Haskell

Edward Z. Yang
Haskell

- First-class functions
- Type inference
- Monads
- Pattern matching
- Type classes
- Continuations

Say more with less!

Reliability and Reuse!

Objects & Inheritance

Modules

Generics

Cross-cutting concerns

Memory management

Concurrency
What is Haskell?

a typed, lazy, pure, functional programming language
What is Haskell?

functions first class
expressions over instructions

a typed, lazy, pure, functional
programming language
What is Haskell?

No mutation/side effects

Functions do the same thing

A typed, lazy, pure, functional programming language

Equational reasoning

Maintainability

Parallelism
What is Haskell?

Expressions not evaluated until needed

A typed, lazy, pure, functional programming language

Custom control structures

Infinite data structures

Compositionality
What is Haskell?

- a typed, lazy, pure, functional programming language
  - Types checked at compile time

more expressive than what you might be used to!
Where is Haskell?

Lisp

Haskell

Algol 60

Algol 64

Pascal

Modula

C

C++

Java

Smalltalk

~ many more ~
Why Haskell?

Types will change the way you think
Raise the level of abstraction
Purity to be essential for multicore
Most Programming Languages

The Quick Death

[Simon PJ]
Successful Research Languages

Practitioners

Geeks

The Slow Death

1 yr  5 yr  10 yr  15 yr
C++, Java, Perl, Ruby

Threshold of Immortality

The Complete Absence of Death
Never was alive...
Haskell

The Second Life?
Announcements

Homework 1 is out (see Course Website)
Not due until next Wednesday
A lot of text and ideas, very little code.