

C# / .Net Core Developer Skill Journeys

Intermediate C# / .Net Core Application Development | Patterns, Generics, Threading, Database, Entity Framework, UI & More (Latest Edition) (TTCN2155)

Level-Up Your C# Skills! Create advanced, efficient and well -designed C# applications using advanced techniques

Course Snapshot

- **Course: Intermediate C# / .Net Core Application Development | Patterns, Generics, Threading, Database, Entity Framework, UI & More (TTCN2155)**
- **Duration:** 5 days
- **Audience & Skill Level:** Intermediate level for experienced C# .Net developers.
- **Format / Hands-on:** This course combines engaging instructor-led presentations and practical demonstrations with hands-on programming exercises, challenge labs, use case exploration and engaging group activities. Student machines are required.
- **Flexible Delivery Options:** This course can be delivered for your team or organization **online-live (virtual), onsite in-person, self-paced** or across our immersive **blended learning experience platform (LXP)**.
- **Public Schedule:** This course is currently available on our Public Open Enrollment Schedule.
- **Customizable:** We're flexible! This course agenda, topics, labs, hours and delivery modalities can be adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Overview

Intermediate C# / .Net Core Programming is a five-day lab-intensive program that enhances your basic knowledge with advanced constructs and industry best practices. Proficiency in C# is imperative for creating scalable, robust applications. This course enables you to refine your skills, using the latest C# features to develop high-quality software solutions. From employing raw literals for cleaner code to applying tuples for improved data structures, and managing resources effectively for globalized applications, the course is designed to strengthen your programming foundation for immediate application in a multitude of projects.

The curriculum is crafted to ensure a balanced mix of theoretical and practical learning, with 50% of the course dedicated to hands-on labs, mirroring real-world challenges. You will enhance your understanding of design patterns and SOLID principles, creating code that's as resilient as it is reliable. Engage in test-driven development with XUnit, and master generics to optimize your coding efforts. With an industry-expert instructor guiding you through each topic, you'll gain invaluable insights into effective problem-solving and practical application of your new skills.

You'll exit the course equipped with a solid understanding of intermediate C# principles, ready to be applied to your role professional work. You'll emerge with the capability to write more maintainable, efficient, and scalable code, seamlessly integrating advanced C# features into your daily development tasks.

Learning Objectives

Our engaging instructors and mentors are highly experienced practitioners who bring years of current "on-the-job" experience into every classroom.

Working in a hands-on learning environment, guided by our expert team, attendees will explore:

- **Proficient Use of Advanced C# Constructs:** You'll be able to integrate advanced C# features such as type specifiers, raw literals, and top-level statements into your daily coding practices to write more concise and readable code.
- **Application of Design Patterns and SOLID Principles:** You'll gain the ability to apply common design patterns and adhere to SOLID principles to create well-structured, maintainable, and scalable code that aligns with professional standards.
- **Development of Robust Classes and Exception Handling:** You'll develop robust classes utilizing properties, functions, delegates, and events, and will be adept at implementing comprehensive exception handling to ensure the resilience of

your applications.

- **Test-Driven Development with XUnit:** You'll learn to employ test-driven development methodologies, utilizing XUnit for creating and managing unit tests, ensuring the reliability and functionality of your code before deployment.
- **Effective Implementation of Generics and LINQ:** You'll learn to effectively use generics to minimize code redundancy and apply LINQ for sophisticated data querying and manipulation, leading to more efficient and powerful data handling within your applications.
- **Concurrency and Asynchronous Programming:** You'll master the concepts of threads, tasks, and the async/await pattern, enabling you to write asynchronous C# code that improves the performance and responsiveness of your applications.

Audience & Pre-Requisites

This is an **intermediate level** .Net programming course, designed for experienced C# developers. This course is NOT for new developers, developers new to C#, or those without a programming background.

Take Before: Students should have incoming practical skills aligned with those in the course(s) below, or should have attended the following course(s) as a pre-requisite:

- TTCN20483 [Programming in C# | Creating Apps in C# and .Net Core / Latest Edition \(20483\)](#)

Next Steps / Related Courses | Microsoft Developer Training Suite

Below is a subset of our follow on courses in .Net development, web development, SOLID design, .Net services, .Net secure coding and others. Please visit our site or contact us for the full course listing and next-step recommendations.

- TTCN10975 [Introduction to Programming in C# \(For C# and .Net Core / Latest Edition\) \(10975\)](#)
- TTCN20483 [Programming in C# | Creating Apps in C# and .Net Core / Latest Edition \(20483\)](#)
- TTCN2155 [Intermediate C# | Patterns, Generics, Threading, Database, Entity Framework, UI & More](#)
- TTCN20486 [Developing ASP.Net Core MVC Web Applications / Latest Edition \(20486\)](#)
- TT8320-N [.Net Secure Coding Camp | Attacking and Securing .Net Web Apps \(for .Net Core / Latest Edition\)](#)

Enhanced Learning Services: Please also ask about our robust Learning Experience Platform (LXP), Skills Assessment & Skills Prep Services, Skills Immersion Programs & Camps, Coaching and Mentoring Services and Extended Learning Support programs.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill level, interests and participation.

