+33 (0)1 48 10 09 46
formation@ambient-it.net 140 Avenue Jean Lolive 93500 Pantin - France

Updated 12/30/2024

Sign up

C# 12 training with .NET 8 and Visual Studio 2022

RAPID UPGRADING FOR ALL DEVELOPERS

5 days (35 hours)

Presentation

C# is undeniably one of the most comprehensive programming languages available to engineers today, coupled with the power of .NET 8 . This training course has been specially designed to bring all developers up to speed quickly on the fundamentals as well as the new features of the .NET 8 Framework with Visual Studio 2022 and the C# language in version 12.

In this multi-purpose course, designed as a veritable "Swiss army knife" for .NET, you'll learn the main concepts of C# to help you become more autonomous on the Microsoft .NET 8 platform**. This refresher , aimed at all programmers, will give you all the solid foundations you need to become autonomous in your future developments.** The course is designed to be fun, educational and interactive.

First, we'll review all the basic concepts of OOP through the modern language of C#, with examples. We'll then cover all the subtleties of .NET and its new features over the last few years, from C# 8 to C#12 and beyond.

We'll cover the proper use of types and delegates, as well as case studies in concurrent and asynchronous programming, so that you can answer even the most advanced technical questions. Finally, we'll take a look at the latest Web Front Frameworks
: Blazor, exposing our own Web APIs and effectively mastering data persistence.

As always, we'll be teaching you the latest version of the tool, i.e. C# 13 with .NET 8 Core Version with Visual Studio 2022 released on November 8, 2021.

Objectives

- Master the main concepts of object-oriented programming in C#
- Discover the latest developments in C# and .NET through Visual Studio 2022
- Acquire autonomy in developing, debugging and understanding the .NET documentation
- Learn the fundamentals of the new Blazor Web Framework and .Net 8 applications
- Build your Web API via ASP.NET Core and easily access a database
- Implementing tests
- Initiate a clean code approach

Target audience

- Developers from any programming language
- Technical project managers

Prerequisites

- Working knowledge of a recent programming language
- Basic knowledge of the .NET Framework is a plus
- Test My Knowledge

SOFTWARE REQUIREMENTS

- Visual Studio Community installed (to create modern applications for Android, iOS and Windows, as well as web applications and cloud services)
- A Microsoft account (to access Visual Studio)
- Install at least the SDK for .Net 8

RECOMMENDATIONS FOR PRE- AND POST-COURSE READING

- I recommend the following blogs:
  - Thomas Levesque's blog C# tips and best practices.
  - Steve Smith's Ardalis blog features articles software development, coding practices and Microsoft technologies, including C#.
- The official C# documentation: contains detailed information on the C# language and its use. its features.
- Videos from the Microsoft Developer YouTube channel: featuring video tutorials developers, including videos on C# and Visual Studio.
- C# developer forums: C# developer forums can be a useful way of obtaining help and advice on C# programming, such as that provided by Developpez.net.

C# 12 with Visual Studio 2022 training program

Introduction: What is .NET in 2024?

- Who is .NET designed for?
- The main developments in the Framework since its creation
- The future of .NET
- Architecture
  - Managed Code
  - The IL / CLR / Jit
- Interpreted language VS Compiled language
- Visual Studio Versus IDE Visual Studio Code Versus Rider editor
- Don't confuse IDE vs Language vs .NET Framework
- Developments and main additions to the C# language
- The pace of LTS & Current releases and their respective support by Microsoft
- .NET 8 :
  - .NET Aspire
  - Improvements made
  - Native AOA support
  - Globalization
  - Containers
  - NuGet
  - AOT compilation

Quick overview of the .NET ecosystem

- Core Versus Mainstream Versus Standard
- Table of language versions by framework version
- The various specialized Frameworks :
  - WPF
  - Entity
- MSDN documentation
- GitHub
- Git + short introduction

Visual Studio basics: the fundamentals

- Solution
- Projet / Shared Project
- Assembly
- DLL
- Nuget Package
- Adding a project to an existing solution
- Reference to a project from another project
- Referencing external libraries
- Namespaces
- Keyboard shortcut

