COMP 333 Lecture 1

Kyle Dewey

About Me

- I research automated testing techniques and their intersection with CS education
- My dissertation used logic programming extensively
- I frequently use functional programming
- This is my first time teaching this class

About this Class

- See something wrong? Want something improved? Email me about it! (kyle.dewey@csun.edu)
- I generally operate based on feedback

Bad Feedback

- This guy sucks.
- This class is boring.
- This material is useless.

Good Feedback

- This guy sucks, *I can't read his writing.*
- This class is boring, *it's way too slow.*
- This material is useless, *I don't see how it relates to anything in reality.*

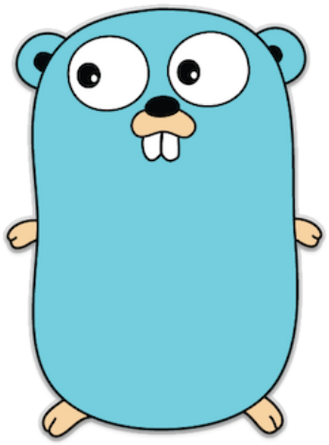
- I can't fix anything if I don't know what's wrong

Why this Course?

- Navigating programming languages
- Understanding how programming languages work
- Shaping how you think about programming

Navigating Languages

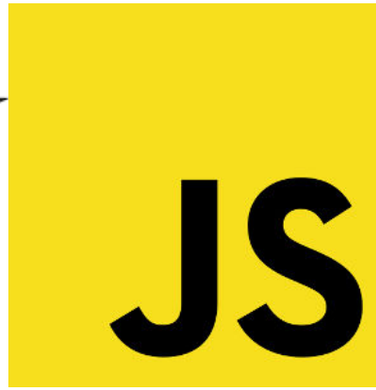
Animals



Birds

Camels

Pointy



Java

Coffee

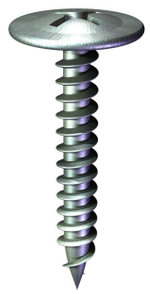
Lambda

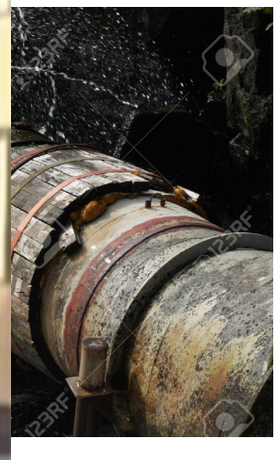
Whatever this is

How Languages Work

- Proper debugging demands knowledge of underlying language
- Knowledge prevents gotchas (and gotchas usually end with greater knowledge)
- While languages abound, language features are sparse

Thinking About Programming





The Point

- Languages influence how you think and approach problems
- The same problem can be MUCH simpler to solve in a different language

The Point

- Languages influence how you think and approach problems
  - The same problem can be MUCH simpler to solve in a different language

---

Scala

```
for {
  a <- Seq(1, 2, 3)
  b <- Seq("foo", "bar")
} yield (a, b)
```

The Point

- Languages influence how you think and approach problems
- The same problem can be MUCH simpler to solve in a different language

Scala

```
for {
  a <- Seq(1, 2, 3)
  b <- Seq("foo", "bar")
} yield (a, b)
```

Java

- Bulk of Summer
- Bulk of semester

Common Misconceptions: Performance

Always Write the Fastest Code

- "Premature optimization is the root of all evil" - Donald Knuth
- Programmer median salary: $82,000/year
- AWS m4.large (reserved): $545/year

High-Level Languages are Slow

- Java can outperform C
- Choice of algorithm usually WAY more important
    - I have written Prolog that dramatically outperformed Java (thousands - millions of times faster)

