Publication date : 04/05/2024

Course : Optimizing .NET applications in C# language

Practical course - 4d - 28h00 - Ref. OPA
Price : 2470 CHF E.T.

4,1 / 5

This course provides you with a methodology for improving the efficiency of your .NET and .Net Core applications. It will teach you how to use the main performance analysis and diagnostic tools, and will enable you to master the various optimization techniques for coding in the C# language.

Teaching objectives

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

- Mastering .Net application architecture
- Diagnose memory allocations and avoid memory leaks
- Optimize C# code with reference to the latest versions of C# 7, 8 and 9
- Master operations on arrays and collections
- Optimize and measure performance gains through parallelism
- Coding in C# to solve real-life problems

Intended audience

Developers, engineers, architects, project managers.

Prerequisites

Good knowledge of C#. Experience required.

Course schedule

1 Introduction

- Evolution of the .Net platform and its ecosystem.
- Optimize: what, how, when? Objectives.

Hands-on work

Analysis of the execution of a .Net application using the WinDebug tool.

PARTICIPANTS

Developers, engineers, architects, project managers.

PREREQUISITES

Good knowledge of C#. Experience required.

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.

2 GC crumb tray, allocation optimization and diagnostics

- Memory management using the garbage collector and API GC.
- Diagnostic tools from Visual Studio, JetBrains and BenchmarkDotNet.
- The Dispose pattern and the implementation of IDisposable.
- Recommendations for avoiding potential memory leaks in a .Net application.
- Design time safeguards via Code Analysis Rules.
- .Net ecosystem tools for tracing boxing, closure and null reference errors.

Hands-on work

Detecting and correcting memory leaks. Analysis rules and diagnostic tools.

3 C# methods and language constructs

- C# methods and the evolution of coding.
- Extension methods and LINQ architecture.
- Asynchronous methods and use of async/await keywords.
- Performance measures for Value allocations compared with Reference allocations.
- New ValueTuple value type in C# 7.0.
- Parameter passing and return by reference in C# 7.2.
- Advantages of functional programming in C# and comparison with the F# functional language.
- Pattern Matching in C#.

Hands-on work

Exercises and hands-on demonstrations of coding elements and functional programming.

4 Paintings and collections

- Performance of data operations.
- .Net collections and features.
- Implementation of the IEquatable interface.
- IntPtr pointer and the C# 7.2 stackalloc keyword to allocate on the stack.

Hands-on work

Exercises and demonstrations on collections and allocations on the stack.

5 The .Net 4.7 TPL library: asynchronism and parallelism

- Evolution of asynchronous call management via async/await keywords.
- The new System.Threading.Tasks classes.
- Parallelizing for and foreach iterations. Design pattern for parallelism.
- Using the PLinq infrastructure.
- Performance diagnostics via NuGet BenchmarkDotNet and JetBrains DotTrace.

Hands-on work

Exercises and demonstrations on parallelism. BenchmarkDotNet diagnostics.

6 Conclusion

- Impact of C# 8 and C# 9 features on .Net code and applications.
- A summary and a few tips.
- Book recommendations and Internet references.

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.

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.

Dates and locations

REMOTE CLASS
2026 : 24 Mar., 16 June, 29 Sep., 8 Dec.