

COMP 333 Introduction

Kyle Dewey

About Me

- My research:
 - Automated program testing + CS education
 - Programming language design
- 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)
- 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

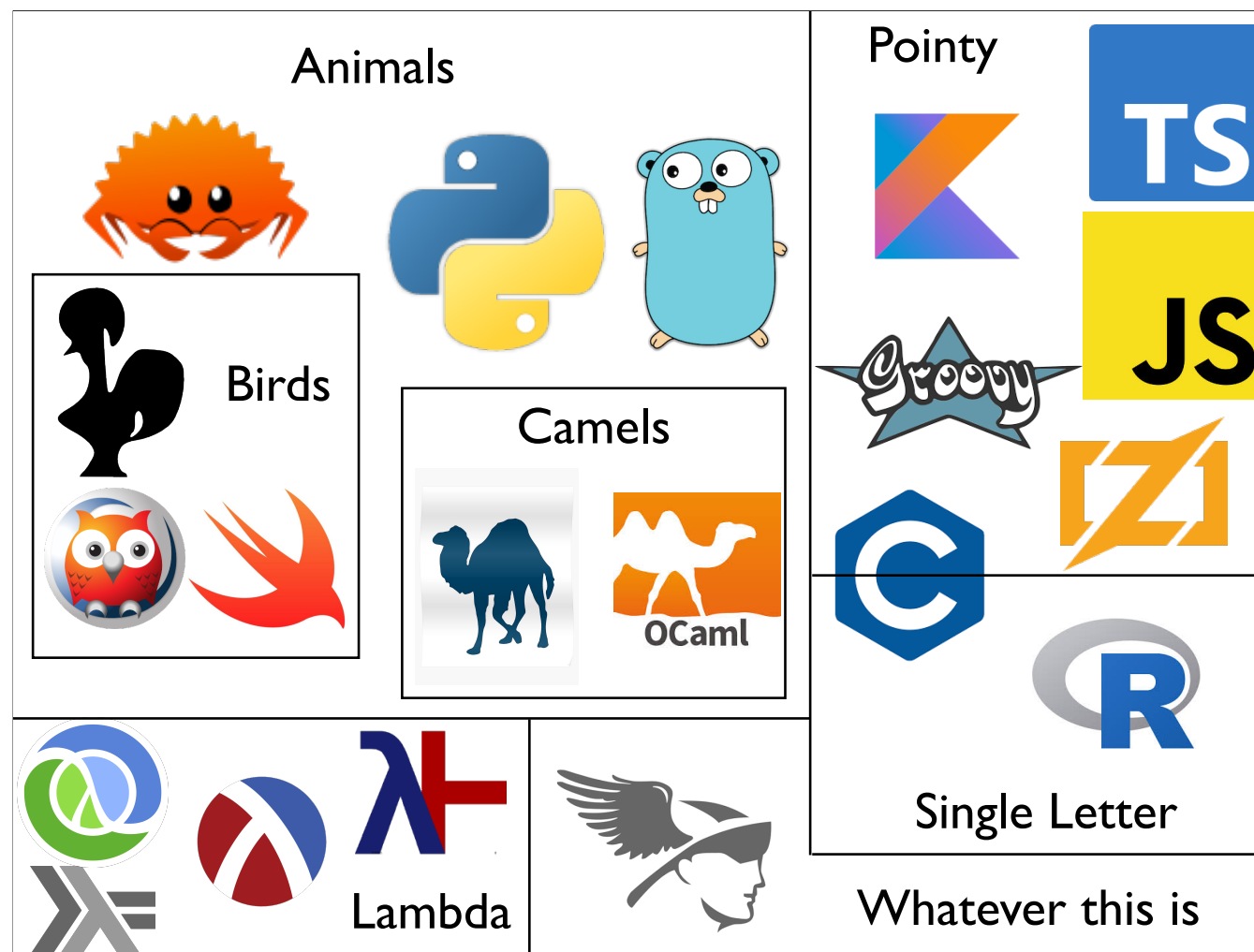
–I can actually do something about this!

Why this Course?

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



- 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?



- 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?



