Lecture 1

Rust Setup

Goals For Today

- Discuss more about the course
- Explore the Rust environment
- Hello, CS 128 Honors!

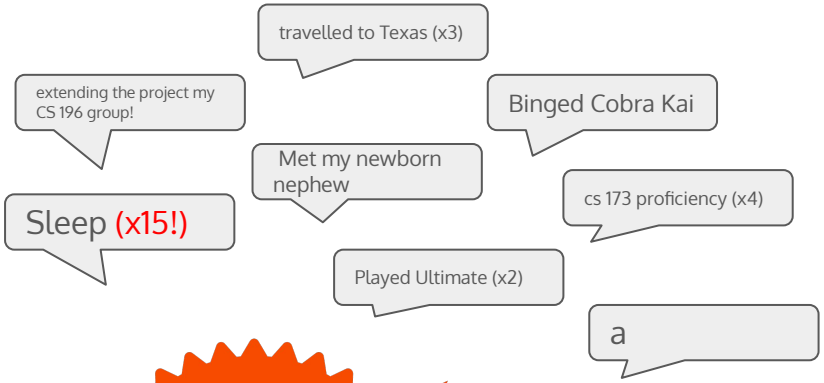
Quick Reminders

Reminders:

- Fill out the checklist form
  - If you're a James Scholar, submit your HCLA
  - If you're not a James Scholar, make sure you're registered on Self Service
- Join the Discord

But First...

What did you all do this break?



This is Ferris the Crab, the Rust mascot

But First...

What are your favorite pastas?

Spaghetti (x12)

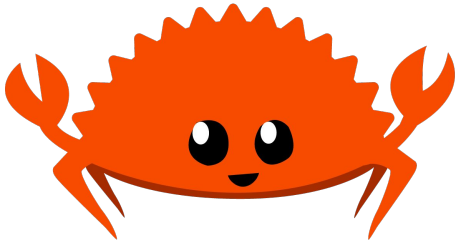
a

Penne (x7)

Conchiglie (x2)

Tortellini (x3)

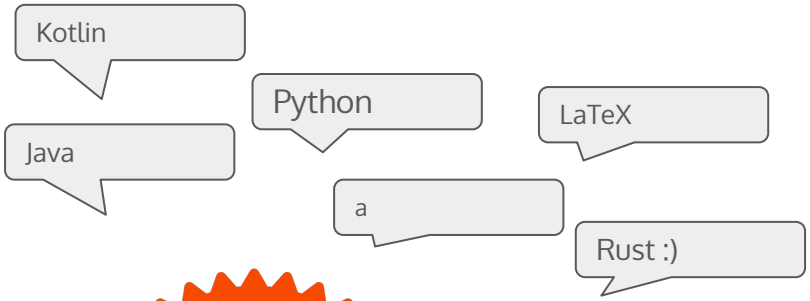
idk, the normal kind



This is Ferris the Crab, the Rust mascot

But First...

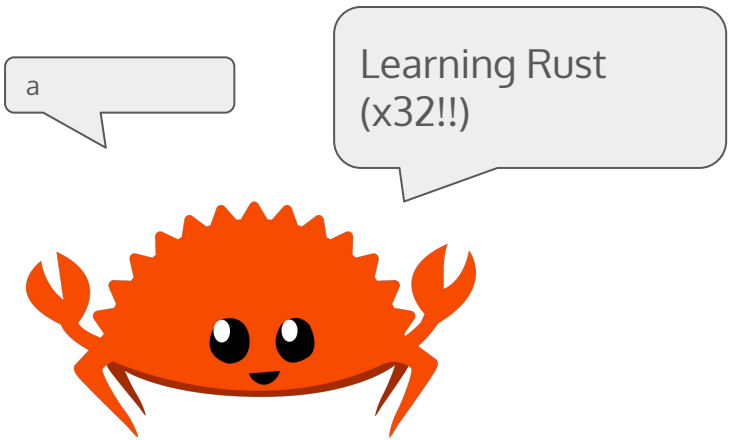
What are your favorite languages?



