This page is for Book title and published information

This page is about the author

Table of contents

### \*\*Table of Contents\*\*

1. \*\*Introduction to Rust\*\*

- 1.1 What is Rust?

- 1.2 Why Learn Rust?

- 1.3 Setting Up the Environment

2. \*\*Basic Syntax and Concepts\*\*

- 2.1 Variables and Mutability

- 2.2 Data Types (Primitive and User-defined)

- 2.3 Operators and Expressions

3. \*\*Control Flow\*\*

- 3.1 Conditional Statements (if, else)

- 3.2 Loops (for, while)

- 3.3 Function Calls

4. \*\*Functions and Ownership\*\*

- 4.1 Defining Functions

- 4.2 Ownership and Borrowing

- 4.3 Lifetimes

5. \*\*Structs and Enums\*\*

- 5.1 Creating Structures

- 5.2 Using Enums for Multiple States

- 5.3 Implementing Methods on Types

6. \*\*Error Handling\*\*

- 6.1 Result and Option Enums

- 6.2 Propagating Errors

- 6.3 Custom Error Types

7. \*\*Testing in Rust\*\*

- 7.1 Unit Tests

- 7.2 Integration Tests

- 7.3 Property-based Testing

8. \*\*Concurrency with Channels\*\*

- 8.1 Sending and Receiving Messages

- 8.2 Thread Safety

- 8.3 Communicating Between Threads

9. \*\*Advanced Topics\*\*

- 9.1 Interior Mutability (RefCell)

- 9.2 Async/await for Asynchronous Code

- 9.3 Memory Safety and Zero-cost Abstractions

10. \*\*Common Libraries and Frameworks\*\*

- 10.1 Using External Packages with Cargo

- 10.2 Web Development (e.g., Actix-web)

- 10.3 Command-line Tools (e.g., clap)

11. \*\*Best Practices\*\*

- 11.1 Writing Idiomatic Rust Code

- 11.2 Debugging Techniques

- 11.3 Performance Optimization

12. \*\*Conclusion and Next Steps\*\*

- 12.1 Resources for Further Learning

- 12.2 Contributing to the Rust Community

- 12.3 Future of Rust

### \*\*Appendices\*\*

- A. Glossary of Rust Terms

- B. Quick Reference Card

- C. Index