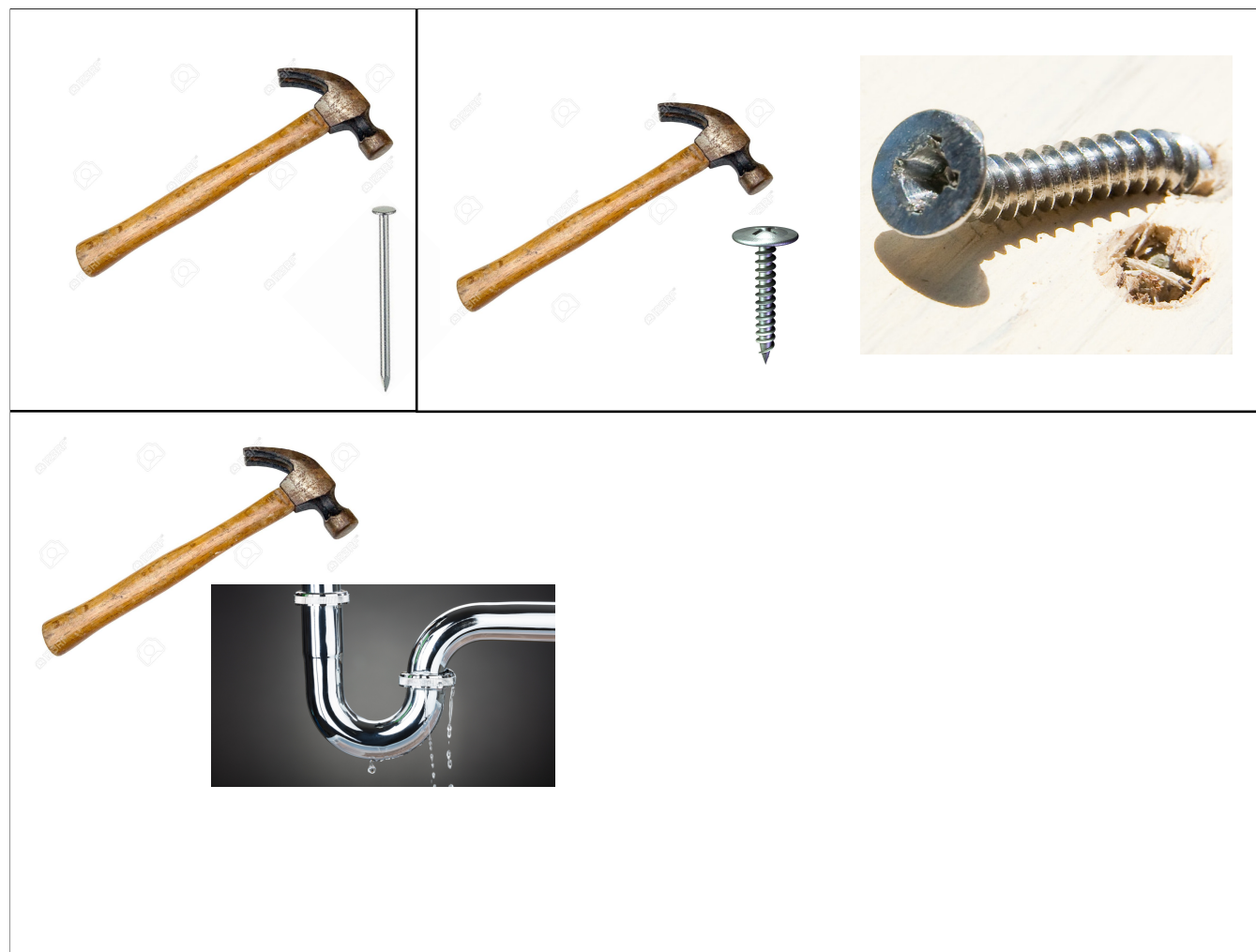
-You hit it with the hammer



-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 <- Seq(1, 2, 3) b <- Seq("foo", "bar") } yield (a, b)</pre>	• 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: \$98,670/year (2024)
- AWS c7g.2xlarge (reserved 3 yr): \$972/year
 - 8 cores, 16 GB RAM
- AWS c7g.16xlarge (reserved 3 yr): \$7,773/year
 - 64 cores, 128 GB RAM

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

-Programmer median salary (2024): <https://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm>. That site has unfortunately not been updated since 2024 due to funding cuts (<https://www.bls.gov/bls/2025-lapse-revised-release-dates.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

Scala Developer Salary in Los Angeles, CA

Yearly Monthly Weekly Hourly Table View

\$43.1K \$150.9K \$170.2K

\$139,374/year
\$67.01/hour

Scala Developer Salary Comparison by Location

Location	Salary
Nationwide United States	\$129,348
Los Angeles California	\$139,374

