SpriralTrain BV
Standerdmolen 10, 2e verdieping.
3995 AA Houten

spiraltrain@gmail.com
www.spiraltrain.nl
Tel.: (0)30 737 0661
Fax: (0)30 737 1050

KvK nr. : 61618039
BTW nr. : NL854415798B01
IBAN : NL58RABO0156271540
BIC : RABONL2U

NET200: Cursus C# / C Sharp Programmeren

Code: NET200 **Duur:** 5 dagen **Prijs:** € 2250

Doelgroep Cursus C# / C Sharp Programmeren

Deze cursus is bedoeld voor ervaren developers met een achtergrond in Java, C++, Delphi of Visual Basic.

Voorkennis C Sharp

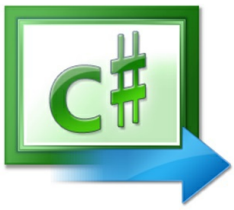
Deelnemers aan deze cursus moeten ervaring in Java, C++, Delphi of Visual Basic hebben. Ook worden zij geacht bekend te zijn met de basisprincipes van het .NET Framework.

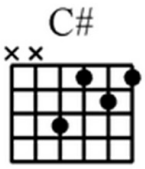
Uitvoering Training C# Programmeren

De theorie wordt gepresenteert 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.

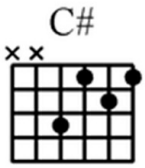
Certificering C#

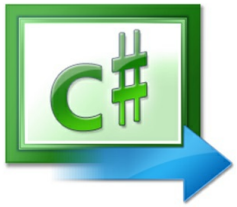
De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat C# Programmeren.





C# Programming





Inhoud Cursus C#

In deze cursus leren de deelnemers programmeren in het .NET platform met de C# taal.
De nadruk van de cursus ligt op de C# syntax, programma structuur en implementatie details. Na het volgen van de cursus zullen de deelnemers in staat zijn de rol die C# speelt in het .NET Framework en het .NET Platform te beschrijven. Ze zijn in staat om een eenvoudige applicatie te programmeren, die te documenteren, te debuggen en te compileren. De deelnemers leren gebruik te maken van variabelen, data types, operatoren, loops en exception handling. Ook leren zij methoden aan te roepen en te schrijven, om te gaan met arrays en object georienteerd programmeren met classes en objects. Bovendien leren de deelnemers werken met delegates, events, properties, indexers and attributes. The meest recente versie van C# wordt in de cursus gebruikt.

Module 1 : C# Intro	**Module 2 : Language Syntax**	**Module 3 : Classes and Objects**
What is C#? C# Versions .NET Versions .NET Architecture Common Language Runtime Managed Code C# Compilation Compilation and Execution Managed Execution Assemblies MSIL and Metadata Application Types Garbage Collection .NET Framework Class Library C# Resources	C# Data Types Console IO Variables Variable Scope Case Sensitivity Operators Flow Control if Statement switch Statement for Loops foreach Statement while Statement do Statements break and continue Strings Arrays Methods Parameter Passing	Classes and Objects Example Class and Objects Class Definition Encapsulation Access Modifiers Constructors Creating Objects Fields Properties Special Properties static Modifier Overloading Constants Common Type System Value Types Reference Types Object References
Module 4 : Inheritance	**Module 5 : Exception Handling**	**Module 6 : Namespaces**
Inheritance Derived Classes Overriding Methods Hiding Methods Polymorphism Abstract Classes Interfaces Implementing Interfaces Type Casting Implicit Casting Explicit Casting	Error Conditions Exceptions in C# Exception Handling Syntax Exception Flow Exceptions Template Exceptions Object finally Clause Throwing Exceptions User Defined Exceptions Catching User Exceptions	What are Namespaces? .NET Namespaces Defining Namespaces Using Namespaces Nested Namespaces Namespace Aliases Namespace Directory Assemblies Modules MathLibrary Module Assembly Manifest AssemblyInfo Using MathLibrary Types of Assemblies Global Assembly Cache Strong Names

Module 7 : Threads	Module 8 : Synchronization	Module 9 : Special Classes
Multiple Threads Benefits and Drawbacks Thread Characteristics C# Thread Model Thread Class Thread Stack Thread Delegate Multiple Threads Autonomous Classes Passing Parameters Thread Naming Background Threads Thread Exceptions Thread Methods	Concurrent Method Invocation Synchronization 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? Benefits of Delegates Multicasting Delegates and Events Simple Event Handling Enumerations Enumeration Base Types 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 Input Conversion Convert Class Process Class Environment Class Globalization Localizing Dates Localizing Numbers XML Documentation	What are Generics? Need for Generics Generic Class Syntax Benefits of Generics Multiple Generic Parameters Runtime Type Parameter Constraints Generic Methods	What are Collections? Framework Classes Properties of Collections Predefined Collections Array Class List Class Queue Class Queue Methods Stack Class Stack Methods Linked List Sorted List Dictionary Hashtable BitArray
Module 13 : File I/O		
Stream I/O I/O Classes File Types Writing Text File Reading Text File Using Directive Accessing Binary Files Buffered Streams Serialization Implementing Serialization Accessing File System Directory Classes		