

# Haskell

Edward Z. Yang

# Haskell

```
graph TD; Haskell --> FirstClass[First-class functions]; Haskell --> Pattern[Pattern matching]; Haskell --> TypeInference[Type inference]; Haskell --> TypeClasses[Type classes]; Haskell --> Monads; Haskell --> Continuations; Haskell --> Reliability[Reliability and Reuse!]; Haskell --> CrossCutting[Cross-cutting concerns]; Haskell --> Memory[Memory management]; Haskell --> Concurrency;
```

Say more with less!

First-class functions

Type inference

Monads

Pattern matching

Type classes

Continuations

Reliability and Reuse!

Objects & Inheritance

Modules

Generics

Cross-cutting concerns

Memory management

Concurrency

# What is Haskell?

a typed, lazy, functional  
programming language

# What is Haskell?

compositional programs  
purity

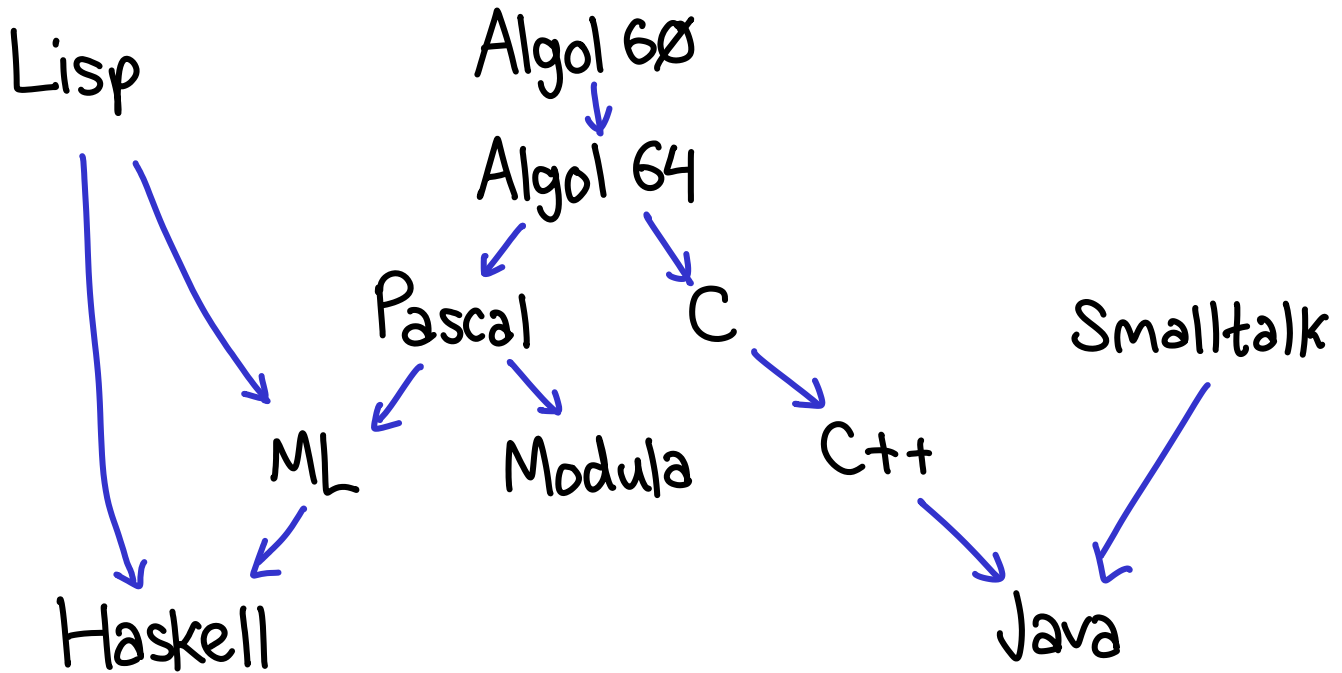
functions first class  
expression is king