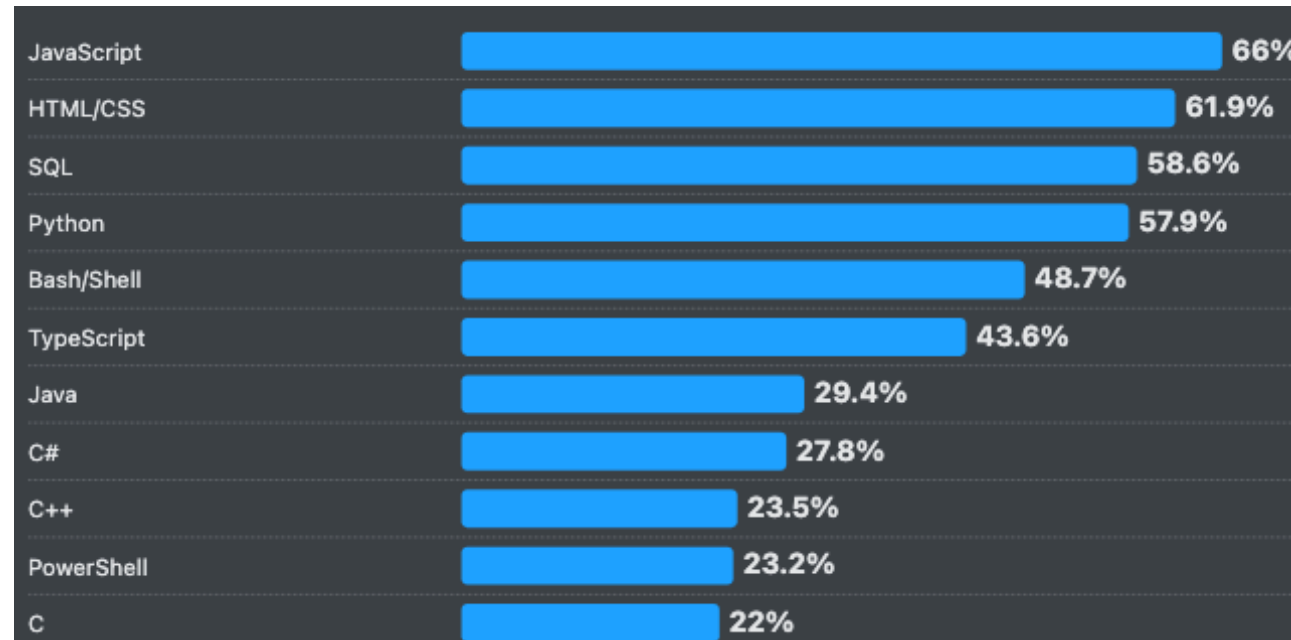
-Via Ziprecruiter – August 2024

Common Misconceptions: Stagnation

"Industry Moves Slowly"

- COBOL was once a vital language
- Perl was once the champion of the Internet
- Java was once most popular
- Companies that cannot adapt, die

"Industry Moves Slowly"



StackOverflow 2025 Developer Popularity Survey

Since last year, Bash/Shell has increased significantly to surpass TypeScript, and Powershell now outranks C.

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, and a little low-level 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, functional programming)
- (TINY bit of) Rust (low-level programming with high-level features, memory-focused)

Why Java?

- 7th most popular language on StackOverflow
- OOP with class-based inheritance
- Even if you have used it, you may be rusty, and you might not have used all the relevant functionality
- 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 Rust?

- 14th most popular on StackOverflow, and most admired language
- Imperative and functional feature set
- Low-level language (fine-grained memory control, pointers, no runtime environment, compiles to machine code)
- ...with traditionally high-level features (algebraic data types, pattern matching, higher-order functions, typeclasses, type inference)

-Has consistently been most admired for years (category formerly called "most loved" on StackOverflow)

-Over the past couple years, it has begun to supplant C/C++ code in major projects. Can be used in Linux kernel code, has Linus Torvalds' blessing, and is used for some device drivers. This is HUGE because even C++ was considered too high-level and overall non-viable for this.

-From my own experiences: I'm willing to bet that Rust will eventually overtake C/C++, but it will take a long time to do so (likely ~20 years)

Syllabus