This is Ferris the Crab, the Rust mascot

But First...

Why are you taking this course?



This is Ferris the Crab, the Rust mascot

This Course

- This is the Honors add-on for CS128
- In this course, we'll (initially) follow topics you learn in CS128, but in Rust
- Eventually we'll branch off to more topics

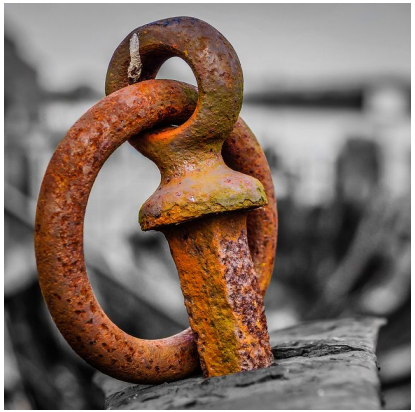
This Course

- This is the Honors add-on for CS128
- In this course, we'll (initially) follow topics you learn in CS128, but in Rust
- Eventually we'll branch off to more topics

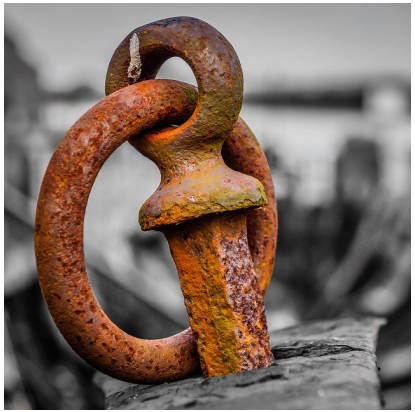
- This course is structured for you to get out as much as you put in

So... What is Rust?

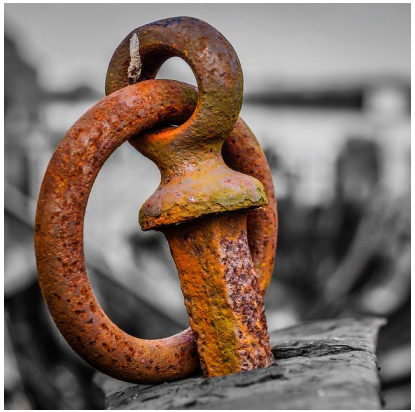
So... What is Rust?



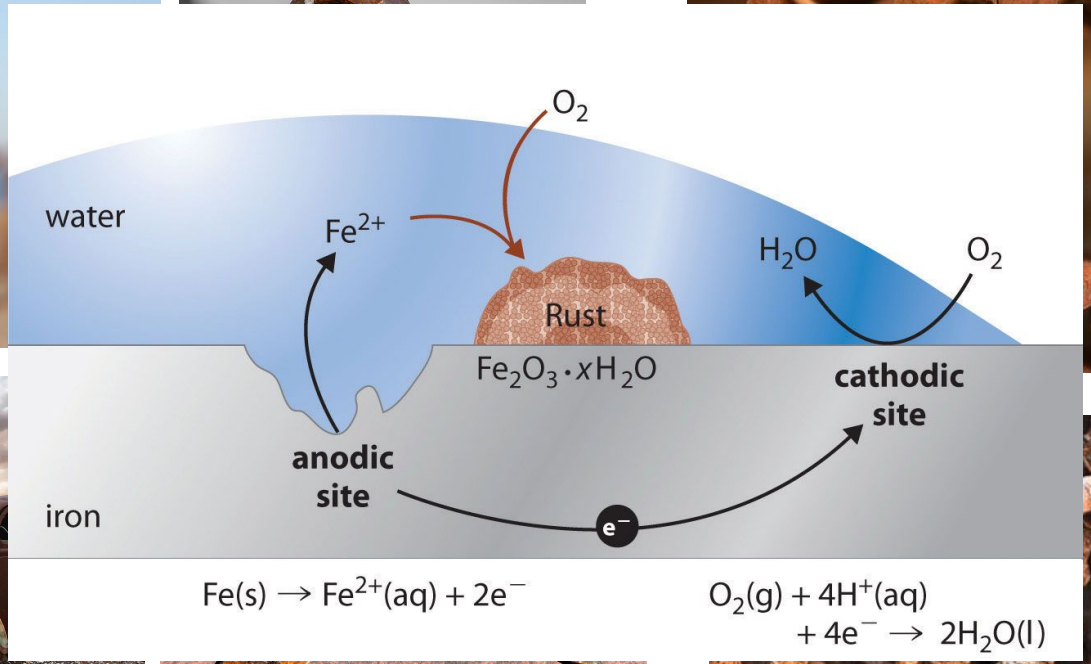
So... What is Rust?