a typed, lazy, functional

programming language

strong types  
type-directed programming

# Where is Haskell?



~ many more ~

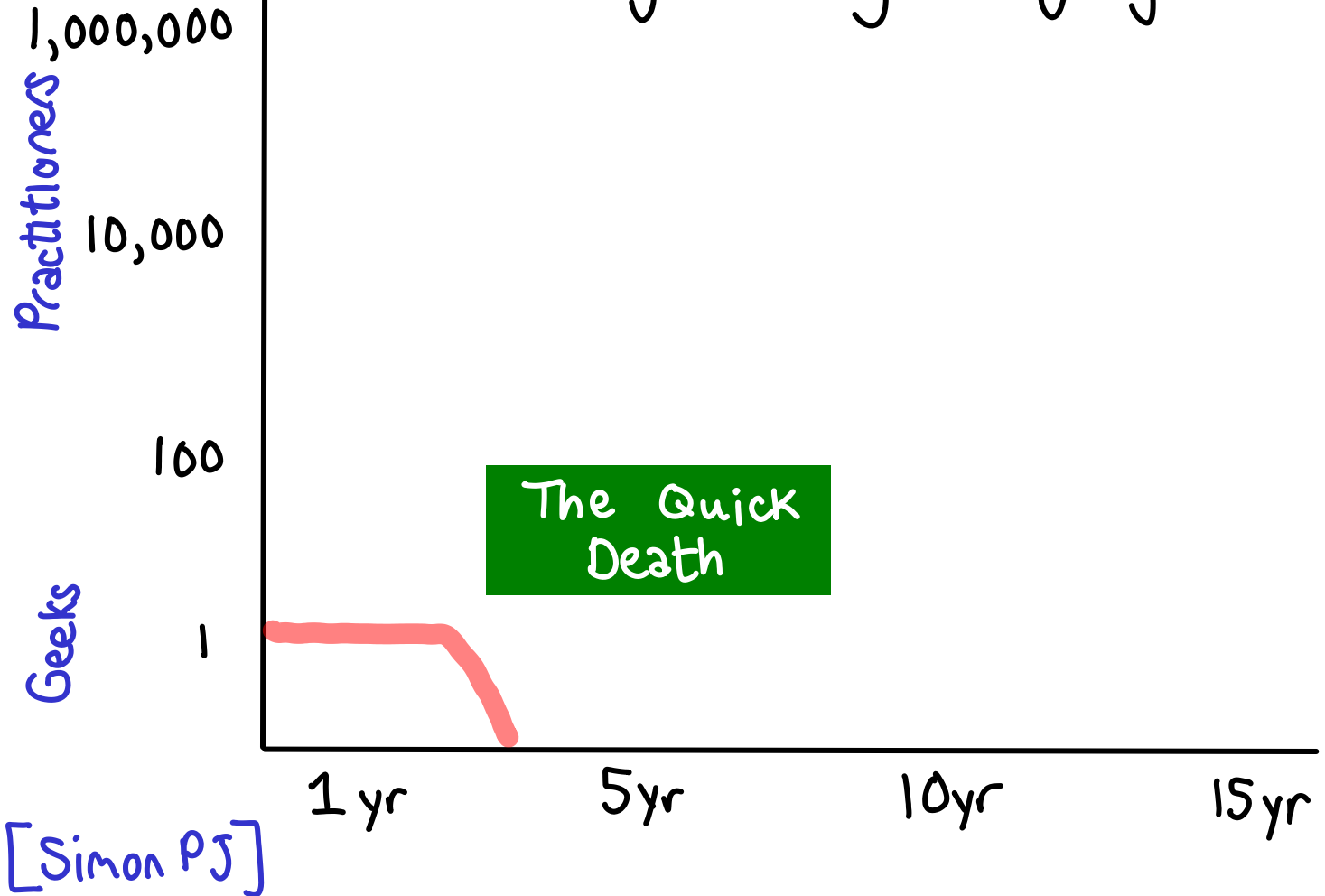
# Why Haskell?

