

**Document Generated: 10/31/2024**

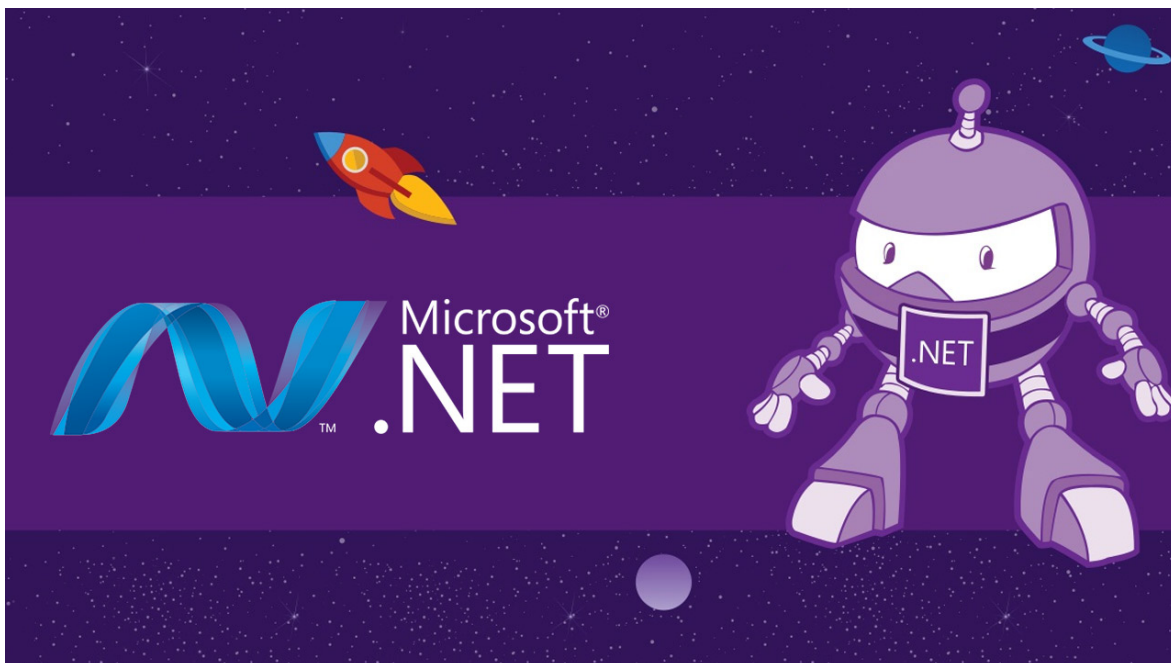
**Learning Style: On Demand**

**Provider: Microsoft**

**Difficulty: Intermediate**

**Course Duration: 12 Hours**

## Asynchronous Programming in C# and .NET Core



### About this Course:

This intermediate training program is designed specifically for Professionals Developers and Programs and gives a comprehensive overview of Asynchronous Programming in .NET Core and C#. Professionals get to learn about innovative programming techniques and their applications. On average, a .NET Programmer earns \$119,500 annually.

This course encompasses a combination of video and text-based learning materials and also includes a series of practical labs. Professionals also get to learn about the art of applying these techniques in enterprise software development lifecycle. In addition to this, this course also reviews real-world programming applications and

identifies the areas where Asynchronous Programming could make the most impact.

In the practical labs, students build familiarity with both web and desktop application development and learn the implementation techniques on both platforms. Students also get to learn the art of using parallel programming techniques to write codes applicable to common use cases.

## **Course Objectives:**

The core objective of this course is to help professionals develop a better understanding and sound knowledge of the following key concepts:

- Fundamental Knowledge and Understanding of Asynchronous Programming
- Built-in Asynchronous Methods for .NET Framework
- Asynchronous Methods Development and Creation
- .NET Framework Asynchronous Method Working Technique
- Multithreading in .NET Core and C#
- Parallel Code Writing with Tasks
- C# Parallel Class Code Writing
- Parameters and Types of Async Returns
- Dealing with the Exceptions of Async Return
- Asynchronous Code Testing and Evaluation

## **Audience:**

This course is tailored for the following group of professionals and interested candidates:

- Developers & Programmers
- Application Developers

## **Prerequisites:**

Professionals planning to enroll in the Asynchronous Programming in C# and .NET Core course must comply with the following prerequisites:

- Conceptual Knowledge and Familiarity with HTML
- Professionals should have Programming in C# (MS-20483) Certification

## **Course Outline:**

### **Introduction to Asynchronous Programming**

- Introducing Asynchronous Programming
- .NET Framework built-in Async Methods
- Asynchronous Support in Microsoft
- Creating Async Methods
- Understand how an Async Method Works

- Lab 00-Setting up the environment
- Lab 01-Build async code in
- Lab 02-Build async code in ASP
- Self Assessment Lab
- ModuleTest

## **Introducing Threads**

- Introduction to Threads and Multithreading
- Introducing Tasks
- The CSharp Parallel class
- Lab 01-Write parallel code with Tasks
- Lab 02-Write parallel code with CSharp Parallel class
- Self Assessment Lab
- Module Test

## **Return Types and Parameters**

- LabAsync Return Types
- Async Exception patterns
- Async Testing patterns
- Lab 00-Set up test projects
- Lab 01-Return Types
- Lab 02-Async exceptions and async testing patterns
- Self Assessment Lab
- ModuleTest

## **Async Best Practices and advanced topics**

- Task-Based Asynchronous Pattern (TAP)
- Legacy async patterns
- Working with legacy async code
- Lab 01-Building TAP-compliant code
- Lab02-Consuming and refactoring legacy async code
- Self Assessment Lab
- ModuleTest

## **Course Final Exam**

- Course Final Exam?

## **Credly Badge:**

**Display your Completion Badge And Get The Recognition You Deserve.**

Add a completion and readiness badge to your LinkedIn profile, Facebook page, or Twitter account to



validate your professional and technical expertise. With badges issued and validated by Credly, you can:

- Let anyone verify your completion and achievement by clicking on the badge
- Display your hard work and validate your expertise
- Display each badge's details about specific skills you developed.

Badges are issued by QuickStart and verified through Credly.

[Find Out More](#) or [See List Of Badges](#)