COMP 333 Lecture 1

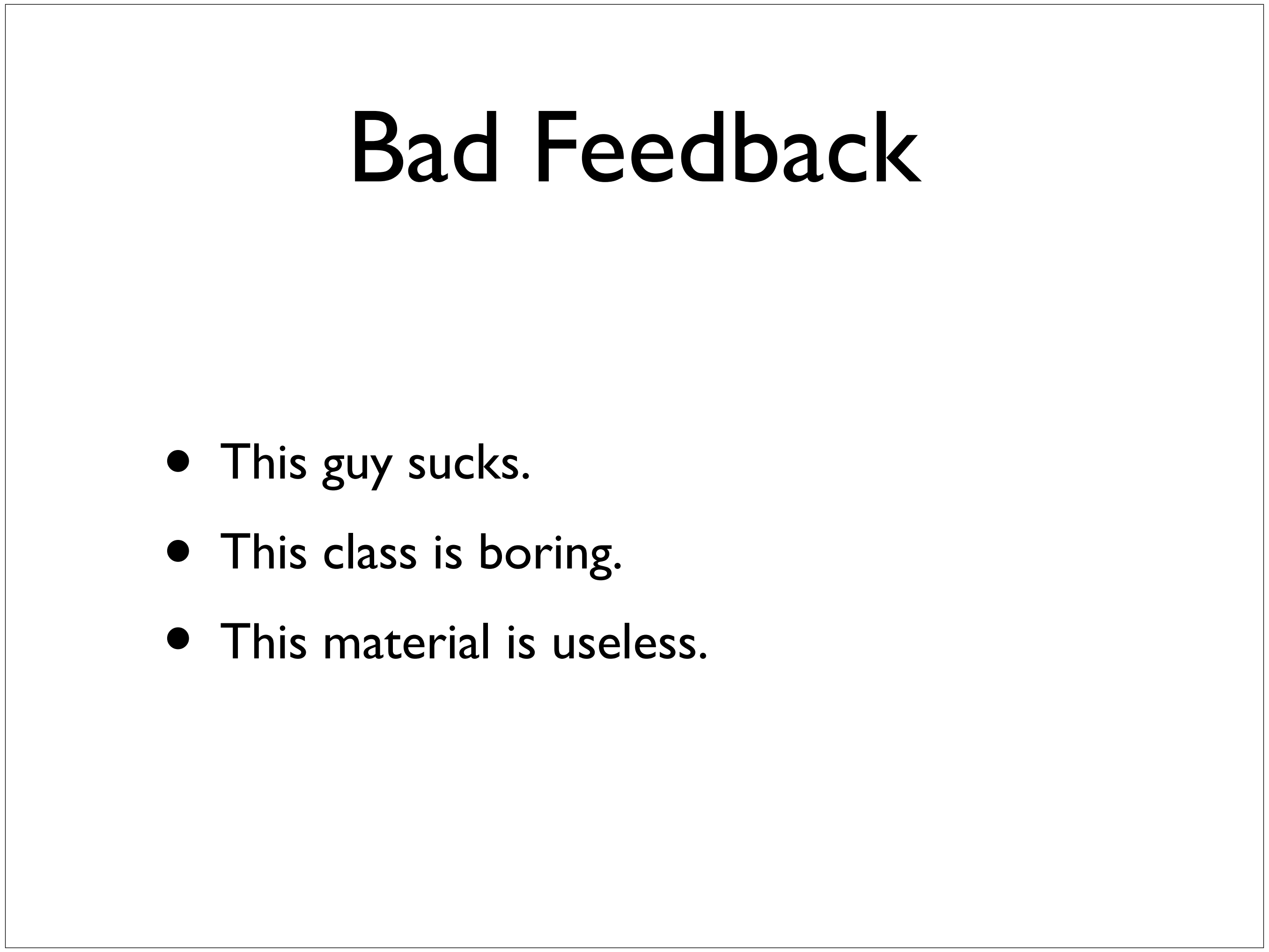
Kyle Dewey

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 second time teaching this class

About this Class

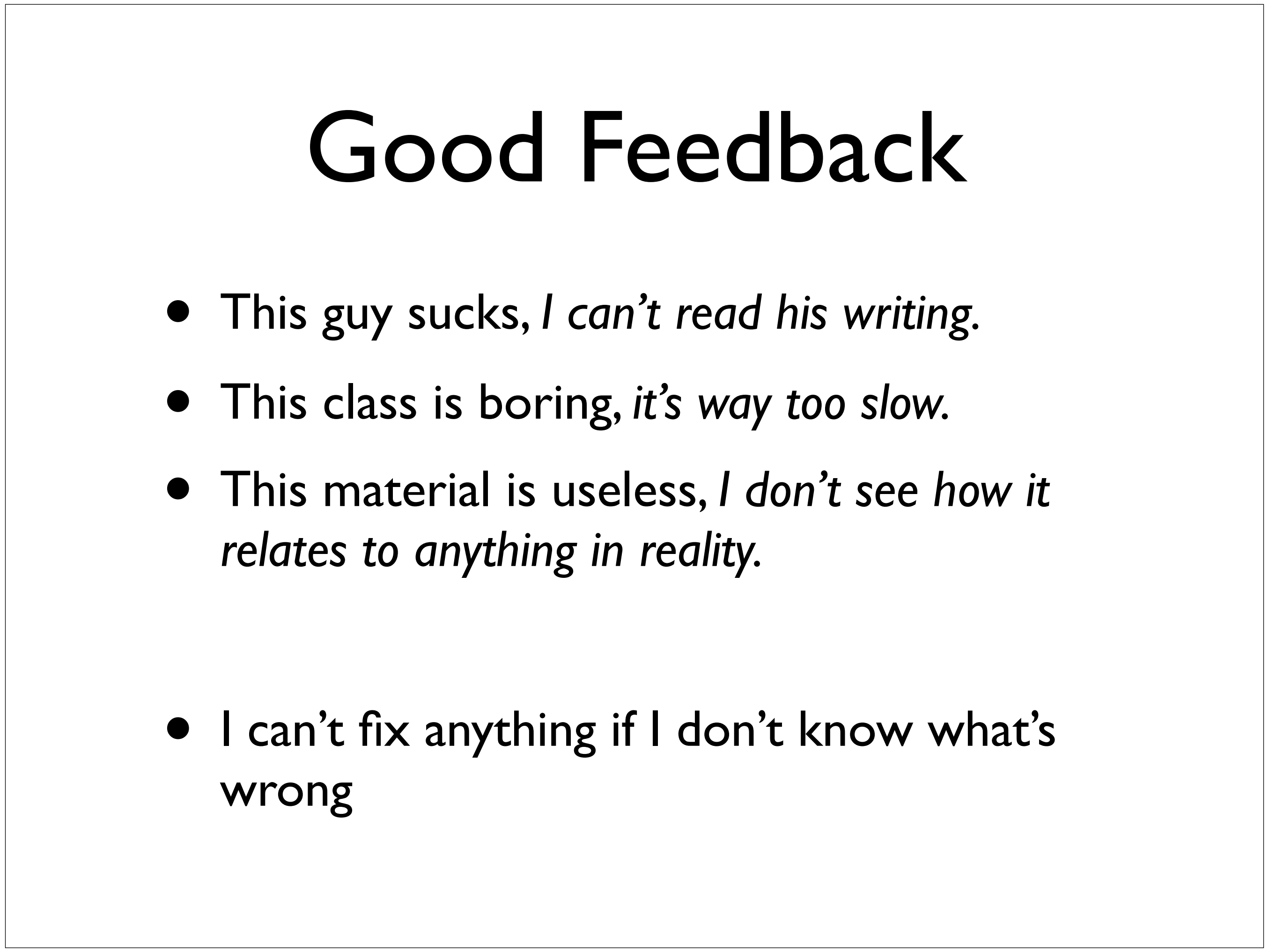
- See something wrong? Want something improved? Email me about it! (kyle.dewey@csun.edu)
- I generally operate based on feedback



