



RUST

OWNERSHIP · BORROWING · LIFETIMES · TRAITS · GENERICS · ENUMS · CARGO · ERRORS · ASYNC · O

DURATION	LEVEL	MODULES	MODE
45 Hours	Advanced	8	Classroom · Online · Hybrid

01 COURSE OVERVIEW

A Rust course built around ownership, borrowing and lifetimes, then applied to real programs. The course spends real time on the borrow checker, because everything else follows from it. Traits, generics, enums and pattern matching come next, then error handling, collections, testing, unsafe boundaries and async.

Learners build a CLI tool and a small concurrent service.

WHO IT IS FOR

- Systems and backend developers
- C and C++ developers
- Performance-focused engineers
- Developers interested in WebAssembly

PREREQUISITES

Solid programming experience in a compiled or systems language.

02 LEARNING OUTCOMES

01 Explain ownership, moves and borrowing rules.

03 Design with traits, generics and enums.

05 Use collections and iterators idiomatically.

07 Reason about when unsafe is justified.

02 Read and satisfy lifetime requirements.

04 Handle errors with Result and the ? operator.

06 Write tests, benchmarks and documentation.

08 Write async code with tokio.

03 MODULE STRUCTURE

8 MODULES

01 RUST BASICS

Cargo

Types

Functions

Variables and mutability

Control flow

Modules

LAB

FIRST PROGRAMS

02 OWNERSHIP

Move semantics

Borrowing rules

Slices

Clone and Copy

References

Dangling prevention

LAB

BORROW CHECKER DRILLS

03 LIFETIMES

Lifetime annotations

Structs with references

Common errors and fixes

Elision rules

Static lifetime

04 TRAITS AND GENERICS

Trait definitions

Trait bounds

Generic functions

Implementations

Associated types

Dynamic dispatch

05 ENUMS AND PATTERN MATCHING

Enums with data

match

Exhaustiveness

Option and Result

if let and while let

Newtype pattern

06 ERROR HANDLING AND COLLECTIONS

Result propagation

Vec, HashMap, BTreeMap

Closures

thiserror and anyhow

Iterators and adaptors

LAB

CLI TOOL

07 CONCURRENCY AND ASYNC

Threads

Arc and Mutex

async/await

Send and Sync

Channels

tokio runtime

LAB

CONCURRENT SERVICE

08 TESTING AND ECOSYSTEM

Unit and integration tests

Benchmarks

Crates

Doc tests

Clippy and rustfmt

WebAssembly overview

PROJECT

RUST SERVICE OR TOOL

04 ASSESSMENT & CERTIFICATION

Module assignments and labs	25%
Mini projects	20%
Internal assessments	15%
Final project and review	40%

CERTIFICATION

Learners who complete all modules, submit the final project and clear the review receive a course completion certificate from Kaizen Infinities Private Limited. Project work is documented for the learner's portfolio, and interview preparation is included in the closing sessions.

05 CAREER OUTCOMES

ROLES THIS PROGRAMME PREPARES FOR

RUST DEVELOPER

SYSTEMS ENGINEER

BACKEND ENGINEER

BLOCKCHAIN DEVELOPER

PERFORMANCE ENGINEER

CALL US

+91 73051 33996

EMAIL US

apply@kaizeninfinities.com

VISIT OUR WEBSITE

www.kaizeninfinities.com