Types will change the way  
you think

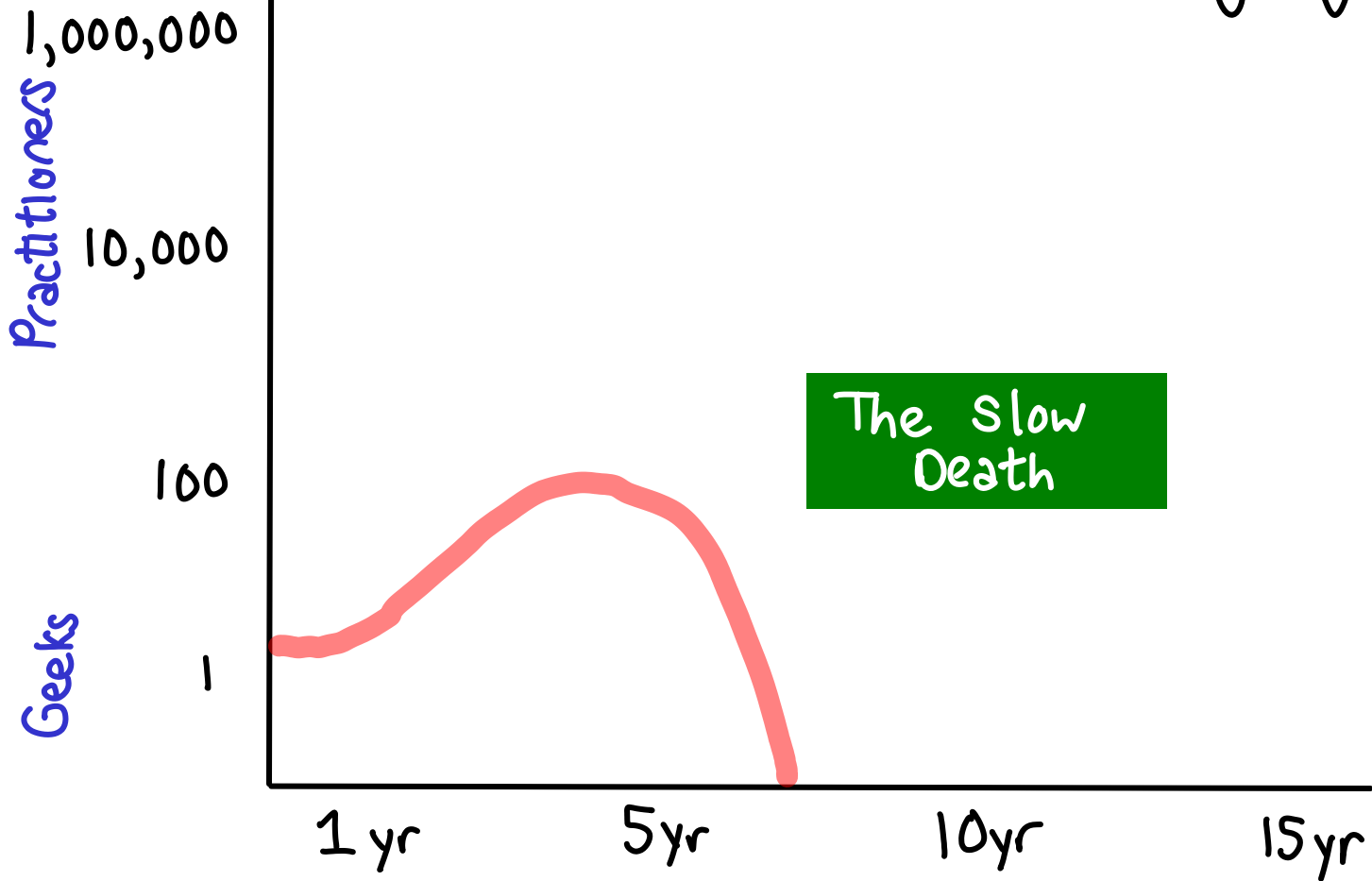
Raise the level of abstraction

Purity to be essential for multicore

# Most Programming Languages

