Rust Programming Fundamentals

Course Summary Table

Duration:	3 Days
Target Audience:	C/C++ developers, C#/Java developers
Objectives:	• Understand the Rust language fundamentals • Write idiomatic Rust code • Use Rust types effectively • Understand Rust ownership and lifetime • Use modules to organize code
Pre Requisites:	• Good knowledge of C and/or C++ languages • Alternatively, knowledge of C# or Java is also acceptable

Instructor: **Pavel Yosifovich**

Abstract

The Rust programming language promises to be safe, fast and productive. Created by Mozilla, Rust provides high level features while maintaining control and safety for low level code if required. Rust plays in the same playing field as C/C++.
This course introduces the fundamentals of Rust, from the basics up to using types, modules, generics, pattern matching, error handling and more. Lab exercises are used to help sink in the theoretical concepts.

Syllabus

- Module 1: Introduction to Rust
  - Why Rust?
  - Hello, Rust!
  - Tools
  - The Rust compiler
  - Cargo
  - Testing
  - Summary

- Module 2: Variables and Data Types
  - Variables
  - Mutability
  - Fundamental Data Types
  - Tuples
  - Arrays
  - Functions
  - Control Flow
  - Summary

- Module 3: Ownership

    - Why ownership?
    - References
    - Borrowing
    - Slicing
    - Summary

- Module 4: Compound Types
    - Structs
    - Creating Objects
    - Implementing traits
    - Enums
    - Methods
    - Summary

- Module 5: Pattern Matching
    - The power of Pattern Matching
    - The match keyword
    - Catch all match
    - Summary

- Module 6: Managing Projects
    - Packages and Crates
    - Using crates.io
    - Modules
    - Visibility and Scope
    - Modules and Files
    - Summary

- Module 7: Common Types
    - Strings
    - Vectors
    - Hash maps
    - Summary

- Module 8: Error Handling
    - Types of Errors
    - Panicking
    - Using the Result type
    - Unwrap and Expect
    - Propagating Errors
    - When to panic
    - Summary

- Module 9: Generics and Traits
    - Generic Data Types
    - Traits
    - Implementing traits
    - Common traits in the standard library
    - Polymorphism with traits

    - Summary

- Module 10: Advanced Topics
    - Threading
    - Synchronization
    - Interoperability
    - Interop with C and C#
    - Summary