Bad Feedback

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

-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

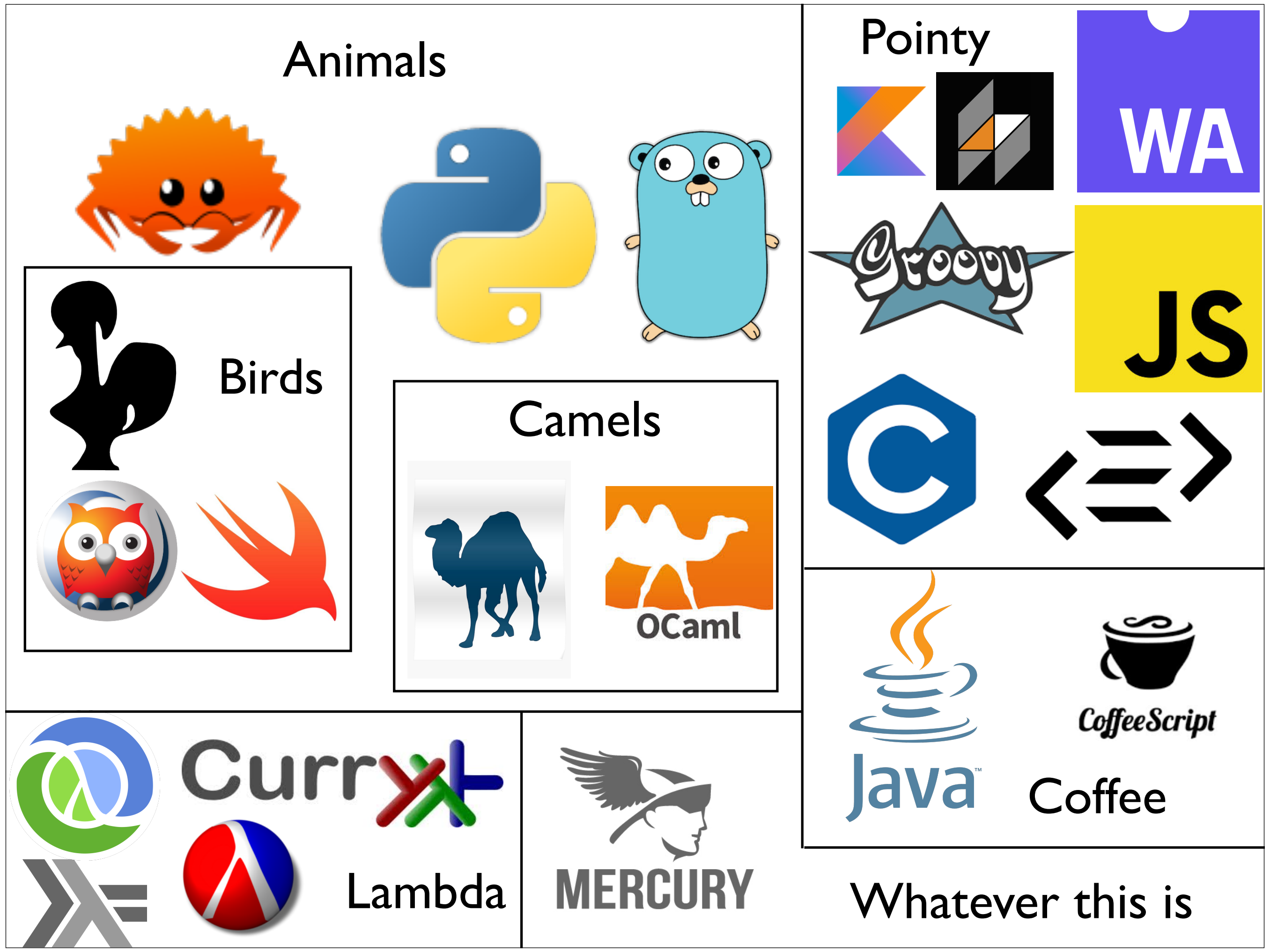
-I can actually do something about this!

Why this Course?

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

Navigating Languages

-There are a LOT of different programming languages.
-Many of these are similar to each other, and many are different
-Basic question: which should you use?



