



To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Modern C# Development with .NET

Class Duration

35 hours of live training delivered over 5 days.

Student Prerequisites

- Professional programming experience in at least one programming language
- Familiarity with Git for everyday version control
- No prior C# or .NET experience required

Target Audience

Professional developers adopting C# as a new language, whether they are coming from Java, Python, JavaScript, or any other background, and .NET Framework developers modernizing their skills for current .NET. Equally valuable for backend and full-stack engineers joining .NET teams and for technical leads who need a current, accurate picture of the platform. Graduates leave ready for framework-specific training: teams focused on AI-augmented workflows can continue to [AI-Assisted Development with C#](#), and desktop teams to [WPF with MVVM and C#](#).

Description

This comprehensive course builds fluency in modern professional C#: the C# 14 language on .NET 10 as it is actually written on production teams today. Participants first master the language's modern core, records, pattern matching, nullable reference types, generics, LINQ, and async/await, learning not just the syntax but the design sensibility behind idiomatic modern C#. They then apply that fluency to the workhorse patterns of production .NET: dependency injection, configuration, and logging with the Generic Host, testing with xUnit, and data access with Entity Framework Core 10. Code is written alongside an agentic AI coding assistant throughout, as modern .NET teams do, so participants learn to prompt for idiomatic C# and to review generated code with a critical eye. Every pattern is taught in the business context where it is actually used, from configuration-driven services to data-backed applications, and the course closes by assembling the pieces into production-shaped code, with particular attention to developers modernizing



To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

from older .NET Framework codebases. Suitable both for developers new to C# and for experienced .NET developers whose habits predate the modern platform.

Learning Outcomes

- Navigate the modern .NET platform: the SDK, the .NET CLI, NuGet, and solution structure in Visual Studio 2026 or VS Code.
- Model data and behavior with classes, structs, interfaces, and records, choosing the right construct for each design.
- Replace branching conditional logic with clear, declarative pattern matching and switch expressions.
- Write null-safe C# with nullable reference types, annotations, and the full family of null-handling operators.
- Design reusable, type-safe code with generics, constraints, and the right collection type for each workload.
- Query and transform data fluently with lambdas and LINQ, and reason correctly about deferred execution.
- Write correct asynchronous code with async/await, task composition, cancellation, and asynchronous streams.
- Apply C# 12 through C# 14 features, from primary constructors and collection expressions to extension members and the field keyword, where they genuinely improve code.
- Build production services on the Generic Host with dependency injection, strongly typed configuration, and structured logging.
- Test C# code with xUnit unit and integration tests, and implement data access with Entity Framework Core 10.
- Pair productively with an agentic AI coding assistant, prompting for idiomatic modern C# and reviewing what it produces.
- Modernize habits and code from the .NET Framework era to current .NET idioms.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Software Requirements

Visual Studio 2026 (Community edition is sufficient) on Windows, or VS Code with the C# Dev Kit on any platform, plus the .NET 10 SDK and Git. Access to an agentic AI coding assistant is required; a free tier is sufficient.

Training Topics

The Modern .NET Landscape and Tooling

- The unified .NET 10 platform: runtimes, SDKs, and workloads
- The .NET CLI: projects, builds, and package management with NuGet
- Visual Studio 2026 and VS Code with the C# Dev Kit
- Solution anatomy: projects, references, and Roslyn analyzers
- Agentic AI coding assistants in the everyday .NET workflow

C# Language Core: Types and Members

- The unified type system: value types, reference types, and boxing
- Classes, structs, and interfaces: choosing the right shape
- Properties, indexers, and the field keyword
- Records and record structs for data-centric design
- Immutability, with expressions, and value equality
- File-scoped namespaces, access modifiers, and top-level statements

Pattern Matching

- Type, constant, and relational patterns
- Property and positional patterns with deconstruction
- Switch expressions vs. switch statements
- List patterns and slice patterns
- Combining patterns with and, or, and not

Nullable Reference Types and Null Safety

- Nullable value types vs. nullable reference types
- Enabling nullable context and reading the warnings
- Null-coalescing, null-conditional, and null-forgiving operators
- Null-conditional assignment in C# 14
- Designing APIs that make null explicit
- Adopting nullable annotations in existing code

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Generics and Collections

- Generic types, methods, and constraints
- Variance: covariance and contravariance in practice
- Choosing collections: List, Dictionary, HashSet, and friends
- Immutable and frozen collections
- Span, ReadOnlySpan, and first-class span conversions

Delegates, Lambdas, and LINQ

- Delegates, Func and Action, and events
- Lambda expressions and closures
- LINQ method syntax and query syntax
- Projection, filtering, ordering, grouping, and joins
- Deferred execution and multiple enumeration pitfalls
- LINQ style: readability, performance, and when not to use it

Async, Await, and Concurrency Basics

- The Task-based asynchronous pattern
- How async/await really works: continuations and contexts
- Composing tasks: WhenAll, WhenAny, and cancellation tokens
- Avoiding async void and sync-over-async traps
- Asynchronous streams with IEnumerableAsync
- A first look at parallelism: Parallel, PLINQ, and channels

Modern C# 12-14 in Practice

- Primary constructors and collection expressions
- Extension members: extension blocks beyond extension methods
- Lambda parameter modifiers and nameof with unbound generic types
- Partial constructors and partial events
- User-defined compound assignment operators

Dependency Injection, Configuration, and Logging

- The Generic Host and the application builder pattern
- Service lifetimes: singleton, scoped, and transient
- Configuration providers: appsettings, environment variables, and user secrets
- The options pattern for strongly typed settings
- Structured logging with ILogger and logging providers

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Testing .NET Code

- xUnit fundamentals: facts, theories, and fixtures
- Test structure, naming, and meaningful assertions
- Test doubles and mocking libraries
- Integration tests with the Generic Host and real dependencies
- Testing asynchronous code without flakiness

Data Access with Entity Framework Core 10

- DbContext, entities, and model configuration
- Migrations and schema evolution
- Querying with LINQ: translation, tracking, and projections
- Loading related data: eager, explicit, and lazy
- Saving changes: transactions and concurrency handling
- Knowing the boundaries: compiled queries, raw SQL, and Dapper

Production-Shaped Applications and Modernization

- Organizing solutions: layers, boundaries, and internal APIs
- Exceptions, error handling, and result-style returns
- From .NET Framework habits to modern .NET idioms
- Reviewing AI-generated C#: nullability, async correctness, and stale APIs
- Performance awareness: allocations, benchmarking, and profiling basics