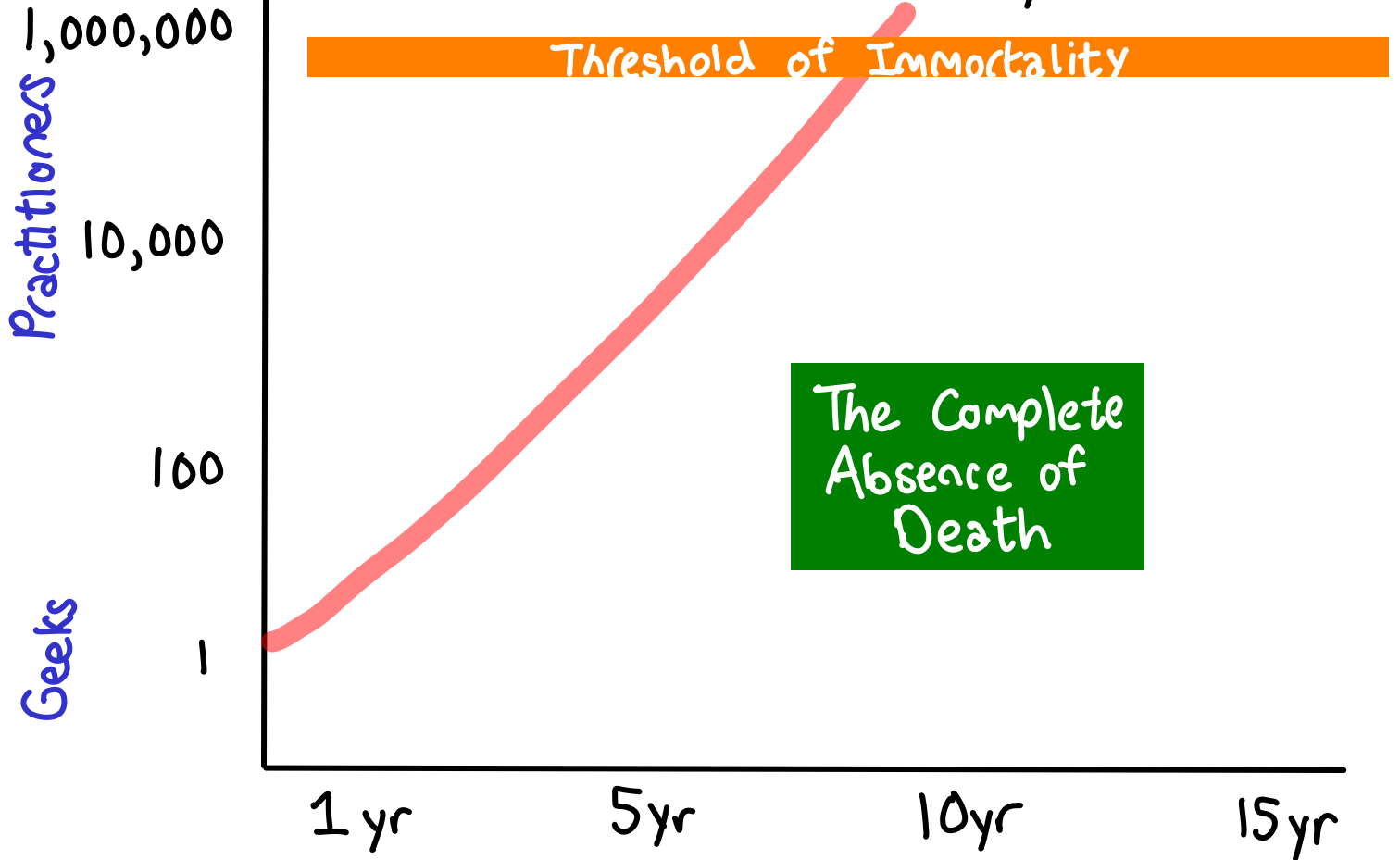


# Successful Research Languages

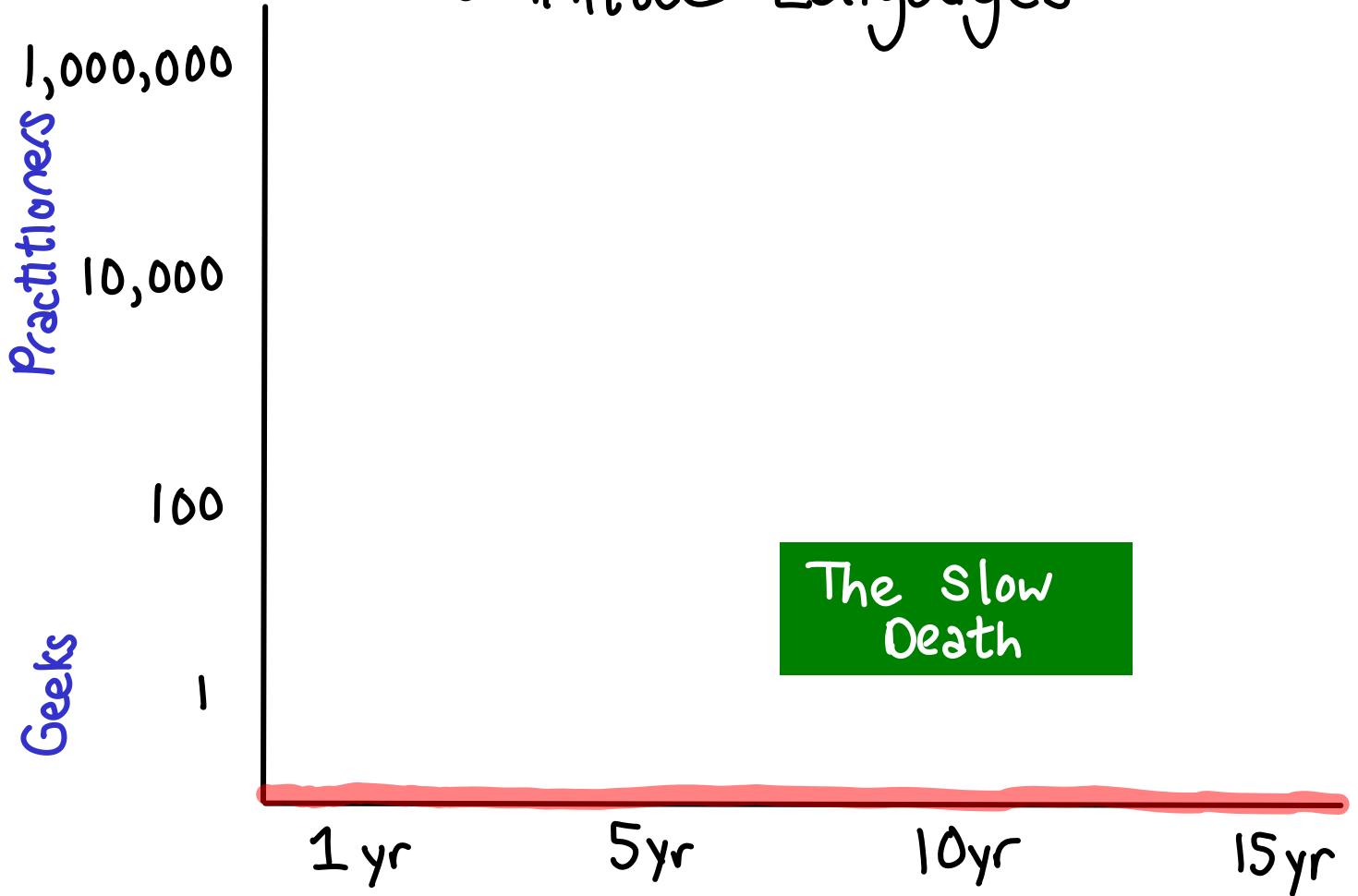


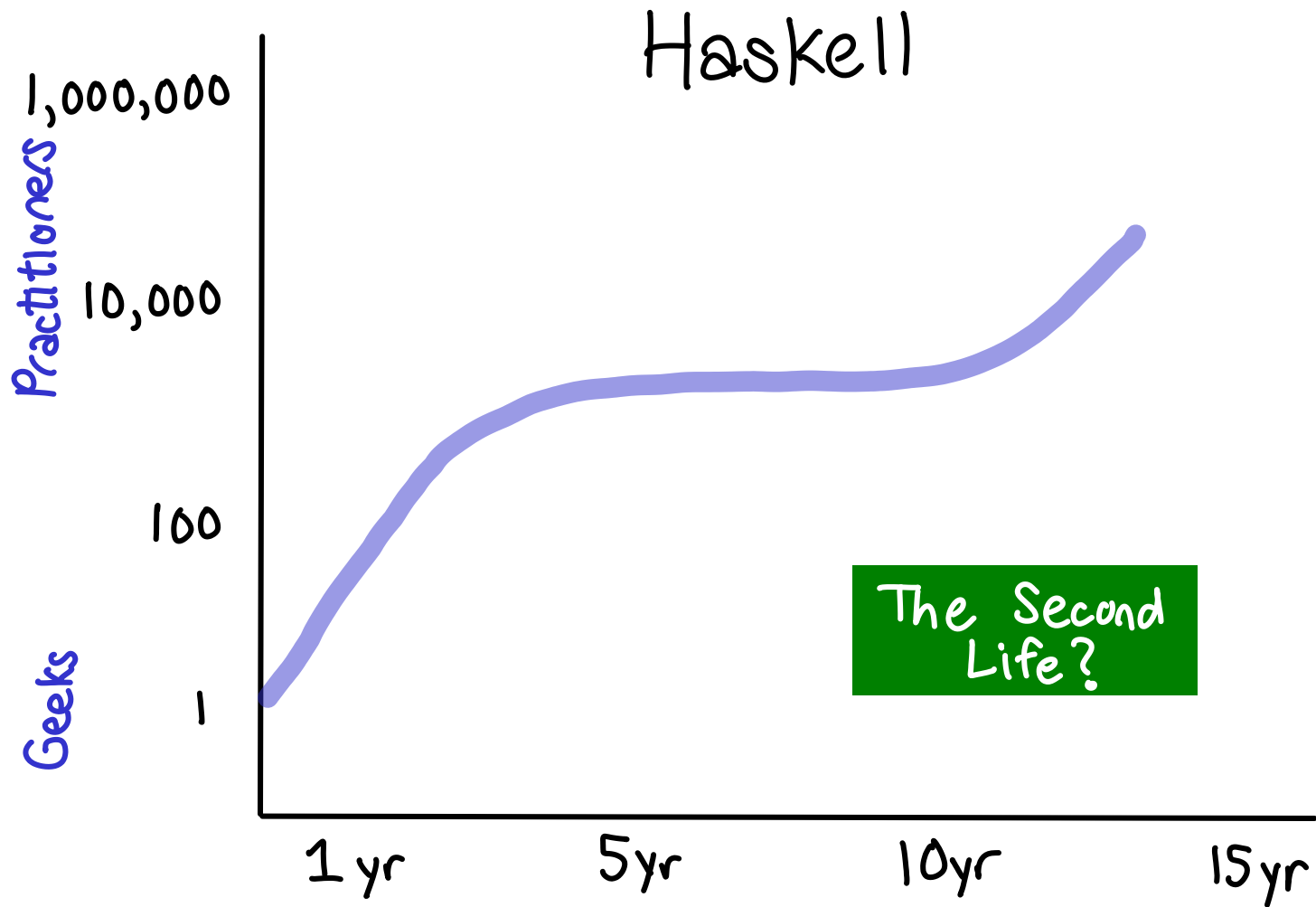


# C++, Java, Perl, Ruby



# Committee Languages





# Announcements

don't forget to  
register!



Homework 1 is out on [cs242.stanford.edu](https://cs242.stanford.edu)

Two labs: one **JavaScript**, one **Haskell**.

**A lot** of text and ideas, **very little** code.

We moved to Piazza!

[piazza.com/stanford/fall2015/cs242/home](https://piazza.com/stanford/fall2015/cs242/home)

