

# COMP 333 Lecture I

Kyle Dewey

# About Me

- My research:
  - Automated program testing + CS education
  - Programming language design (with JPL)
- Lots of experience with functional and logic programming
- Taught this class a bunch

# About this Class

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

# Bad Feedback

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

–I can't do anything in response to this



# Good Feedback

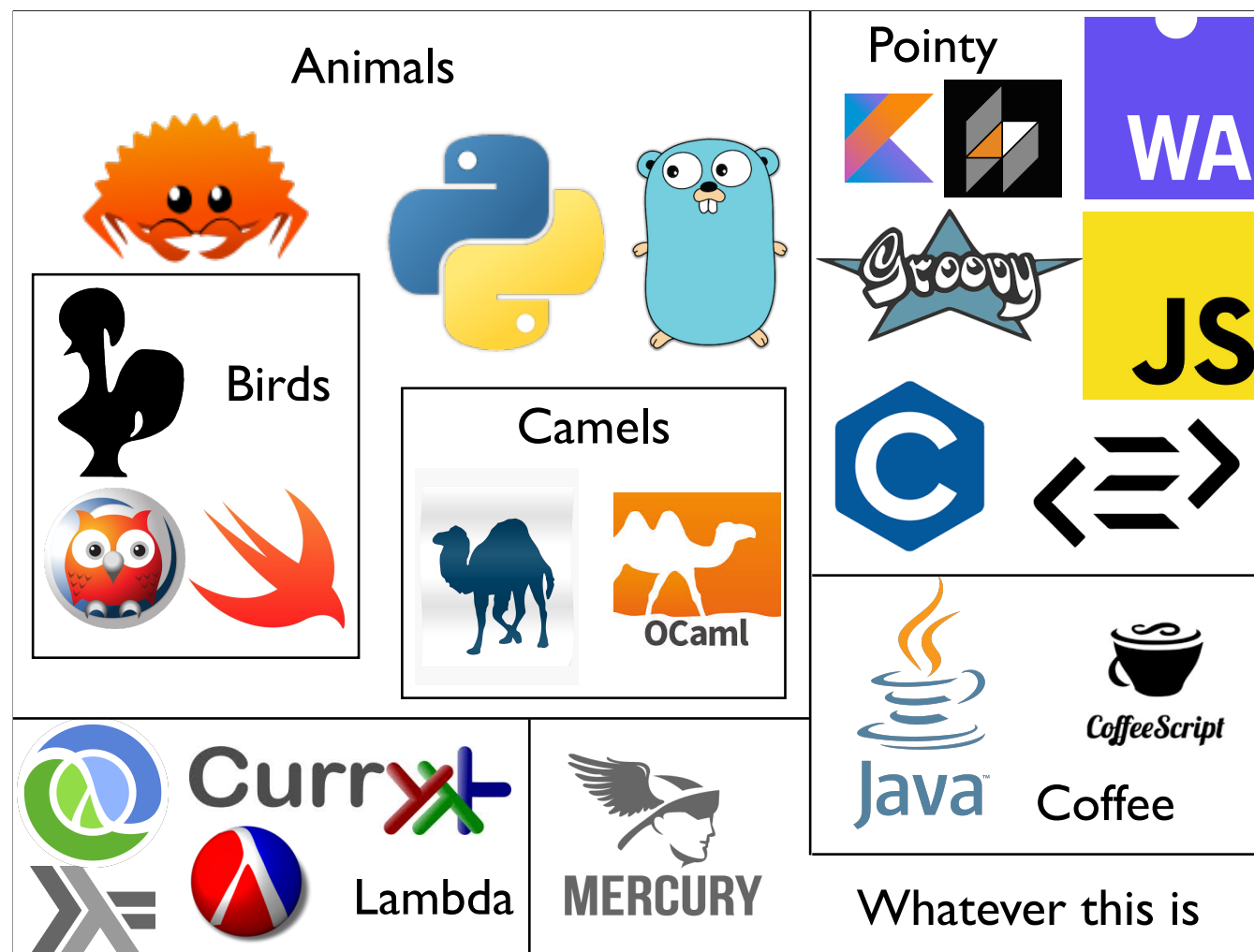
- This guy sucks, *I can't read his writing.*
- This class is boring, *it's way too slow.*
- This material is useless, *I don't see how it relates to anything in reality.*
- I can't fix anything if I don't know what's wrong

–I can actually do something about this!