Animals

Birds

Camels

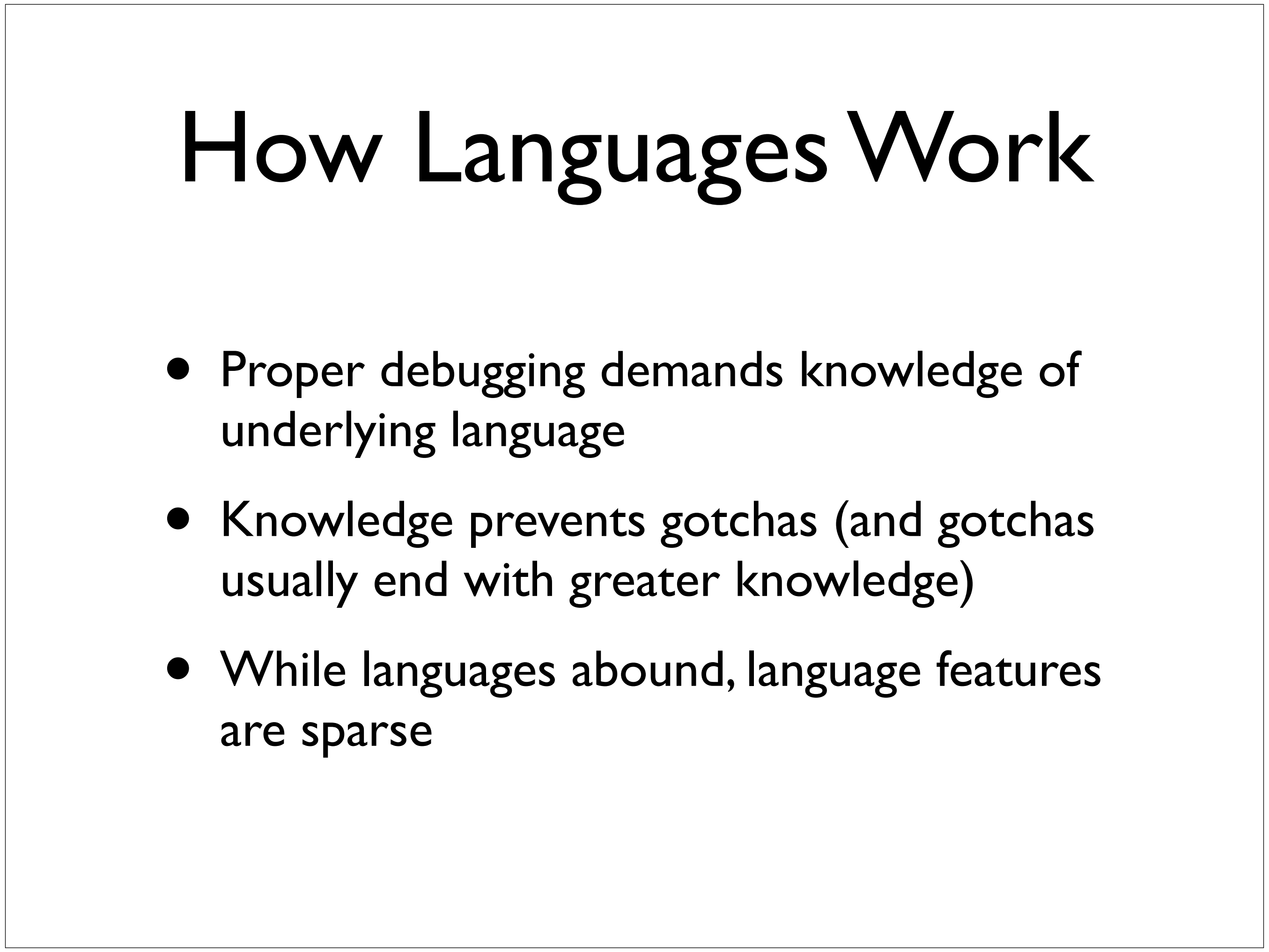
Pointy

Coffee

Lambda

Whatever this is

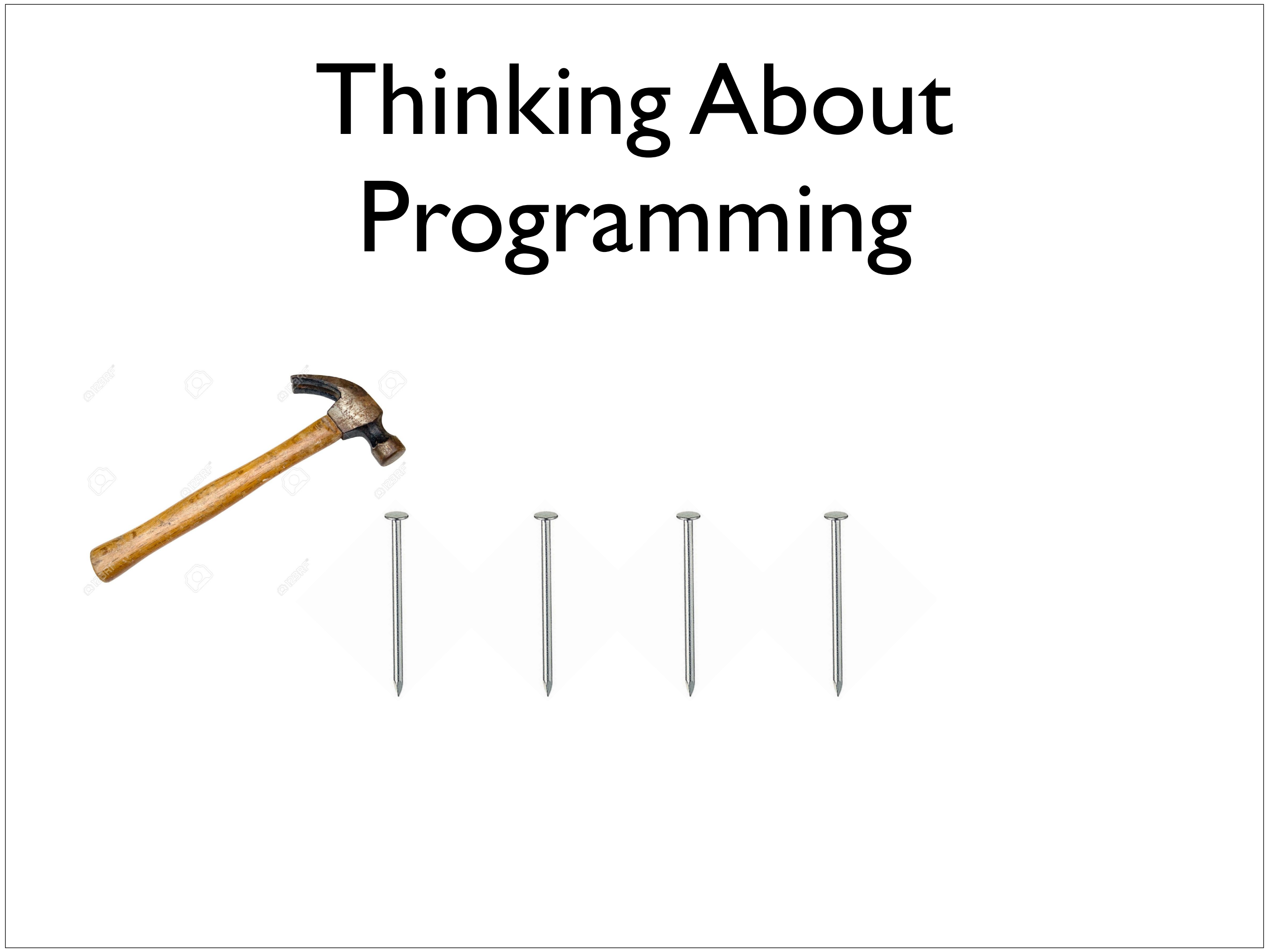
-Without knowing about language features, we can't properly classify them
-If we can't classify them, we don't understand them, and we can't select the right tool for the job



