Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Memory Safe Programming with Rust

Class Duration

- 2 days of comprehensive training
- Focus on the most challenging Rust concepts

Target Audience

- This course assumes prior experience with Rust programming language. The course focuses on upskilling Rust developers to better understand and apply Rust's memory safety concepts.

Description

The Rust Memory Safe Programming course focuses on the most difficult concept for new Rust developers to learn - memory safety. Programmers coming from other languages are able to pick up most Rust features very quickly and easily. However, learning and applying ownership and borrowing rules is one of the most difficult concepts to master. This course is designed to help developers understand the Rust memory model and how to apply it to their code.

Objectives

- Clarify Rust's Philosphy of Memory Safety
- Learn about Memory Leaks and Dangling Pointers
- Review Concepts such as the Stack and Heap
- Deepen Understanding of Ownership and Borrowing
- Understand References and Mutability
- Utilize Smart Pointers
- Learn how to Build Self Referential Structures

Training Materials

All students receive comprehensive courseware covering all topics in the course. Courseware is distributed via GitHub in the form of documentation and extensive code samples. Students practice the topics covered through challenging hands-on lab exercises.

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install Rust and Visual Studio Code on their computers. Also, students will need permission to install Rust Crates and Visual Studio Extensions. If students are unable configure a local development environment, a cloud-based environment can be provided.

Training Topics

Introduction

- How Memory is Managed on a Computer
- How the Operating System Views Memory
- How Memory is Allocated in a Process
- How Programming Languages and Runtimes Manage Memory
- Pitfalls with manual memory management in languages like C or C++
- Pitfalls with garbage collection in languages like Python or C#
- Memory Leaks and Dangling Pointers
- Ensuring Memory Safety at Compile Time
- Rust’s approach to Safe Memory Management
- Other Safe Memory Management Approaches

Memory Management

- Variables and their Data
- Variable Addresses and Data Addresses
- Mutability of Variables and their Data
- Variable and Data Ownership
- Rust’s Approach to Variables and their Data

Rust Memory Model

- Ownership and Borrowing
- References and Mutability
- Stack Allocation vs Heap Allocation
- Smart Pointers
- Thread Safety through Atomics and Locks
- Unsafe Rust

Smart Pointers

- What are Smart Pointers?

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

- When to use Smart Pointers?
- Unknown Size at Compile Time
- Self-Referential Structures
- Interior Mutability

Smart Pointer Types

- Box
- Rc, Weak, and Arc
- Cell and RefCell
- RwLock and Mutex

Conclusion

- Course wrap-up and next steps

Copyright 2025 Cloud Contraptions LLC – All Rights Reserved