# Why this Course?

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





- Without knowing about language features, we can't properly classify them
- If we can't classify them, we don't understand them, and we can't select the right tool for the job

# 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

–"Gotchas", meaning completely unintuitive behavior, usually leading to subtle bugs

–Surprisingly, there aren't that many language features out there. This is good for learning languages, but somewhat depressing (most features were developed in the 60's)

# Thinking About Programming



–Old adage: if all you have is a hammer, then every problem is a nail



-This is great if you have a nail



-If you have a screw?

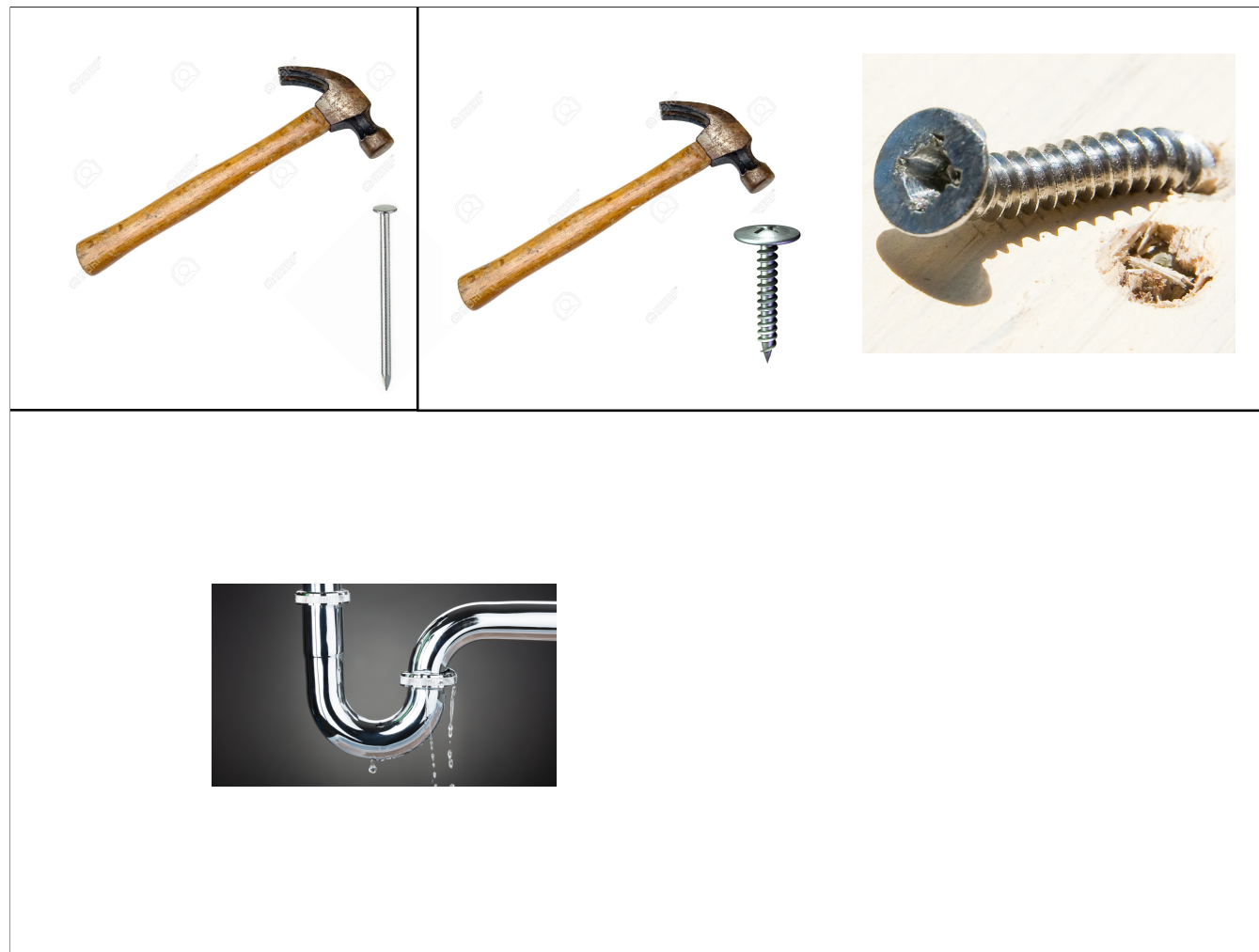




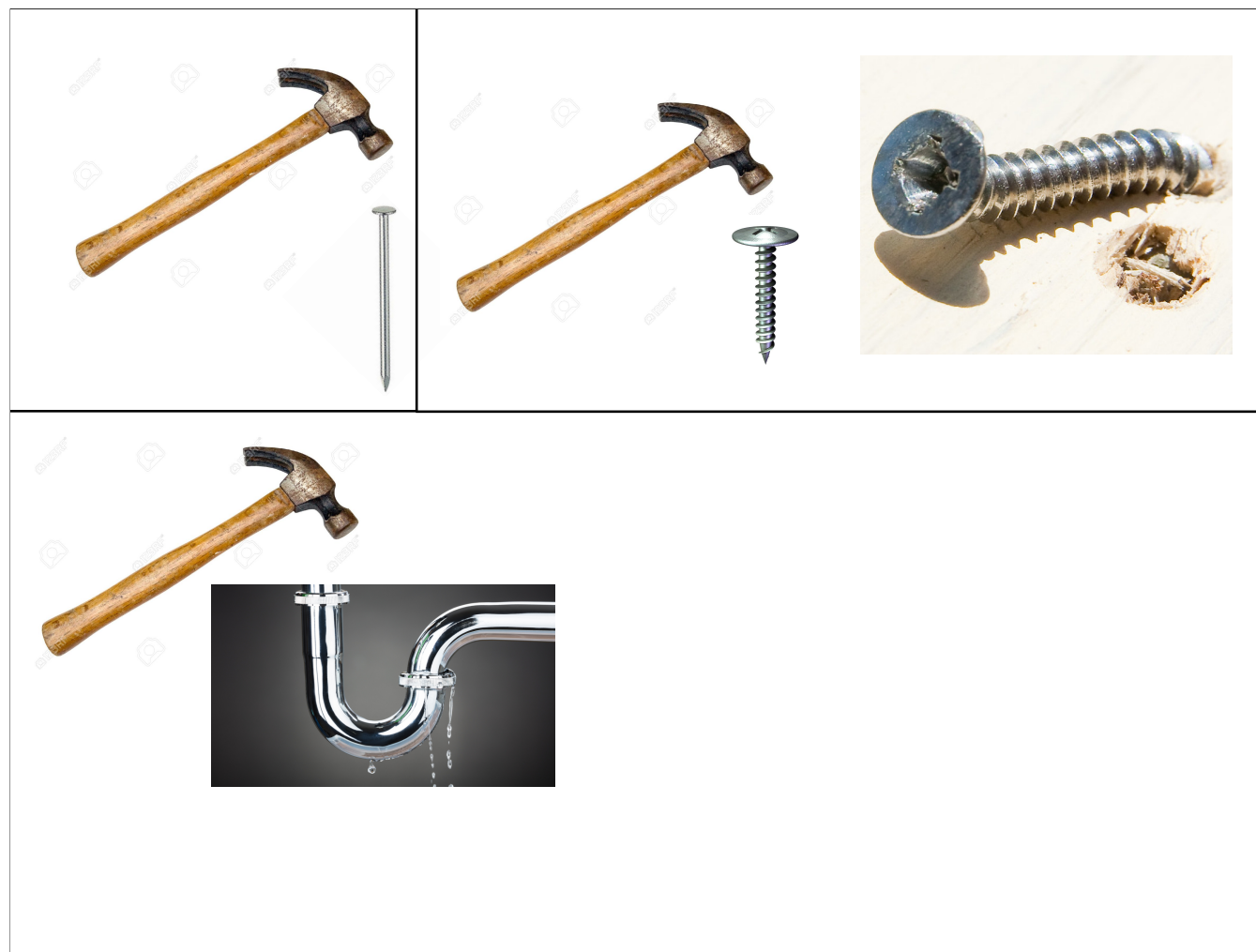
-You hit it with the hammer



-Ehh success?



-Leaky pipe?



-You hit it with the hammer!



-Leaks more?



-NEEDS MORE HAMMER



-Still leaking?





-HAMMER



# 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	Java
<pre>for {   a &lt;- Seq(1, 2, 3)   b &lt;- Seq("foo", "bar") } yield (a, b)</pre>	<ul style="list-style-type: none"><li>• Bulk of Summer</li><li>• Bulk of semester</li></ul>

- "Bulk of Summer": a student worked on something that did this for the bulk of a Summer
- "Bulk of semester": another student did a big part of this as part of a class project
- Four lines of code in Scala

# Common Misconceptions: Performance

# "Always Write the Fastest Code"

- "Premature optimization is the root of all evil" - Donald Knuth
- Programmer median salary: \$93,000/year
- AWS c7g.2xlarge (reserved): \$1,563/year
  - 8 cores, 16 GB RAM
- AWS c7g.16xlarge (reserved): \$12,506/year
  - 64 cores, 128 GB RAM

-This gets pushed to sell low-level, imperative languages

-Programmer median salary (2021): <https://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm>

# "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



26,322 Scala Jobs



### Scala BE Engineer - REMOTE

AvantStay West Hollywood, CA

Type Contractor

What are we looking for We are looking for a brilliant **Scala** Backend engineer who has the capacity and desire to help the engineering organization on a path to fast product cycles and exciting ...