1. Overview of New Features

- Apply modern C# enhancements to streamline code development and maintainability.
- Type specifiers
- Raw Literals
- Top-level statements
- Records
- Global Usings
- Tuples
- Deconstruction

2. Design Guidelines

- Develop code that adheres to industry-standard design

guidelines for clarity, sustainability, and effectiveness.

- Conventions
- Class Design guidelines
- Exception Design guidelines
- Common Design Patterns

3. Robust Classes

- Construct robust classes with advanced property and function implementations for more dynamic and responsive code.
- Properties
- Functions
- Delegates

- Events

4. Unit Testing with XUnit

- Implement test-driven development with XUnit to ensure the creation of reliable, bug-resistant applications.
- Utilizing Test Driven Development (TDD)
- Survey of Unit test frameworks
- Using XUnit
- Mocking with Moq

5. Exceptions

- Master sophisticated exception handling techniques for more

- robust error management in applications.
 - Syntax
 - Exceptions vs Return code
 - Testing for exceptions
- 6. Generics**
- Utilize generics to create flexible, reusable, and type-safe code structures, reducing redundancy.
 - Generic Classes
 - Generic Functions
 - Understanding invariance
 - Survey of Generics in the .Net Library
- 7. Delegates**
- Leverage delegates to design more versatile and modular code with methods as first-class objects.
 - Methods as objects
 - Generics and Delegates
 - Delegates in the library
- 8. Events**
- Efficiently manage and respond to application events through proper event definition and handling.
 - Defining
 - Register/Deregister
- 9. Lambda Expressions**
- Streamline event handling and delegate instantiation using concise lambda expressions.
 - Using lambdas to generate a delegate
- 10. Inheritance**
- Implement inheritance and polymorphism to create extensible and maintainable
- object-oriented designs.
 - Syntax
 - Override/Virtual
 - Abstract
 - Polymorphism
- 11. SOLID Principles & Design Patterns**
- Apply SOLID principles and design patterns to diagnose and refactor code smells for improved software architecture.
 - Design Smells
 - Single Responsibility
 - Open/Close
 - Liskov’s Substitution Principle
 - Using SOLID to correct design smells
- 12. Interfaces**
- Design and utilize interfaces effectively to decouple code and enhance modularity and testability.
 - Interface motivations and SOLID
 - Interface Segregation
 - Dependency Inversion
 - Default interface methods
- 13. Linq**
- Manipulate collections efficiently using LINQ queries and extension methods for more expressive data operations.
 - Extension Methods
 - Linq API
 - Linq Query
- 14. Attributes, Reflection & Metadata**
- Employ reflection and attributes to manipulate
- metadata and enhance the functionality of code at runtime.
 - Reflections basics
 - Attributes
- 15. Resources & Globalization**
- Manage application resources and adapt to different cultures and languages for global software compatibility.
 - Resource Bundles
 - Specifying Language/Region
- 16. Threads, Tasks, Async/Await**
- Develop efficient, responsive, and scalable applications using advanced concurrency and asynchronous programming techniques.
 - Creating Threads
 - Coordinating threads
 - Tasks
 - Async/Await
- 17. Basics of Entity Framework**
- Leverage Entity Framework for streamlined data access and manipulation, utilizing code-first or database-first approaches effectively.
 - DbContext
 - Code first vs Db first
 - Entities
 - Mapping
- 18. Survey of GUI Applications**
- Explore and understand the latest trends in GUI application development to create intuitive and modern user experiences.
 - MAUI
 - Blazor
 - Asp.Net

Setup Made Simple! Learning Experience Platform (LXP)

All applicable course software, digital courseware files or course notes, labs, data sets and solutions, live coaching support channels, CodeCoach.AI anytime tutor access, and rich extended learning and post training resources are provided for you in our “easy access, single source, no install required” online **Learning Experience Platform (LXP)**, remote lab and content

environment. Access periods vary by course. We'll collaborate with you to ensure your team is set up and ready to go well in advance of the class. Please inquire about set up details and options for your specific course of interest.

For More Information

For more information about our training services (instructor-led, self-paced or blended), collaborative coaching services, robust Learning Experience Platform (LXP), Career Experiences, public course schedule, partner programs, courseware licensing options or to see our complete list of course offerings, solutions and special offers, please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**. Our pricing and services are always satisfaction guaranteed.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS • PUBLIC / OPEN ENROLLMENT COURSES
LEARNING EXPERIENCE PLATFORM (LXP) • COACHING / MENTORING • ASSESSMENTS • CONTENT LICENSING & DEVELOPMENT
LEARNING PLAN DEVELOPMENT • SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS
PARTNER & RESELLER PROGRAMS • CORPORATE TRAINING MANAGEMENT • VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm