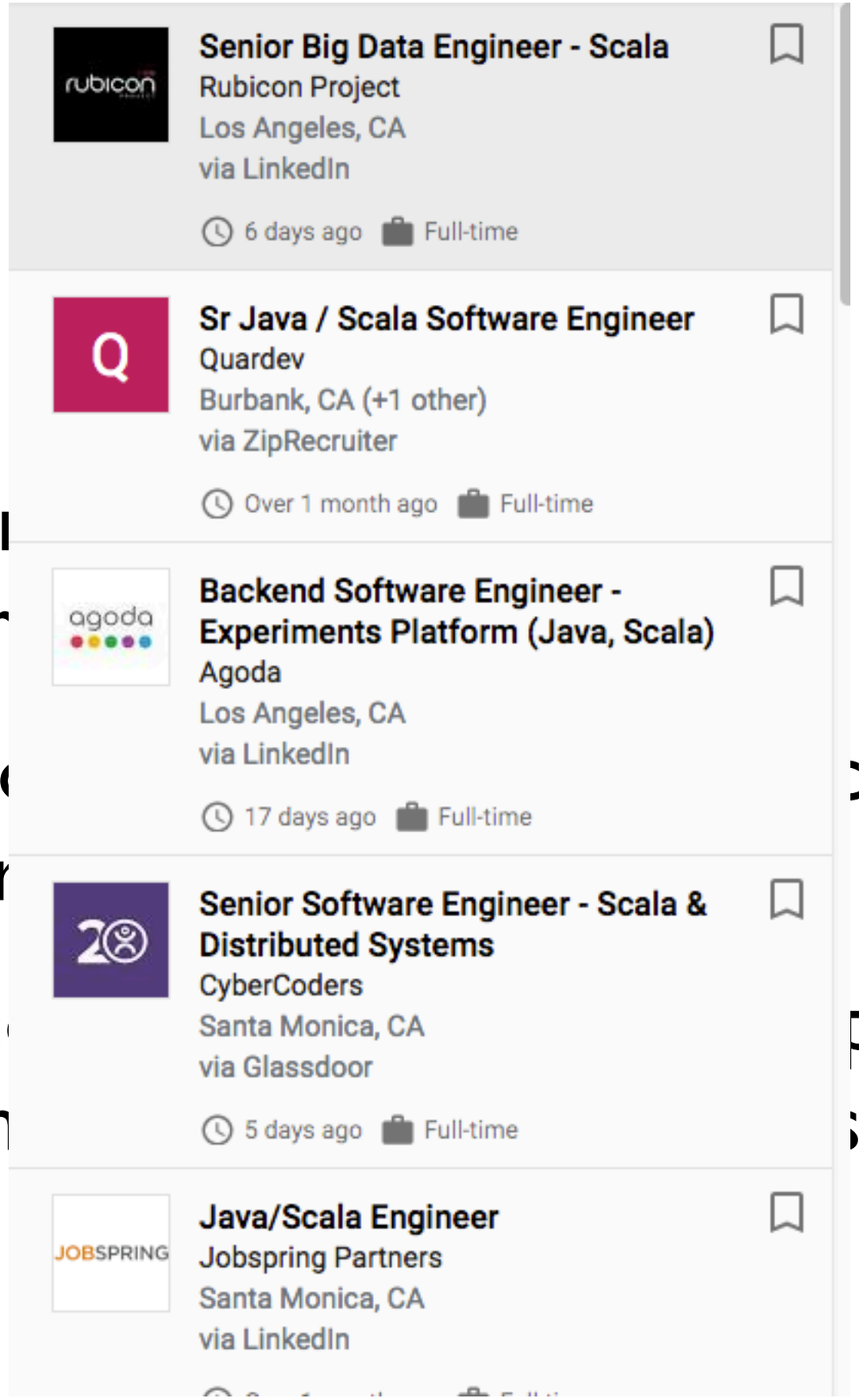
Common Misconceptions: Utility

FP is Purely Academic

- Functional programming makes concurrency much simpler
- Good software engineering practices tend to enforce functional styles
- Most modern languages now support functional programming features

FP is emic

- Functio
- concurr
- Good s ctices tend to enfor
- Most m pport function s



LP is Useless

- Logic programming is highly specialized, but not useless
- Recall: Prolog 9 million times faster than Java
- I've used it to find bugs in multiple compilers

LP is Useless

UAB - News - A 'high-speed Dr.

