CMSC 330: Organization of Programming Languages

Safe, Low-level Programming with Rust
The C programming language

• Was invented in the 70s but is still among the most popular languages. Why?
  – Low-level control, careful memory management
  – Despite serious risks
• Why not a type-safe language?
  – GC de-facto requirement of type safety
    • Increases memory footprint
    • Adds unpredictable pauses
  – Also: important limitations on concurrency patterns
• Can’t we do better?
Type safety and low-level control

• Several research projects in the 90s and 2000s on type safety without GC
  – MLKit, Vault (Microsoft), Cyclone (Cornell, AT&T, UMD, UW) – see http://cyclone.thelanguage.org
• Likewise, lots of work on safe concurrency
  – Cyclone (in principle), Java dialects, software transactions (language add-on)
• Big challenge: Sufficient expressiveness
  – Cyclone made it the furthest, but fizzled
  – Finding the right tradeoffs and balance is a big task
Rust

- Begun in 2006 by Graydon Hoare
- Sponsored as full-scale project and announced by Mozilla in 2010
  - Changed a lot since then; source of frustration
  - But now: most loved programming language in Stack Overflow annual surveys of 2016, 2017, and 2018
- Takes ideas from functional and OO languages, and recent research
- Key properties: Type safety despite use of concurrency and manual memory management
  - And: No data races
Features of Rust

- **Lifetimes and Ownership**
  - Key feature for ensuring safety
- **Traits** as core of object system
- **Variable default is im muta bility**
- **Data types and pattern matching**
- **Type inference**
  - No need to write types for local variables
- **Generics** (aka parametric polymorphism)
- **First-class functions**
- **Efficient C bindings**
Rust in the real world

• Firefox Quantum and Servo components
  – https://servo.org
• REmacs port of Emacs to Rust
  – https://github.com/Wilfred/remacs
• Amethyst game engine
  – https://www.amethyst.rs/
• Magic Pocket filesystem from Dropbox
• OpenDNS malware detection components
Information on Rust

- Rust book free online
  - We will follow it in these lectures
- More references via Rust site