations that can use “mutable” state

State s a
Computations that interact with the world

IO a
In most languages, any code can...

return null
raise an exception
mutate anything
print/makeHTTP/blowUpUofT
expressing constraints!!
What comes next?
Implementation of functional/declarative language features
“Zero-cost abstractions”
Pushing the boundaries of static analysis
“Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.”

www.rust-lang.org
dependent types

[1, 2, 3] :: Vector Int 3
formal verification
Thank you!