LISP is over half a century old and it still has this perfect, timeless air about it.

I wonder if the cycles will continue forever.

A few coders from each new generation re-discovering the LISP arts.

These are your father's parentheses.

Elegant weapons for a more... civilized age.

COS 326 Functional programming: an elegant weapon for the modern age
In 1936, Alonzo Church invented the lambda calculus. He called it a logic, but it was a language of pure functions -- the world's first programming language.

He said:

"There may, indeed, be other applications of the system than its use as a logic."
Robert Harper (CMU): *The lambda directly and immediately relevant to this day, rather than something that collects dust on the shelf. No one cares one bit about the details of a Turing Machine; for it fails to address the central issue of modularity.*
Vastly Abbreviated FP Designer History

Alonzo Church: lambda calculus 1930’s

Guy Steele & Gerry Sussman: Scheme late 1970’s

John McCarthy: LISP 1958

Robin Milner, Mads Tofte, & Robert Harper Standard ML 1980’s

Xavier Leroy: Ocaml 1990’s

Don Syme: F# 2000’s
Where do I fit in?

Alonzo Church
Princeton Prof 1929–1967

Steven Kleene
Princeton PhD 1934
IAS 1939-1940

Robert Constable
Developed Nuprl Thereom Prover

Bob Harper
Developed Standard ML

Greg Morrissett
Developed Typed Assembly Language

David Walker
Princeton Prof 2002–
A bit of fun:

• [http://www.malevole.com/mv/misc/killerquiz/](http://www.malevole.com/mv/misc/killerquiz/)

Have a go at these I recently made for E4: [Janey Thomson's Marathon](http://www.malevole.com/mv/misc/killerquiz/) · [Captcha Invaders](http://www.malevole.com/mv/misc/killerquiz/) · [The Rather Difficult Game](http://www.malevole.com/mv/misc/killerquiz/)
Vastly Abbreviated FP Geneology

LCF Theorem Prover (70s) → Edinburgh ML

- Miranda (80s)
- Standard ML (90s - now)
- Haskell (90s - now)

- LISP (50s - now)
- Scheme (70s - now)
- Racket (00s - now)

- Caml (80s - now)
- OCaml (90s - now)
- F# (now)
- Coq (80s - now)

Lazy: Haskell

Typed, polymorphic:

- Edinburgh ML
- Standard ML
- OCaml

Call-by-value:

- Miranda
- Haskell
- Caml
- OCaml
- F#

Dependently typed:

- Racket
- Coq
But Why Functional Programming Now?

- Functional programming will introduce you to new ways to *think about* and *structure* your programs:
  - new reasoning principles
  - new abstractions
  - new design patterns
  - new algorithms
  - elegant code

- Technology trends point to increasing parallelism:
  - multicore, gpu, data center
  - functional programming techniques such as map-reduce provide a plausible way forward for many applications
Functional Languages: Who’s using them?

- map-reduce in their data centers
- Scala for correctness, maintainability, flexibility
- Erlang for concurrency
- Haskell for managing PHP
- Haskell to synthesize hardware
- O’Caml for reliability
- Twitter
- Facebook
- Google
- Microsoft
- Jane Street
- Barclays
- Bluespec
- Mathemeticians

- www.artima.com/scala/article/twitter_on_scala.html
- http://www.haskell.org/haskellwiki/Haskell_in_industry
- Coq proof of 4-color theorem
Functional Languages: Join the crowd

- Elements of functional programming are showing up all over
  - F# in Microsoft Visual Studio
  - Scala combines ML (a functional language) with Objects
    - runs on the JVM
  - C# includes “delegates”
    - delegates == functions
  - Python includes “lambdas”
    - lambdas == more functions
  - Javascript
    - find tutorials online about using functional programming techniques to write more elegant code
  - C++ libraries for map-reduce
    - enabled functional parallelism at Google
  - Java has generics and GC
  - ...

COURSE LOGISTICS
Course Staff

Prof David Walker
office: COS 211
email: dpw@cs
office hours: MW 12:20-1

TA Naga Katta
office: COS 311
email: nkatta@cs
office hours: M 2:30-3, T 3-3:30
Resources

• Web: http://www.cs.princeton.edu/~dpw/courses/cos326-12/

• Lecture schedule and readings:
  – $(coursehome)/lectures.php

• Assignments:
  – $(coursehome)/assignments.php
  – first assignment due next week: tuesday sept 25!

• Precepts (Friend 007, 50min, Th 7:30, Fr 11)
  – first half of semester (intermittent in 2nd half)

• Install OCaml: $(coursehome)/resources.php