C# basics: the fundamentals

- Paradigm & Typing
- Primitive types
- Class members
- Encapsulation and visibility

- Structures vs Classes vs Record
- Tuples
- Usings
- Builders and destroyers
- Generic types
- Operators and expressions
- Differentiate between 2 universes: static and instance
- Differentiate between the 2 worlds: managed and unmanaged

Debug & Breakpoint

- Debugging an application under development
  - Setting a stop point
  - Set a condition for a breakpoint
- Variable modification in real time
- Recording during development and execution
  - Instrumentation with debugging and tracing
    - Write to default trace listener
    - Configuring trace listeners

Introduction to object-oriented programming: OOP fundamentals in C#

- Object-oriented programming: it's basically simple!
- Procedural vs. OOP
- The main founding principles
  - Encapsulation
  - Heritage
  - Polymorphism

Introduction to good developer practices

- Naming convention and programming rules
- Liability
- Factoring
- SOLID design principles
- Dry/Yagni/Kiss/Wet/....
- Intention / Naming
- Validation cycle

Basic C# coding

- Local variables
- Static / const
- Statements and expressions
- Declarations
- Expressions
- Booleans

- Comments
- Pre-treatment guidelines
- Compilation symbols
- Type value and reference

Data structures and writing optimized code in C#

- LIFO
- FIFO
- Choosing the right data structure to optimize performance
  - Tables
  - Lists
  - Batteries
  - Waiting tail
  - Dictionaries, hash tables and hashesets
  - Generic lists
- Best practices in writing optimized code in C#
  - Boxing and unboxing
  - Concatenation of character strings
  - Exception handling
  - Instructions while, for & foreach
  - Delegates

C# classics

- Using and implementing abstract / virtual / Partial classes
- Using and implementing interfaces
- Differences between an abstract class and an interface: when to use one rather than the other?
- Mutable vs Immutable
- Generics
- Type nullable
- Dynamic type
- Operator overload: defining equality
- Delegates
- Try / Catch / Finally
- Serialization

Introduction to design patterns

- The different types of design pattern / GoF
  - Creation
  - Structural
  - Behavioral
- Factory implementation
- Decorator implementation
- Observer implementation

Introduction to testing

- Different types of test
  - Pyramid
  - Quadrant
- How to test and why?
- Mock...
- TDD / BDD

Features from C# 8 to C# 11

- Most useful by version

THE FUTURE: NEW FEATURES IN C# 12

Introduction to concurrent and asynchronous programming

- Concurence and Threading
- Multi-threading use cases
- Thread and Task
- Synchronization and communication
- ThreadPool and efficiency
- Deadlocks and other pitfalls to avoid

Introduction to asynchronous programming

- Difference between synchronous and asynchronous calls
- Async operations
- async and await
- Managing Progress and Abandonment
- CancellationToken

Reflection and Attributes

- Introspection of assemblies and classes
- Attribute types
- Attribute parameters
- Code generation: Emitters

.NET PROJECT 7/8

- Startup / program use
- IoC & Co
- Middleware
- Environment Variable + section
- Secrets

Web basics

- How the Web works
- HTTP/2 and HTTP/3
- Cookies
- The MVC model
- Model validation
- Different architectures
  - N-Tiers
  - Onion Aka Clean Archi Aka Hexagonal Aka Port/adapter
  - Vertical Slice

The Web with Blazor (WebAssembly)

- Blazor Server vs Blazor client (WASM)
- NavBar HTML CSS
- Model
- Page
- HotReload
- CRUD

Web API / Minimal API

- Design your first access API
- Controller
- Consuming the API from the client

Persistence

- Using DbContext
- Automatic database generation via Code-First
- CRUD
- Introduction to LINQ queries
- Migration

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new advanced computer technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning on entry to training complies with Qualiopi quality criteria. As soon as enrolment is finalized, the learner receives a self-assessment questionnaire enabling us to

assess their estimated level of proficiency in different types of technology, and their expectations and personal objectives for the training to come, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) that could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical course: 60% Practical, 40% Theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Sanction

A certificate will be issued to each trainee who completes the course.