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

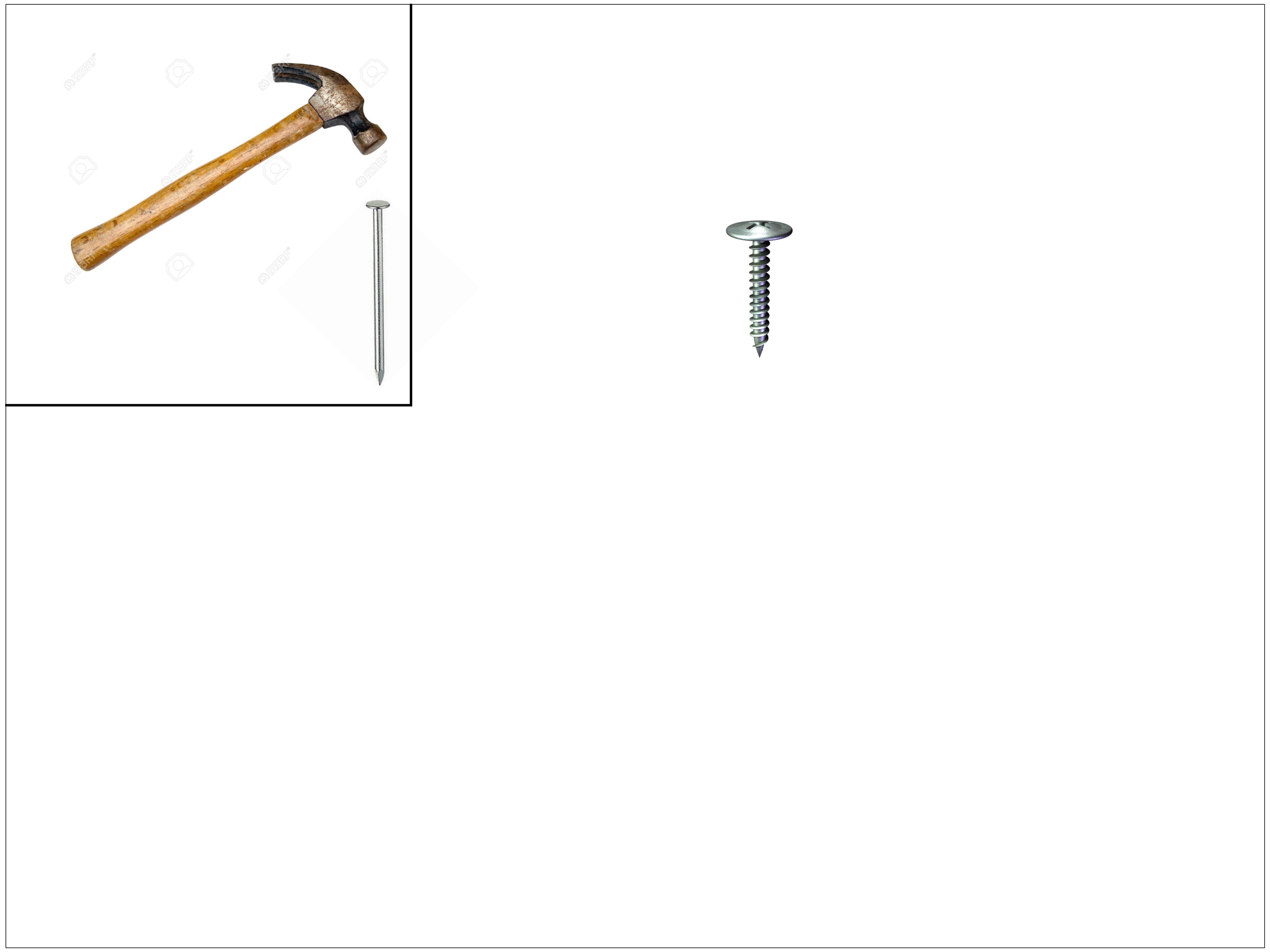
-"Gotchas", meaning completely unintuitive behavior, usually leading to subtle bugs
-Surprisingly, there aren't that many language features out there. This is good for learning languages, but somewhat depressing (most features were developed in the 60's)



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

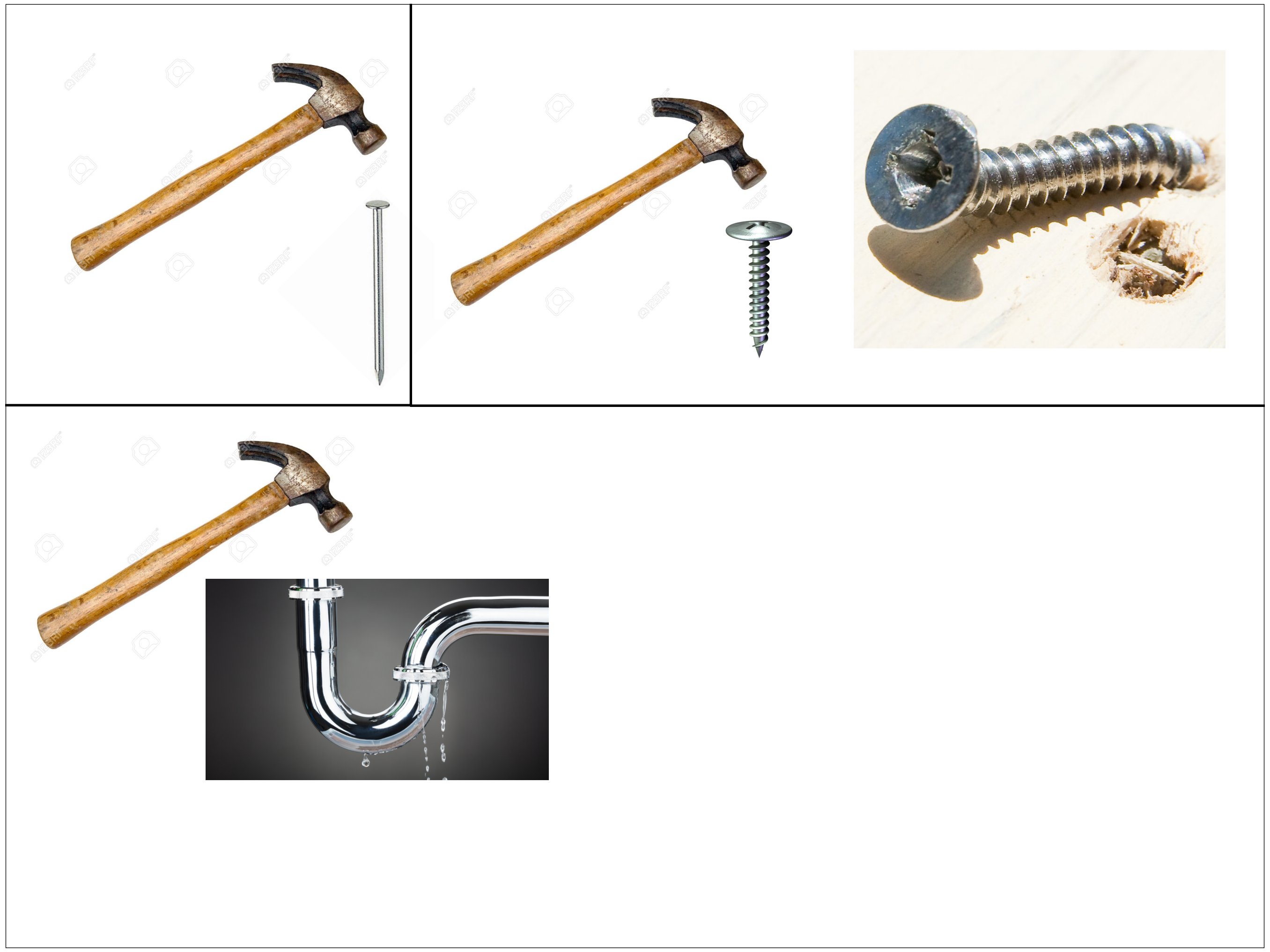


-If you have a screw?

