

# Course : Rust Advanced

**Building High-Performance, Secure Applications with Rust**

**Practical course - 3d - 21h00 - Ref. RUY**

**Price : 1650 € E.T.**

NEW

Deepen your mastery of Rust by leveraging its advanced memory management mechanisms (complex ownership, nested lifetimes) and mastering the language's idiomatic patterns. Develop high-performance, secure applications by implementing concurrent programming without data races, low-level optimization, and advanced use of traits, generics, and zero-cost abstractions.

## Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Use the features that allow operator overloading
- ✓ Mastering Traits and Credits
- ✓ Designing and Writing Asynchronous and Multithreaded Programs
- ✓ Interfacing with Python, C, and C++

## Intended audience

Developers, project managers, architects.

## Prerequisites

Practical experience with Rust or completion of the RUX course.

## Practical details

### Hands-on work

Demonstrations and hands-on exercises will help participants apply the concepts covered.

### Teaching methods

Each new theoretical concept will be immediately applied in a practical way.

## Course schedule

### PARTICIPANTS

Developers, project managers, architects.

### PREREQUISITES

Practical experience with Rust or completion of the RUX course.

### TRAINER QUALIFICATIONS

The experts leading the training are specialists in the covered subjects. They have been approved by our instructional teams for both their professional knowledge and their teaching ability, for each course they teach. They have at least five to ten years of experience in their field and hold (or have held) decision-making positions in companies.

### ASSESSMENT TERMS

The trainer evaluates each participant's academic progress throughout the training using multiple choice, scenarios, hands-on work and more.

Participants also complete a placement test before and after the course to measure the skills they've developed.

### TEACHING AIDS AND TECHNICAL RESOURCES

- The main teaching aids and instructional methods used in the training are audiovisual aids, documentation and course material, hands-on application exercises and corrected exercises for practical training courses, case studies and coverage of real cases for training seminars.
- At the end of each course or seminar, ORSYS provides participants with a course evaluation questionnaire that is analysed by our instructional teams.
- A check-in sheet for each half-day of attendance is provided at the end of the training, along with a course completion certificate if the trainee attended the entire session.

## 1 A Review of Generics, Traits, and Standard Traits in Rust

- Reminders about moving, borrowing, and returning.
- String Formatting in Rust.
- The Orphan Rule and "Coverage Implementation."
- Implement the features that enable operator overloading.
- Design using features related to references, such as AsRef, Deref, and Borrow.
- The different types of closures: Fn, FnMut, FnOnce.

### Hands-on work

Implementation of standard features.

#### TERMS AND DEADLINES

Registration must be completed 24 hours before the start of the training.

#### ACCESSIBILITY FOR PEOPLE WITH DISABILITIES

Do you need special accessibility accommodations? Contact Mrs. Fosse, Disability Manager, at psh-accueil@orsys.fr to review your request and its feasibility.

## 2 Rust Project Structure

- Manage multiple projects using workspaces.
- Make features optional using features.
- Compilation profiles for performance, debugging...

### Demonstration

## 3 Rust macros for automating code generation

- Declarative macros.
- The different types of procedural macros.

### Hands-on work

Writing a macro.

## 4 Asynchrony in Rust

- The `async` and `await` keywords to retrieve a result from the "Future"
- Survey and wake-up call.
- "Pin" and "Unpin" features for self-referencing objects.

### Hands-on work

Implementation of asynchronous code.

## 5 Multithreading with Rust

- Starting a Thread and Safe Concurrency
- Rust Locks: Mutex and RwLock
- Exchanging Data Between Threads Using Channel<T>
- Send and Sync features for passing objects to threads
- Sending an inter-thread signal using Condvar and simple modifications with Atomic
- Modify a global variable

### Hands-on work

Multithreaded code synchronization using a lock

## 6 Error Handling

- Reminders on Error Handling and Error Propagation
- Best Practices for Handling Errors
- Handling an Error with &dyn or Box<dyn Error>

### Hands-on work

## 7 Integrating Rust with C/C++ and Python

- Calling Rust Code from C/C++
- Calling C/C++ code from Rust: unsafe code
- Calling Rust Code from Python

### Hands-on work

*This programme is an original creation, developed by the teaching teams at ORSYS Formation. Any reproduction, representation, adaptation or use, in whole or in part, without the prior written authorisation of ORSYS, is strictly prohibited. ORSYS reserves the right to take any action necessary to protect its intellectual property rights.*

## Dates and locations

### REMOTE CLASS

2026 : 1 Dec.

2027 : 31 Mar., 30 June, 29 Sep., 15 Dec.

### PARIS LA DÉFENSE

2027 : 31 Mar., 30 June, 29 Sep., 15 Dec.