

C# Programmeren

Doelgroep Cursus C#

Deze cursus is bedoeld voor aankomende developers die de programmeertaal C# en de toepassing daarvan in .NET applicaties willen leren.

Voorkennis C# Sharp

Voor deze cursus is geen specifieke voorkennis vereist. Ervaring in andere programmeer talen zoals JavaScript, [Java](#) of [C++](#) is bevorderlijk voor de begripsvorming.

Uitvoering Training C# Programmeren

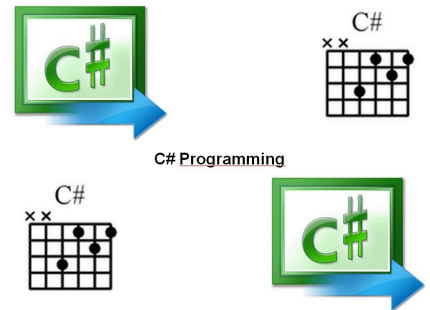
De theorie wordt gepresenteerd aan de hand presentatie slides. Demo's dienen ter verheldering van de behandelde concepten. De theorie wordt afgewisseld met oefeningen. De cursustijden zijn van 9.30 tot 16.30.

Certificering C#

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat C# Programmeren.

Duur: 5 dagen

Prijs: € 2650

[Open Rooster](#)


Inhoud Cursus C# Programmeren

In de cursus C# Programmeren leren de deelnemers programmeren in het .NET platform met de C# (C Sharp) taal. De nadruk van de cursus ligt op de C# syntax, programma structuur en implementatie details. De opgedane C# kennis is zowel toepasbaar in het .NET Framework als in .NET Core. The meest recente versie van C# wordt in de cursus gebruikt.

C# Intro

De cursus C# Programmeren gaat van start met een bespreking van de essenties van het .NET Framework en .NET Core. Daarbij wordt ingegaan op de Common Language Runtime, managed code, assemblies en garbage collection.

Language Syntax

Vervolgens wordt aandacht besteed aan variabelen, data types, operatoren en loops. Ook het aanroepen van methoden en omgaan met arrays en strings staat op het programma van de cursus.

Classes en Objects

Dan komt object georiënteerd programmeren met classes en objects aan de orde. Begrippen als encapsulation, inheritance en polymorphism worden uitgelegd. En ook is er aandacht voor fout afhandeling door middel van exception handling.

Multithreading

Vervolgens leren de deelnemers werken met multiple threads en de implementatie van concurrent tasks. Ook de coördinatie tussen threads door middel van synchronisatie mechanismen als events en Monitor Wait en Pulse komt aan bod.

Special Classes

Op het programma van de cursus C# Programmeren staan ook een aantal special classes zoals delegates, lambda's, properties, indexers and attributes. En er wordt aandacht besteed aan Reguliere Expressies met de RegExp class.

Generics en Collections

De cursus C# Programmeren wordt afgesloten met de behandeling van geparametriseerde types en methoden die generics worden genoemd. Generics worden veel toegepast bij collection classes die vervolgens op het programma staan. Tenslotte wordt aandacht besteed aan File I/O met C# libraries.

Modules Cursus C# Programmeren

Module 1 : C# Intro	Module 2 : Language Syntax	Module 3 : Classes and Objects
C# Versions .NET Architecture .NET Core Common Language Runtime Managed Code C# Compilation and Execution Managed Execution Assemblies MSIL and Metadata Garbage Collection .NET Framework Class Library	C# Data Types Variables and Scope Operators Flow Control if and switch Statement for and foreach Loops while Statement do while Statements break and continue Strings and Arrays Methods and Parameter Passing	Class Definition Encapsulation Access Modifiers Constructors Creating Objects Fields and Properties static Modifier Overloading Constants Common Type System Value and Reference Types
Module 4 : Inheritance	Module 5 : Exception Handling	Module 6 : Namespaces
Derived Classes Overriding Methods Hiding Methods Polymorphism Abstract Classes Interfaces Implementing Interfaces Type Casting Implicit and Explicit Casting	Error Conditions Exceptions in C# Exception Handling Syntax Exception Flow Exceptions Template Exceptions Object finally Clause Throwing Exceptions User Defined Exceptions	Defining Namespaces Using Namespaces Nested Namespaces Namespace Directory Assemblies and Modules Assembly Manifest Types of Assemblies Global Assembly Cache Strong Names
Module 7 : Threads	Module 8 : Synchronization	Module 9 : Special Classes
Thread Benefits and Drawbacks C# Thread Model Thread Class Thread Stack Thread Delegate Autonomous Classes Passing Parameters Thread Naming Background Threads Thread Exceptions Thread Methods	Concurrent Method Invocation Blocking on Monitor Lock Statement Mutual Exclusion in C# Joining Threads Interrupting Threads DeadLock Wait Handles Interthread Communication Condition Synchronization Monitor Wait and Pulse	What is a Delegate? Multicasting Delegates and Events Enumerations Extension Methods Partial Classes Attributes Attribute Parameters Custom Attributes Nullable Types Static Classes
Module 10 : Utility Classes	Module 11 : Generics	Module 12 : Collections
Object Class Boxing and Unboxing Overriding Equals Math Class DateTime Structure Regex Class Process and Environment Class Localizing Dates and Numbers	What are Generics? Need for Generics Generic Class Syntax Multiple Generic Parameters Bounded Types Runtime Type Parameter Constraints Generic Methods	Framework Classes Predefined Collections Array and List Class Queue and Stack Class Linked List Sorted List Dictionary Hashtable
Module 13 : File I/O		
I/O Classes Accessing Text Files Using Directive Accessing Binary Files Buffered Streams Serialization Accessing File System Directory Classes		