

# Go & Rust

MF, KM, LS, NN

# Modern Languages

Our group picked emergent languages that have experienced a surge in popularity over the last couple of years.

- Rust

- Started by a Mozilla employee, now a Mozilla- and community-driven effort.

- Go

- Started by Robert Griesemer, Rob Pike, and Ken Thompson, now a Google-backed effort.
- (Not to be confused with another, similarly-named language, called "Go!")

# With Similar Goals

- Rust:

- fast execution (like C and C++)
- immutability by default (unlike C and C++)
- memory safety (ownership)
- concurrency
- emphasis on correctness
- crude garbage collection (more from ownership)

- Go:

- ease of coding (dynamic typing)
- emphasis on fast compilation (clean dependencies management)
- safety (no pointer arithmetic)
- multiprocessing and concurrency
- built-in garbage collecting
- built-in networking

# Both Born From A Dissatisfaction With C and C++

Rust addresses:

- problems with concurrent access to variables (now immutable by default; *Arcs*)
- clunky syntax for concurrency (tasks, interprocess communication via *channels*)

Go addresses:

- slow compilation of large programs
- confusing syntax features that too often lead to bugs (compare *int\* a,b;* with *a,b \*int*)
- repetition; no more *Object a = b.function(blah);*

# Goods and Bads

## Readability and Writability

- Due to a lack of familiarity, they can be hard to understand
  - Go has slices, which are weird
  - Go reverses variable declarations (though for good reason)
  - Rust uses function declarations that look more like ML or functional programs than like C or C++
  - Rust uses *match* statements, which are useful, but again unfamiliar to the C{++}? programmer

# Goods and Bads

## Reliability

- Rust prides itself on encouraging correctness, and therefore reliability, but...
- It's kind of hard to take that seriously while it's still in alpha.
- Likewise, Go is also new (though somewhat more stable), so it's hard to suggest it for production use.
- They'll grow, but they need an eager and competent community to make that happen.

# That Said

They're both cool languages, with some great ideas behind them.

If they actually complete their goals, they would deserve to be called "amazing".

\*&

## Rust 30-Minute Intro

<http://static.rust-lang.org/doc/master/intro.html>

## Interview With Ken Thompson

<http://www.infoq.com/news/2012/08/Interview-Rust>

## The Go FAQ

<http://golang.org/doc/faq>