Async / Parallel & Multithreaded Programming Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Async / Parallel & Multithreaded Programming Training is designed to provide participants with the knowledge and practical skills required to manage asynchronous operations, apply parallel programming techniques, and develop thread-safe applications within the **.NET** environment.

In modern software development, application performance and scalability depend on selecting appropriate execution models. This course examines the differences between **Synchronous** and **Asynchronous Programming**, while exploring suitable approaches for **CPU-Bound** and **I/O-Bound** operations.

Participants learn how to use **async/await**, **Task**, **Task<T>**, **Thread**, and **Thread Pool** in **C#** and **.NET** applications, gaining an understanding of asynchronous method design and concurrent task execution.

The program covers coordinating multiple operations through **Task.WhenAll**, **Task.WhenAny**, **CancellationToken**, and **Exception Handling**. It also examines **Parallel Programming** techniques for optimizing CPU-intensive workloads and introduces **Thread Safety** principles for multithreaded applications.

Additional topics include common concurrency problems such as **Race Conditions**, **Deadlocks**, and **Shared State**, along with modern .NET capabilities such as **IAsyncEnumerable**, **ValueTask**, and **BackgroundService**.

The course also explores asynchronous programming approaches relevant to applications built with **EF Core**, **HttpClient**, and **ASP.NET Core**.

Prerequisites

Participants are expected to have basic knowledge of **C#** and **application development with .NET**.

Who Should Attend

- **.NET Developers**
- **Backend Software Engineers**
- **Full-Stack Developers**
- **DevOps Professionals**
- **Systems Engineers**

Async / Parallel & Multithreaded Programming Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

Upon completing this training, you will be able to:

- Explain the differences between **Synchronous** and **Asynchronous Programming**.
- Evaluate suitable execution models for **CPU-Bound** and **I/O-Bound** operations.
- Develop asynchronous methods using **async/await**, **Task**, and **Task<T>**.
- Understand the fundamental behavior of **Threads** and the **Thread Pool**.
- Apply asynchronous programming techniques in **EF Core**, **HttpClient**, and **ASP.NET Core** applications.
- Coordinate multiple tasks using **Task.WhenAll** and **Task.WhenAny**.
- Manage cancellation of asynchronous operations with **CancellationToken**.
- Handle errors in asynchronous tasks through **Exception Handling**.
- Evaluate **Parallel Programming** techniques for CPU-intensive workloads.
- Identify **Race Conditions**, **Deadlocks**, and **Shared State** problems and apply appropriate **Thread Safety** practices.
- Use modern .NET features such as **IAsyncEnumerable**, **ValueTask**, and **BackgroundService**.
- Select appropriate programming approaches for building more **scalable**, **performance-oriented**, and **resilient** .NET applications.

Outline

1. Async Programming Fundamentals

Synchronous and Asynchronous Programming

- Comparing **Synchronous Programming** and **Asynchronous Programming**
- Understanding **CPU-Bound** and **I/O-Bound** operations
- Selecting suitable programming approaches for different workloads

Thread and Task Structures

- Understanding **Threads**
- Working with **Tasks**
- **Thread Pool** fundamentals

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Using **async/await**
- Working with **Task** and **Task<T>**

Asynchronous Method Design

- **Async Method Design** principles
- Structuring asynchronous methods
- Applying asynchronous operations in .NET applications

2. Async Programming in .NET Core

This section focuses on asynchronous programming capabilities within the**.NET Core** ecosystem.

- Asynchronous programming in **.NET Core**
- Asynchronous data operations with **EF Core**
- Making asynchronous requests using **HttpClient**
- Applying async patterns in **ASP.NET Core**
- Evaluating the impact of asynchronous programming on application performance and scalability
- Modern asynchronous constructs such as **IAsyncEnumerable** and **ValueTask**

3. Concurrent Task Management

- Managing multiple **Task** operations
- Using **Task.WhenAll** to await the completion of multiple tasks
- Using **Task.WhenAny** to identify the first completed task
- Implementing cancellation with **CancellationToken**
- **Exception Handling** in asynchronous operations
- Coordinating concurrent tasks

4. Parallel Programming

- Fundamental principles of **Parallel Programming**
- Applying parallel execution to **CPU-Bound** operations
- Evaluating the performance of CPU-intensive workloads
- Managing parallel operations
- Understanding the differences between parallel and asynchronous programming

Async / Parallel & Multithreaded Programming Training

Duration: **1 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

5. Multithreading and Thread Safety

- Introduction to **Multithreading**
- Fundamental principles of multithreaded applications
- **Thread Safety** concepts
- Understanding **Race Conditions**
- Identifying **Deadlocks**
- Managing **Shared State** issues
- Principles for developing safe and resilient multithreaded applications
- Managing background operations with **BackgroundService**