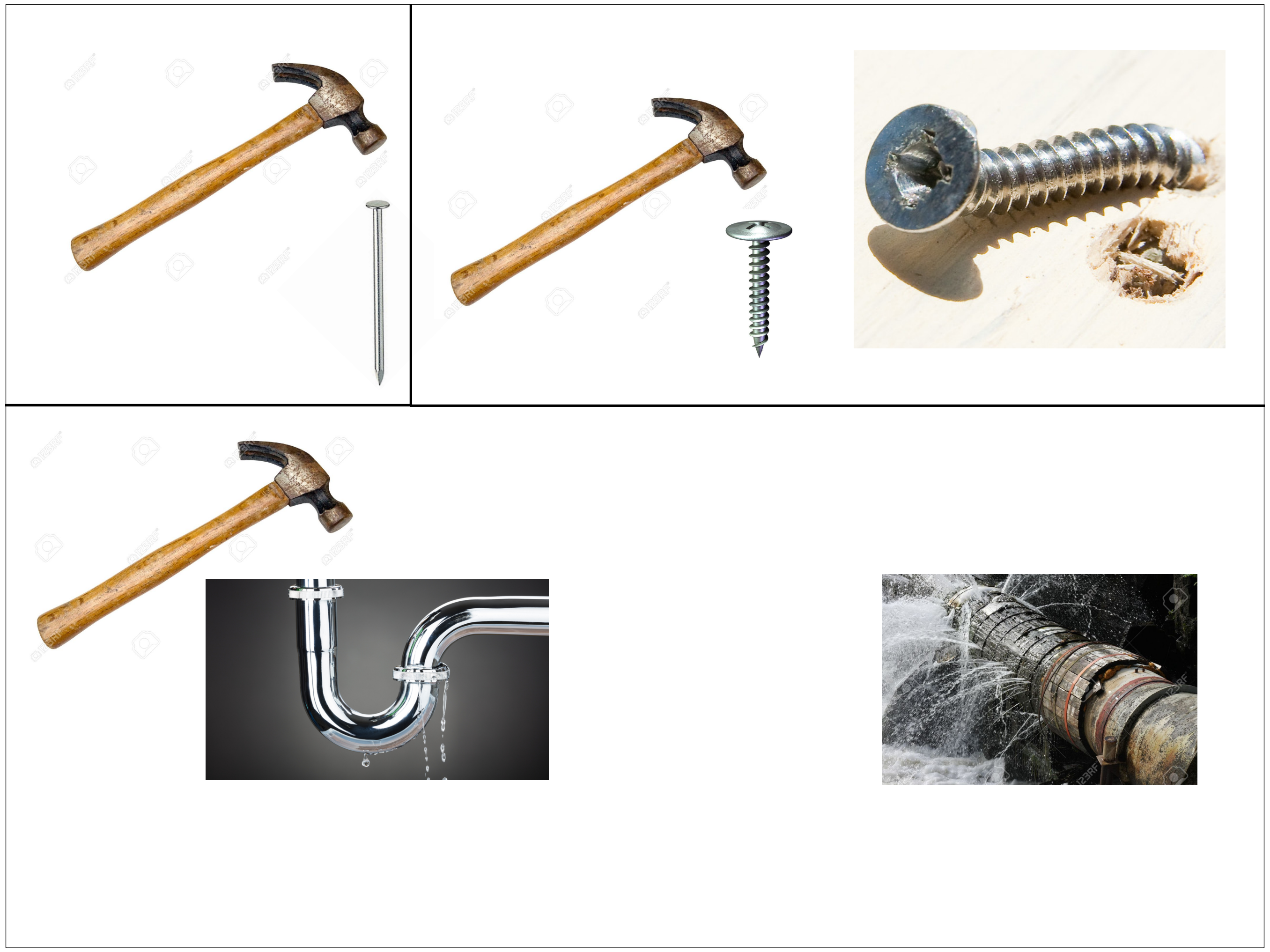
-You hit it with the hammer

-Ehh success?

–Leaky pipe?



-You hit it with the hammer!



-Leaks more?

-NEEDS MORE HAMMER

-Still leaking?

–HAMMER

The Point

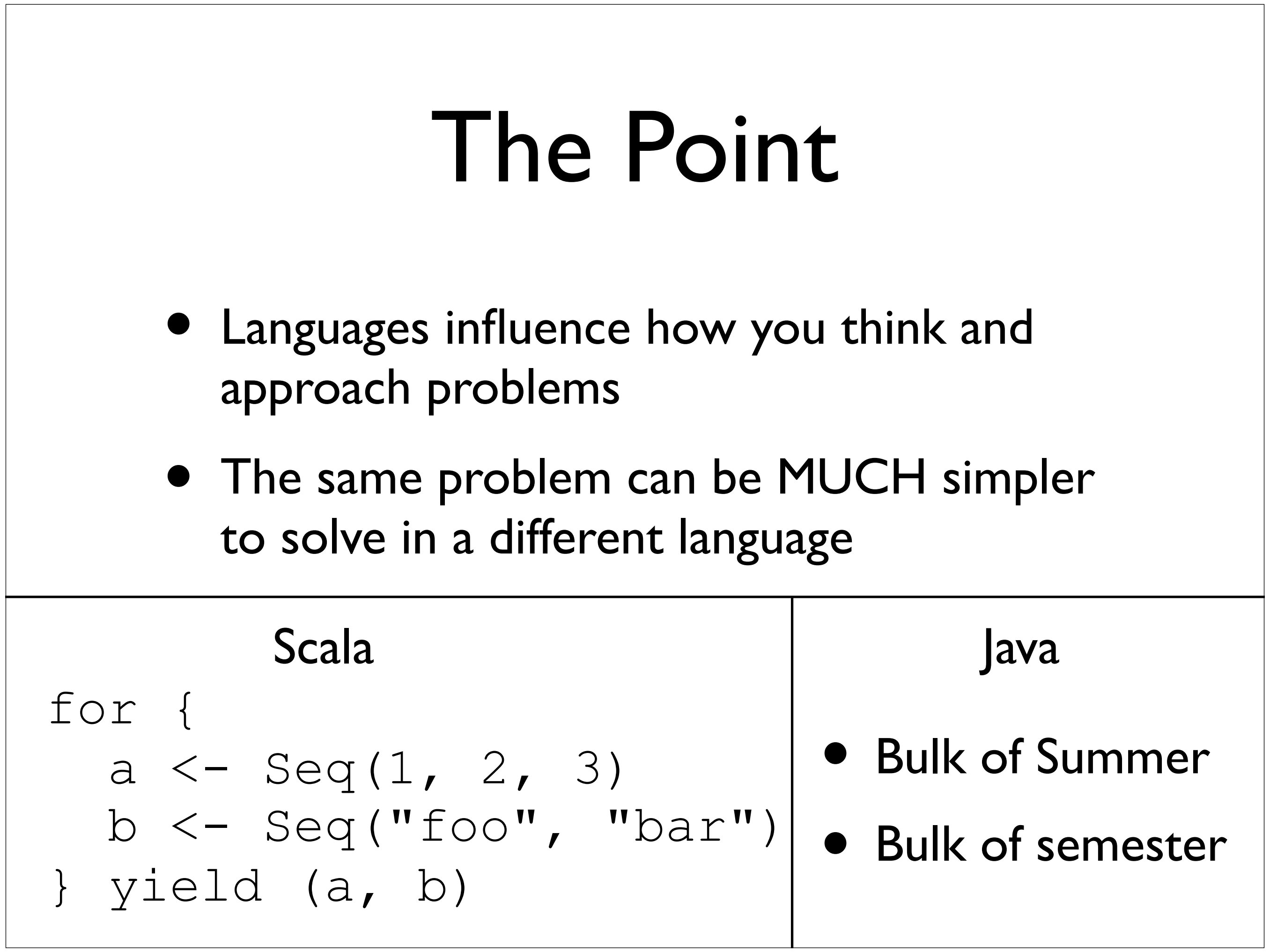
- Languages influence how you think and approach problems
- The same problem can be MUCH simpler to solve in a different language

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

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

Java

- Bulk of Summer
- Bulk of semester

-"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: $84,280/year
- AWS m4.large (reserved): $545/year

-This gets pushed to sell low-level, imperative languages
-Programmer median salary (2018): https://money.usnews.com/careers/best-jobs/computer-programmer/salary