University of Alabama at Bir... (US) https://www.uab.edu/news/research/item/10382-a-high-sp

UAB THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

UAB News

Stay informed with the latest updates

Search

Public Relations: (205) 934-3884 UAB Quicklink

HOME CATEGORIES TRENDING TOPICS UAB IN-DEPTH JOURNALISTS FACULTY & STAFF STUDENTS BLAZER ATHLETICS

RESEARCH

A 'high-speed Dr. House' for medical breakthroughs

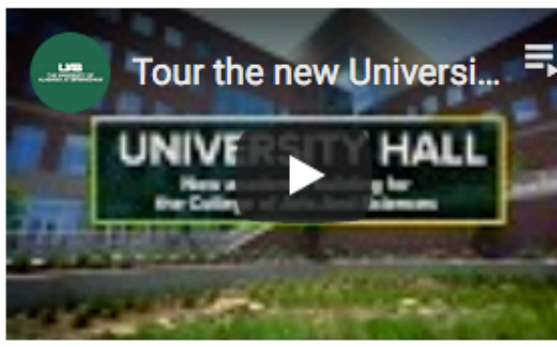
May 08, 2018 | Print | Email

Written by: Matt Windsor
Media contact: Bob Shepard

Human biology is full of surprises — especially for drug makers. Viagra wasn't designed for erectile dysfunction. Rogaine didn't start out as a hair-loss cream. Both drugs were meant to treat cardiovascular issues (as sildenafil and minoxidil, respectively), until patients reported their sexual and follicular side effects.

When his son was diagnosed with an ultra-rare disease, computer scientist Matt Might, Ph.D., kicked off a search for answers. His quest led to partnerships with researchers across the country, a White House

Featured Video



See more from UAB Visual Con

Tweets by @UABNews

UAB
@UABNews

Incorporating climate change in the classroom provides hope for the future of our planet. @JiminAntarctica @UAB_CAS @UABBiology @UABEducation go.uab.edu/r293k

Common Misconceptions: Stagnation

Industry Moves Slowly

- COBOL was once a vital language
- Perl was once the champion of the Internet
- Java has lost tons of ground to Python
- Companies that cannot adapt, die

Staying in a Comfort Zone

- "I know Python *and* Ruby, so I already am pretty flexible"

Staying in a Comfort Zone

- "I know Python *and* Ruby, so I already am pretty flexible"

Staying in a Comfort Zone

- "I know Python *and* Ruby, so I already am pretty flexible"

What this Course Is

- Heavy on programming
- Exposure to object-oriented, functional, logical, and concurrent programming
- Exposure to various language features in the context of the languages you'll use

What this Course **Isn't**

- Advanced topics in any one style
- In-depth look at language implementations
- Heavy on theory

Languages We Will Use

- JavaScript (object-oriented programming)
- Swift (functional programming)
- Prolog (logic programming)
- Rust (concurrency)

Why JavaScript?

- Most popular language on StackOverflow
- Object-oriented, but not (traditionally) class-based
- Dynamically typed, garbage collected, (typically bytecode) interpreted, typically just-in-time compilers available

Why Swift?

- 15th most popular on StackOverflow, and 6th most loved
- Not e*xactly* a functional language, but it has key functional features without getting too weird
- Statically typed, unbounded and bounded generics, compiled, algebraic data types, pattern matching, typeclasses, optional call-by-name, reference counting

Why Prolog?

- Arguably the simplest logic programming language out there
- For better or worse, logic programming is largely synonymous with Prolog's features
- Unification, nondeterminism, both (bytecode) interpreted and compiled

Why Rust?

- Most loved language on StackOverflow
- Enables memory-safe, concurrent, low-level programming (eat it, C/C++)
- Ownership types with borrowing for static memory management, algebraic data types, typeclasses, compiled, very expressive generics

Syllabus