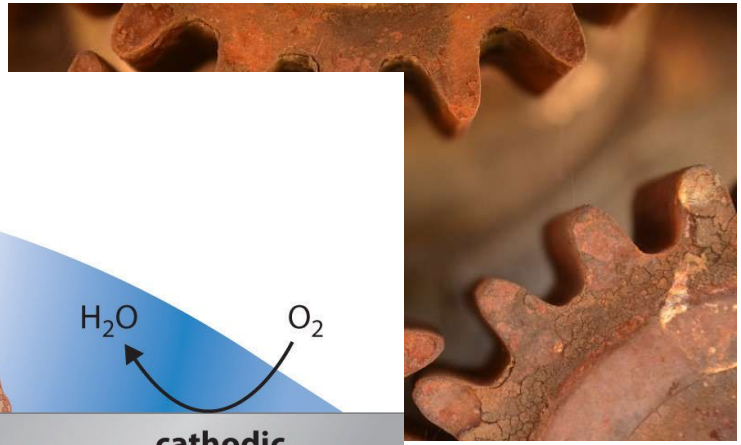


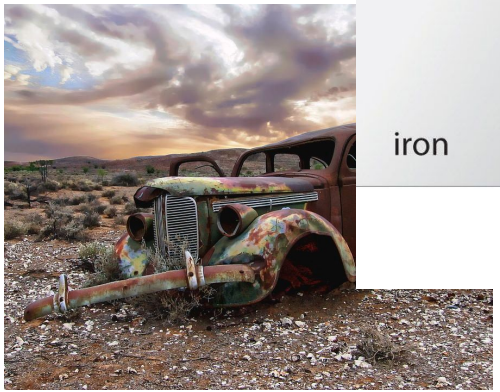
So... What is Rust?

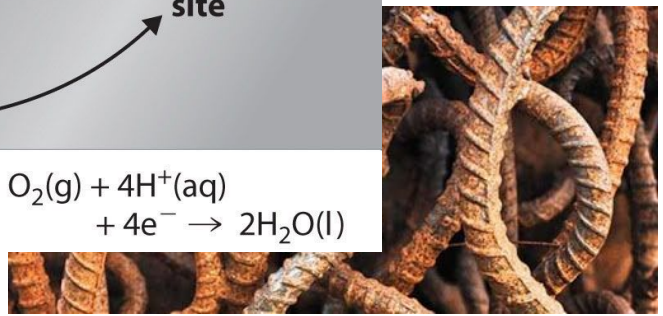


So... What is Rust?





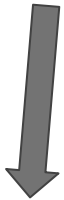




So... What is  ust?

So... What is Rust?

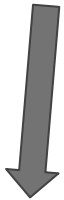
- Rust is a **multi-paradigm system programming language**



Programming Paradigms: Categorizing programming languages based on their features

So... What is Rust?

- Rust is a **multi-paradigm system programming language**

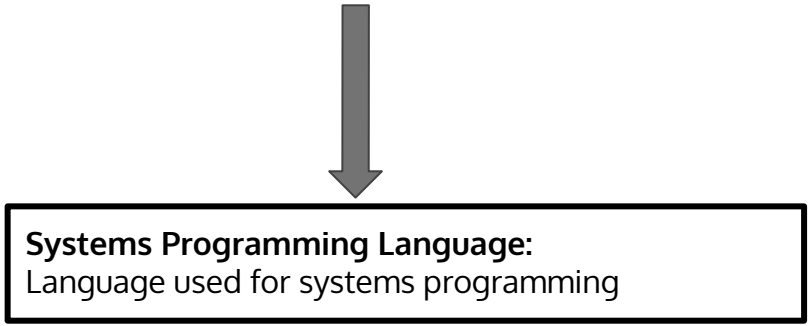


Programming Paradigms:
Categorizing programming languages based on their features

Multi-Paradigm:
Languages that support several programming paradigms and don't restrict the programmer to a single paradigm

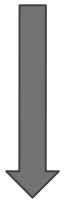
So... What is Rust?

- Rust is a **multi-paradigm system programming language**



So... What is Rust?

- Rust is a **multi-paradigm system programming language**

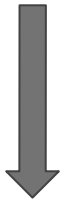


Systems Programming Language:
Language used for systems programming

Systems programming is **used for system software**. System software provides a platform for other software.

So... What is Rust?

- Rust is a **multi-paradigm system programming language**



Systems Programming Language:
Language used for systems programming

Systems programming is **used for system software**. System software provides a platform for other software.

System programming languages are designed to provide easy **access to the underlying hardware** and for **performance**

So... What is Rust?

- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**

What even is Safety?

> Safe Rust is the true Rust programming language. If all you do is write Safe Rust, **you will never have to worry about type-safety or memory-safety**. You will never endure a dangling pointer, a use-after-free, or any other kind of Undefined Behavior.

The Rustonomicon, Chapter 1

So... What is Rust?

- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**

What even is Safety?

> Safe Rust is the true Rust programming language. If all you do is write Safe Rust, **you will never have to worry about type-safety or memory-safety**. You will never endure a dangling pointer, a use-after-free, or any other kind of Undefined Behavior.

The Rustonomicon, Chapter 1

So... What is Rust?

- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**

What even is Safety?

> Safe Rust is the true Rust programming language. If all you do is write Safe Rust, **you will never have to worry about type-safety or memory-safety**. You will never endure a dangling pointer, a use-after-free, or any other kind of Undefined Behavior.

The Rustonomicon, Chapter 1

So... What is Rust?

- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**
- A key feature is **safe concurrency**

So... What is Rust?

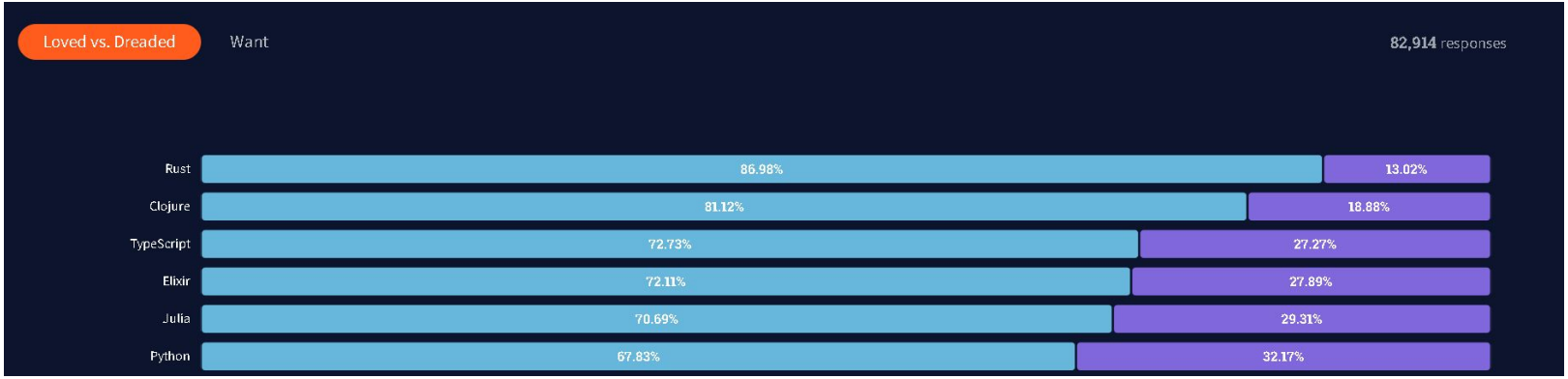
- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**
- A key feature is **safe concurrency**
- Syntactically, Rust is similar to C++
-

So... What is Rust?

- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**
- A key feature is **safe concurrency**
- Syntactically, Rust is similar to C++
- Rust is really great

So... What is Rust?

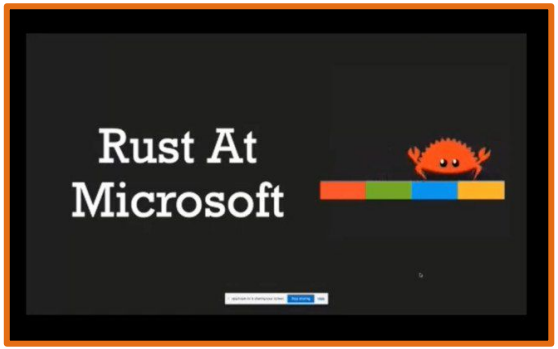
- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**
- A key feature is **safe concurrency**
- Syntactically, Rust is similar to C++
- Rust is really great



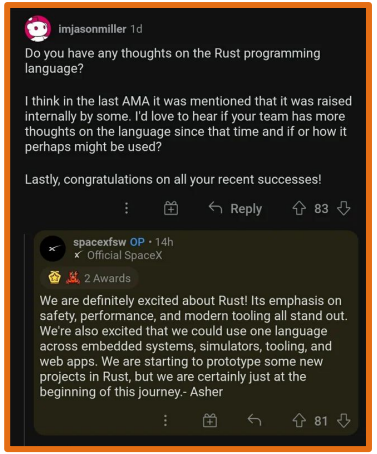
StackOverflow 2021 Developer Survey - Rust is the most loved language

So... What is Rust?

- Rust is a **multi-paradigm system programming language**
- Rust focuses on **safety**
- A key feature is **safe concurrency**
- Syntactically, Rust is similar to C++
- Rust is really great



Microsoft: Rust Is the Industry's 'Best Chance' at Safe Systems Programming



How will we learn Rust?

- We will often use the Rust textbook
  - Google “Rust Textbook”, it’s free online
  - https://doc.rust-lang.org/book/

How will we learn Rust?

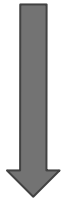
- Rust is already installed in your CS128 virtual machine
- Please test this whenever possible with

`cargo -V`

```
vagrant@0d603b65c801:~$ cargo -V
cargo 1.57.0 (b2e52d7ca 2021-10-21)
vagrant@0d603b65c801:~$
```

Cargo?

- Cargo is Rust's **build system** and **package manager**

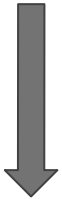


Build Systems:
Cargo **compiles*** your Rust project so that it can be executed by your computer

* Technically rustc is Rust's Compiler. Cargo acts as an abstraction layer that manages packages, handle metadata, and invokes rustc for you

Cargo?

- Cargo is Rust's **build system** and **package manager**



Package Manager:
Modern projects depend on external packages and libraries. A package manager handles those packages (and the packages that they might depend on)

Cargo?

- Cargo is Rust's **build system** and **package manager**
- How will we (primarily) use Cargo?
  - Create, Compile, and Execute your code
  - Download and manage packages you use (final project!)

Your First Rust Project - Hello, CS 128 Honors!

Lets walk through the process of creating and executing some Rust code from scratch.

1. Create a new project
2. Edit your code
3. Compile
4. Run

Your First Rust Project - Hello, CS 128 Honors!

Lets walk through the process of creating and executing some Rust code from scratch.

1. Create a new project
2. Edit your code*
3. Compile
4. Run

*You should use whatever editor you want, but we recommend VSCode with the rust-analyzer extension

More about Cargo

Let's learn some of the basic Cargo (build) functionality

- `cargo check`
- `cargo build`
- `cargo run`
- `cargo clean`

- `cargo test`
- `cargo bench`

More about Cargo

Let's learn some of the basic Cargo (package) functionality

- `cargo new`
- `cargo install`