Rust Programming Training Certification Course Outline

Module 1: Introduction to Rust

- First, understand what Rust is and why developers prefer it.
- Next, install Rust using rustup and set environment.
- Then, run your first program using cargo build.
- Finally, read basic syntax and project structure clearly.

Module 2: Variables and Data Types

- First, declare variables using let keyword simply.
- Next, use mut to change values safely.
- Then, explore integers, floats, booleans, and characters.
- Finally, practice small examples to strengthen understanding.

Module 3: Control Flow

- First, use if and else for decisions.
- Next, apply match for multiple pattern checks.
- Then, create loops using loop, while, and for.
- Finally, combine conditions to control program flow clearly.

Module 4: Functions and Modules

- First, define functions using fn keyword properly.
- Next, pass parameters and return values correctly.
- Then, organize code into modules for clarity.
- Finally, reuse functions to keep programs clean.

Module 5: Ownership and Borrowing

- First, understand ownership rules for memory safety.
- Next, move values between variables carefully.
- Then, borrow data using references safely.
- Finally, avoid dangling pointers through Rust checks.

Module 6: Structs and Enums

- First, create structs to store related data.
- Next, define enums for multiple possible states.
- Then, implement methods using impl blocks.
- Finally, model real problems using custom types.

Module 7: Collections

- First, use vectors to store dynamic lists.
- Next, apply strings for text handling.
- Then, use hash maps for key value storage.
- Finally, iterate collections using loops efficiently.

Module 8: Error Handling

- First, handle errors using Result type properly.
- Next, use Option for possible missing values.
- Then, apply match to manage failures gracefully.
- Finally, write safe code without unexpected crashes.

Module 9: File Handling and I/O

- First, open files using Rust standard library.
- Next, read and write data securely.
- Then, process user input from terminal.
- Finally, handle errors during file operations carefully.

Module 10: Concurrency

- First, create threads to run tasks parallelly.
- Next, share data using channels safely.
- Then, use mutex for controlled access.
- Finally, build fast programs with safe concurrency.

Module 11: Building Projects with Cargo

- First, manage projects using cargo commands daily.
- Next, add dependencies through Cargo.toml easily.
- Then, test code using cargo test regularly.
- Finally, build and release optimized applications professionally.

Module 12: Certification and Final Assessment

- First, review all Rust concepts and practice exercises.

- Next, complete a real world coding project.
- Then, pass the final assessment confidently.
- Finally, receive your Rust Programming Certification successfully.

+92-308-5145-822 www.technocation.pk info@technocation.pk