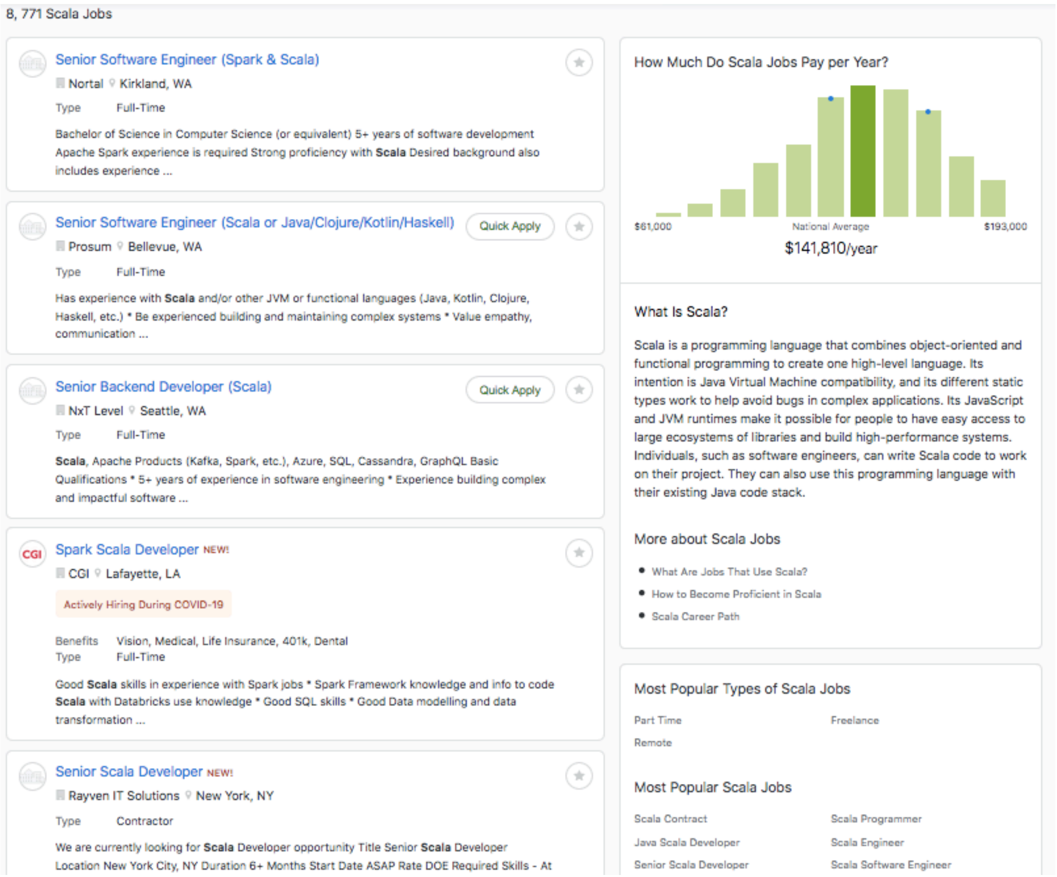
High-Level Languages are Slow

- 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



8, 771 Scala Jobs

Senior Software Engineer (Spark & Scala)

Nortal Kirkland, WA

Type Full-Time

Bachelor of Science in Computer Science (or equivalent) 5+ years of software development Apache Spark experience is required Strong proficiency with **Scala** Desired background also includes experience ...

Senior Software Engineer (Scala or Java/Clojure/Kotlin/Haskell)

Prosum Bellevue, WA

Type Full-Time

Has experience with **Scala** and/or other JVM or functional languages (Java, Kotlin, Clojure, Haskell, etc.) * Be experienced building and maintaining complex systems * Value empathy, communication ...

Senior Backend Developer (Scala)

NxT Level Seattle, WA

Type Full-Time

Scala, Apache Products (Kafka, Spark, etc.), Azure, SQL, Cassandra, GraphQL Basic Qualifications * 5+ years of experience in software engineering * Experience building complex and impactful software ...

Spark Scala Developer NEW!

CGI Lafayette, LA

Actively Hiring During COVID-19

Benefits Vision, Medical, Life Insurance, 401k, Dental
Type Full-Time

Good **Scala** skills in experience with Spark jobs * Spark Framework knowledge and info to code **Scala** with Databricks use knowledge * Good SQL skills * Good Data modelling and data transformation ...

Senior Scala Developer NEW!

Rayven IT Solutions New York, NY

Type Contractor

We are currently looking for **Scala** Developer opportunity Title Senior **Scala** Developer Location New York City, NY Duration 6+ Months Start Date ASAP Rate DOE Required Skills - At

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.

More about Scala Jobs

- What Are Jobs That Use Scala?
- How to Become Proficient in Scala
- Scala Career Path