### Scala Engineer NEW!

The Judge Group Dallas, TX

Type Contractor

Our client is currently seeking a **Scala** Engineer Your responsibilities will include rapid development of prototypes/concepts along with regular development. You are experienced with agile development ...



### Scala Spark Developer NEW!

United Software Group Inc Philadelphia, PA

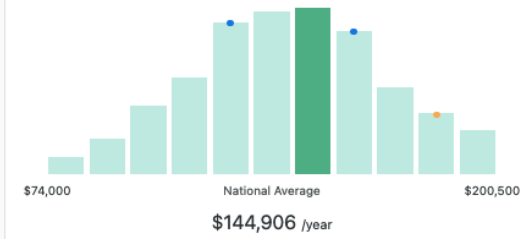
Type Full-Time

We are looking for a **Scala** Spark Developer Location \* Philadelphia, PA \* On-site | Full-time position. Skills \* Power BI / PowerQuery \* Must have knowledge on Spark/Spark SQL - Spark Core, Spark ...



### Spark/Scala Developer

#### How Much Do Scala Jobs Pay per Year?



#### What Is Scala?

Scala is a programming language that combines object-oriented and functional programming to create one high-level language. Its intention is Java Virtual Machine compatibility, and its different static types work to help avoid bugs in complex applications. Its JavaScript and JVM runtimes make it possible for people to have easy access to large ecosystems of libraries and build high-performance systems. Individuals, such as software engineers, can write Scala code to work on their project. They can also use this programming language with their existing Java code stack.

-Via Ziprecruiter

# "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

UAB - News - A 'high-speed Dr.

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

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

UAB News

Stay informed with the latest updates

Public Relations: (205) 934-3884 UAB Quicklinks

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 appointment, a faculty position at Harvard, and a profile in the *New Yorker*. It also led to the discovery that off-the-shelf drugs

f
t
i
l
y

**Featured Video**  
  
See more from UAB Visual Content

**Tweets by @UABNews**

–NASA NPAS (fault detection and response using ideas from logic programming): <https://techport.nasa.gov/view/94884>

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

University of Alabama at Birmingham (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

Public Relations: (205) 934-3884 UAB Quicklinks

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 appointment, a faculty position at Harvard, and a profile in the *New Yorker*. It also led to the discovery that off-the-



–NASA NPAS (fault detection and response using ideas from logic programming): <https://techport.nasa.gov/view/94884>

# 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"



-This is kind of like saying I know hammer and other hammer



# Staying in a Comfort Zone

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



–Pick up a screwdriver, already

# What this Course Is

- Heavy on programming
- Exposure to object-oriented, functional, logical, and a little parallel 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

–We don't have enough time to become experts on any of these topics; each one needs their own course (and hint hint there is a Logic Programming course (COMP 410))

–If you want language implementations, take compilers and language design (COMP 430)

# Languages We Will Use

- Java (class-based object-oriented programming)
- JavaScript (prototype-based object-oriented programming)
- Swift (functional programming)
- Prolog (logic programming)

# Why Java?

- 6th most popular language on StackOverflow
- OOP with class-based inheritance
- Even if you have used it, you may be rusty
- Statically typed, garbage collected, just-in-time compilation

–Lost one position since last year

# Why JavaScript?

- Most popular language on StackOverflow
- OOP with prototype-based inheritance
- Dynamically typed, garbage collected, (typically bytecode) interpreted, just-in-time compilers available

–It's prototype-based instead of class based, which is a different kind of object-oriented. Though classes are now a thing

# Why Swift?

- 19th most popular on StackOverflow, and 12th most loved
- Not *exactly* 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

–Was formerly 15th most popular and 6th most loved in 2019; went to 17th now 19th in 2022; 9th → 12th for most loved

–We'll probably not have time to cover typeclasses, but they work in a distinct manner from object-oriented classes, despite solving similar problems

# 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



# Syllabus