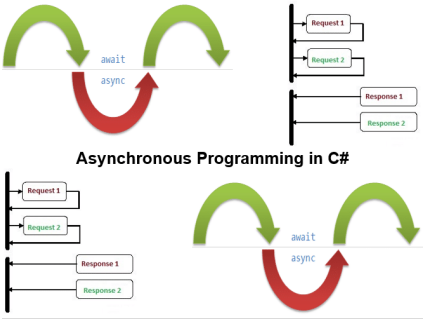
# Asynchronous Programming in C#

**Duur: 2 dagen** **Prijs: € 1499**

Open Rooster




Asynchronous Programming in C#




### Doelgroep Cursus Asynchronous Programming in C#

Deze cursus is bedoeld voor personen die een overzicht willen krijgen van de kenmerken en functies van het .NET framework.

### Voorkennis Cursus Asynchronous Programming in C#

Kennis van software development en een enige kennis van programmeren is vereist om aan deze cursus te kunnen deelnemen.

### Uitvoering Training Asynchronous Programming in C#

De concepten worden behandeld aan de hand van presentatie slides. De besproken concepten worden gedemonstreerd met demo's in Visual Studio.NET. De cursustijden zijn van 9.30 tot 16.30.

### Certificering Cursus Asynchronous Programming in C#

De deelnemers krijgen na het goed doorlopen van de cursus een certificaat Asynchronous Programming in C#.

## Inhoud Cursus Asynchronous Programming in C#

In de cursus Asynchronous Programming in C# staat centraal hoe C# en het .NET Framework kunnen worden gebruikt voor de schrijven van asynchrone code met het async en await mechanisme.

### Intro Asynchronous Programming

De cursus Asynchronous Programming in C# gaat van start met een bespreking van de verschillen tussen synchrone en asynchrone code. Aan de orde komen de nadelen van blocking behavior en de voordelen van parallelism en concurrency met threads en tasks.

### Synchronous versus Asynchronous

Ingegaan hoe bij synchrone code een thread met een blocking call wacht op het resultaat. Bij asynchrone code wordt niet gewacht, maar voert een andere thread de call uit, terwijl de oorspronkelijke thread via een callback of ander mechanisme wordt gewaarschuwd als het resultaat klaar is.

### Async Programming in .NET

Vervolgens wordt de implementatie van asynchrone code in het .NET Framework en .NET Core behandeld. Diverse patterns voor het schrijven van asynchrone code komen daarbij aan de orde. De voordelen van het gebruik van .NET Core komen aan de orde en er wordt ingegaan op asynchrone algorithmes.

### Async Await

Dan wordt het async await mechanisme behandeld. Aan de orde komt hoe een method voorafgegaan door het async keyword een asynchrone method wordt. In de body van de method kan dan met await op het resultaat van een asynchrone call worden afgewacht.

### Synchronization

Ook het voorkomen van data corruptie door middel van synchronisatie primitieven zoals locks, mutexes en semaphores staat op het programma van de cursus Asynchronous Programming in C#. Daarbij wordt ook aandacht besteed aan race conditions en deadlock.

### Exception Handling

Verder komt exception handling in een asynchrone omgeving en de orde. Daarbij wordt ingegaan op faulted tasks en disposable objects. Tenslotte wordt het asynchroon aanroepen van services besproken en de interactie tussen afhandeling in de frontend en de backend.

# Modules Cursus Asynchronous Programming in C#

| Module 1 : Async Intro | Module 2 : Async in .NET | Module 3 : Async Await |
|---|---|---|
| Synchronous Code | Async in .NET | Async Keyword |
| Blocking Behavior | IAsyncResult | Async Method |
| Asynchronous Code | Asynchronous Patterns | Await Keyword |
| Callbacks | Event Based Pattern | Suspending Execution |
| Completion Events | Task Based Pattern | Yielding Control |
| Threads and Tasks | Async .NET Core | Awaitable Tasks |
| Parallelism and Concurrency | .NET Core Benefits | ConfigureAwait |
| IO Bound Tasks | Asynchronous Algorithms | GetAwaiter |
| CPU Bound Tasks | Thread Pools | Task Completion |
| Long Running Tasks | Thread Pool Starvation | Task Composition |
| Background Workers | Memory Consumption | Task Object |
| **Module 4 : Synchronization** | **Module 5 : Exceptions** | **Module 6 : Advanced Topics** |
| Race Conditions | Exception Handling | Async Services |
| Deadlock | Asynchronous Exceptions | Async Request |
| Need for Synchronization | Throwing Exceptions | Ajax Calls |
| Thread Safe Code | Task.Exception Property | Async Frontend |
| Lock Objects | Faulted Tasks | Async Backend |
| Mutexes | Catching Exceptions | Await Tasks Efficiently |
| Semaphores | Disposable Objects | WhenAll |
| Timing and Synchronization | AggregateException | WhenAny |

**SpiralTrain BV**
Standerdmolen 10, 2e verdieping
3995 AA Houten

**info@spiraltrain.nl**
www.spiraltrain.nl
Tel.: +31 (0) 30 – 737 0661

**Locaties**
Houten, Amsterdam, Rotterdam, Eindhoven, Zwolle, Online