Most Popular Types of Scala Jobs

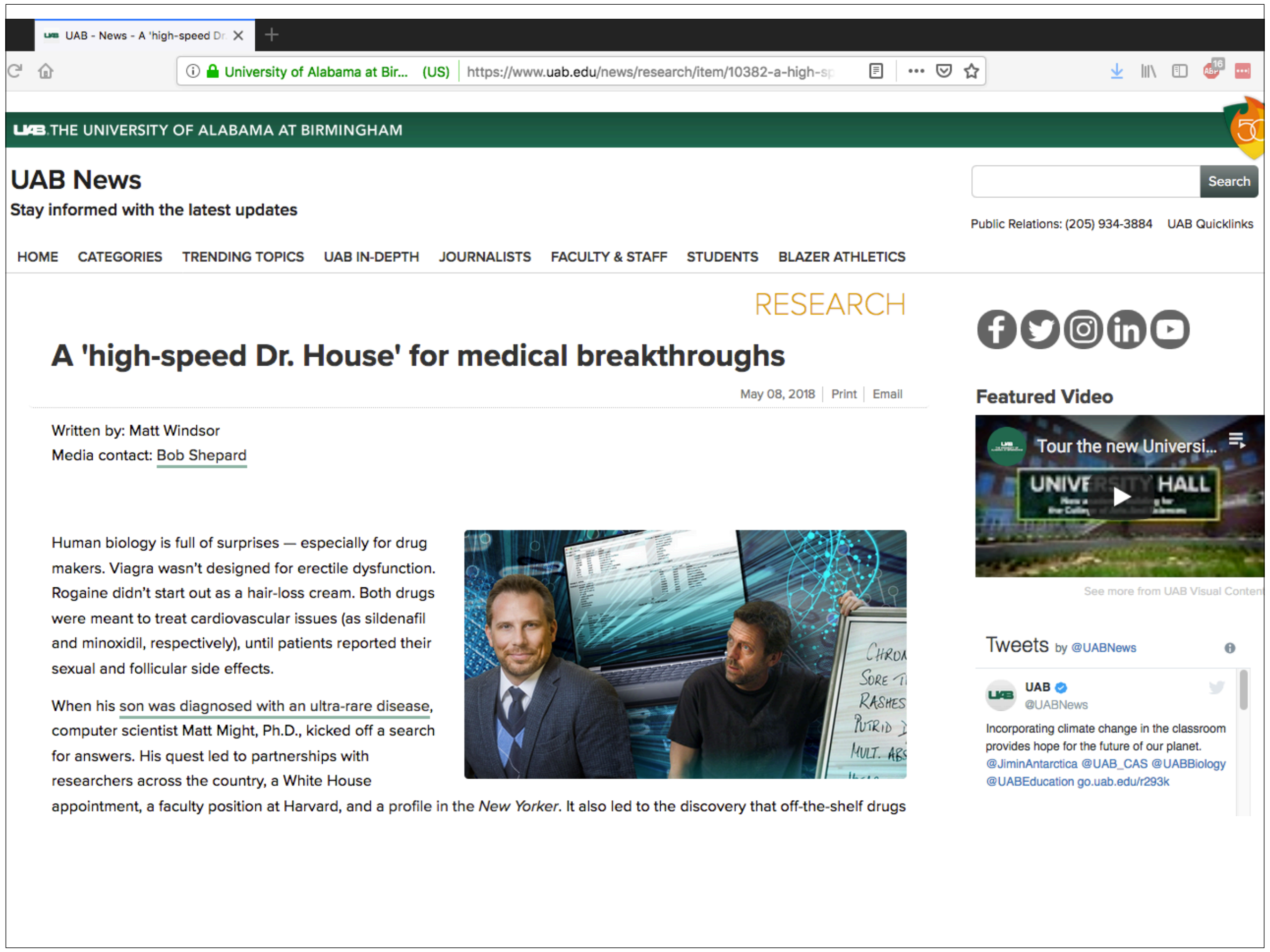
Part Time
Freelance
Remote

Most Popular Scala Jobs

Scala Contract
Scala Programmer
Java Scala Developer
Scala Engineer
Senior Scala Developer
Scala Software Engineer

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
UAB THE UNIVERSITY OF ALABAMA AT BIRMINGHAM
Search
Public Relations: (205) 934-3884
UAB Quicklinks
HOME
CATEGORIES
TRENDING TOPICS
UAB IN-DEPTH
JOURNALISTS
FACULTY & STAFF
STUDENTS
BLAZER ATHLETICS
Print
Email
Featured Video
Tour the new Universi...
UNIVERSITY HALL
See more from UAB Visual Conten

UAB News

Stay informed with the latest updates

RESEARCH

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

May 08, 2018

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.

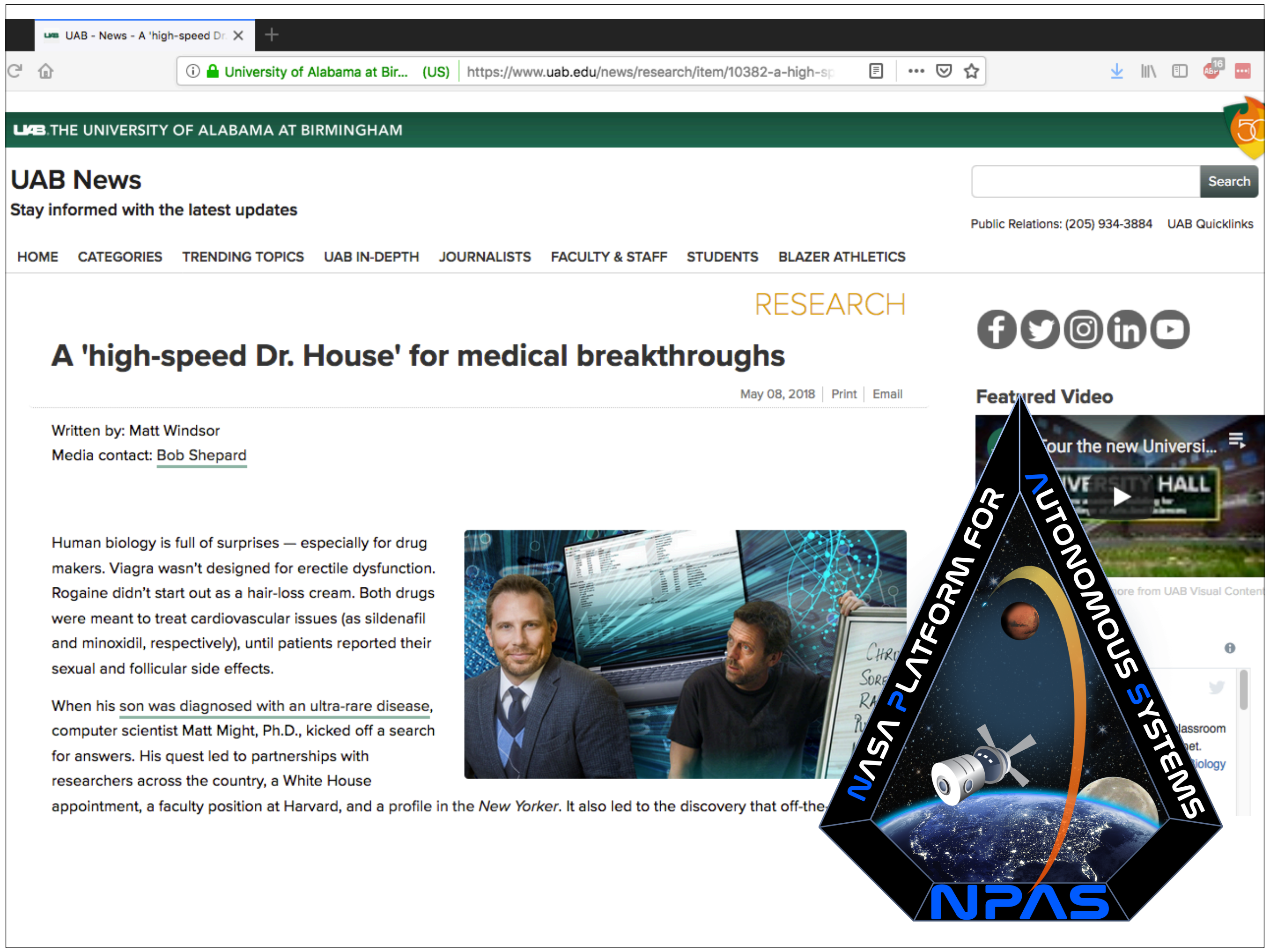
When his son was diagnosed with an ultra-rare disease, computer scientist Matt Might, Ph.D., kicked off a search for answers. His quest led to partnerships with researchers across the country, a White House appointment, a faculty position at Harvard, and a profile in the *New Yorker*. It also led to the discovery that off-the-shelf drugs

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

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


UAB - News - A 'high-speed Dr
University of Alabama at Bir... (US)
https://www.uab.edu/news/research/item/10382-a-high-sp
NASA PLATFORM FOR AUTONOMOUS SYSTEMS
NPAS

UAB THE UNIVERSITY OF ALABAMA AT BIRMINGHAM

UAB News

Stay informed with the latest updates

Search

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

Featured Video

Tour the new Universi...

-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

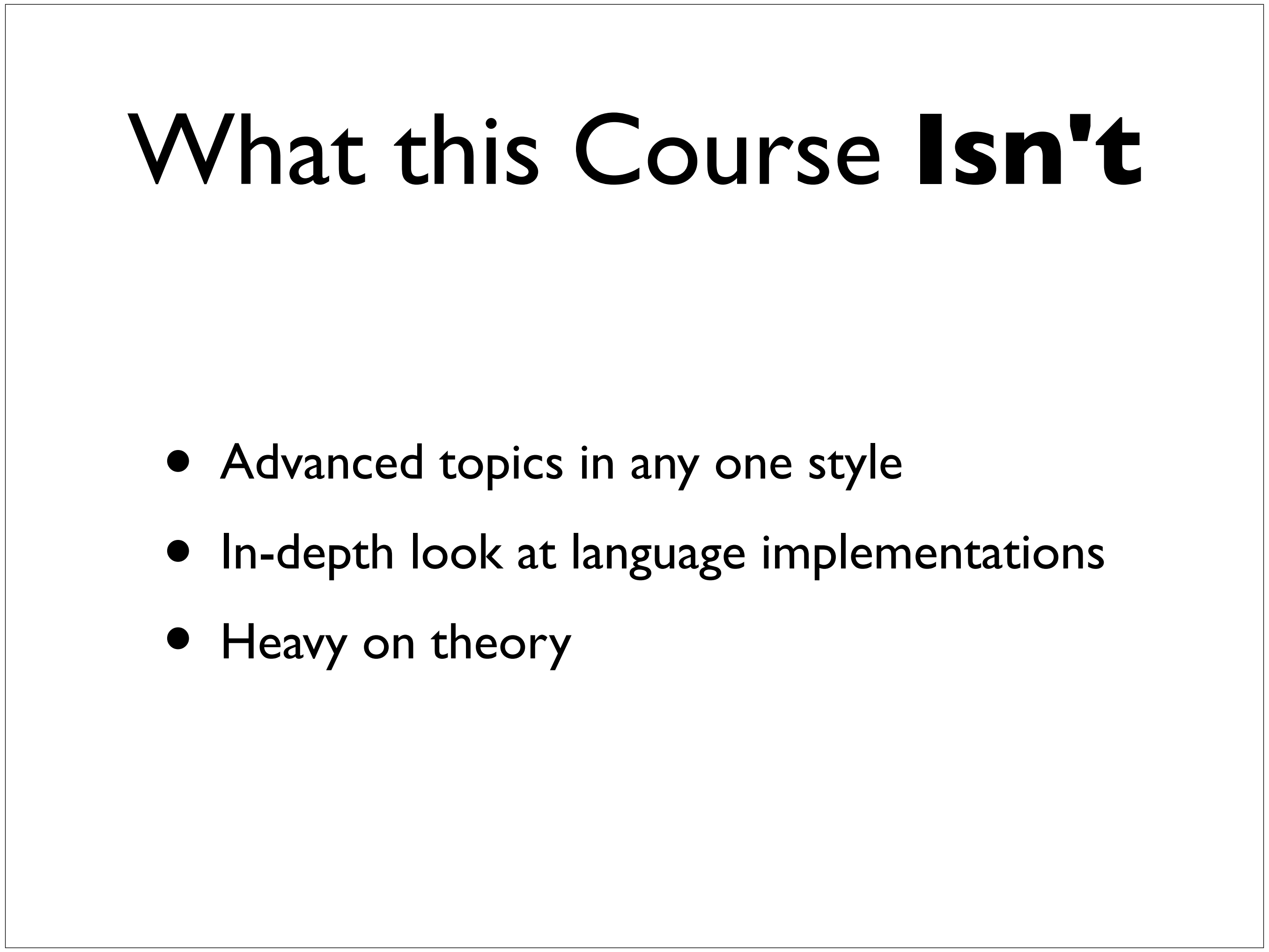
Staying in a Comfort Zone

- "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

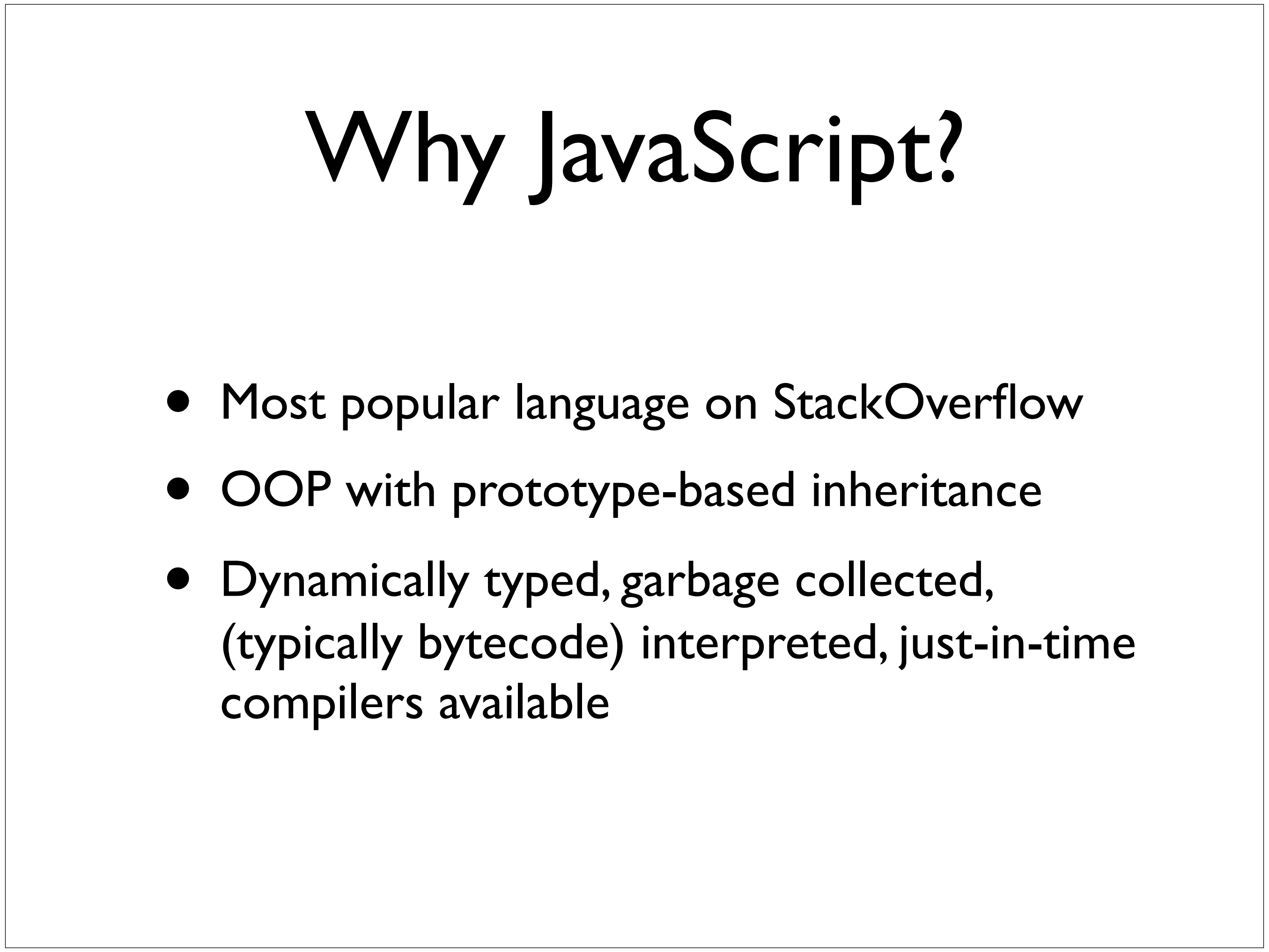
-We don't have enough time to become experts on any of these topics; each one needs their own course (and hint hint there is a Logic Programming course (COMP 410))
-If you want language implementations, take compilers and language design (COMP 430)

Languages We Will Use

- Java (class-based object-oriented programming)
- JavaScript (prototype-based object-oriented programming)
- Swift (functional programming)
- Prolog (logic programming)
- Java 8 (parallelism)

Why Java?

- 5th 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



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?

- 17th most popular on StackOverflow, and 9th 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; this will probably be the last year we use Swift if it keeps declining
-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