



# Startel

*Dé IT-opleider*

## C# Fundamentals

In deze training leer je de benodigde programmeervaardigheden om programma's te maken met behulp van de C#-taal. Hiervan leer je de basisprincipes van de programmastructuur, taalsyntaxis en implementatiedetails. Vervolgens creëer jij een programma dat verschillende functies van .NET bevat.

### Algemene omschrijving

Deze training leert ontwikkelaars de benodigde programmeervaardigheden om programma's te maken met behulp van de C#-taal. Je leert de basisprincipes van de programmastructuur, taalsyntaxis en implementatiedetails. Vervolgens creëer jij een programma dat verschillende functies van .NET bevat.

### Doelgroep

Deze training is bedoeld voor ervaren ontwikkelaars die al programmeerervaring hebben in C, C++, JavaScript, Objective-C, Microsoft Visual Basic of Java en de concepten van objectgeoriënteerd programmeren begrijpen.

### Leerdoelen

Na afloop van deze training ben je in staat om het volgende te doen:

- Uitleggen hoe je Visual Studio kunt gebruiken om een programma te maken en uit te voeren.
- De functies en syntaxis beschrijven van de programmeertaal C#.
- De benodigdheden voor monitoring definiëren voor grootschalige programma's.
- Methoden maken en aanroepen, uitzonderingen vastleggen en beheren.
- Je begrijpt het .NET-ontwikkelingsplatform en de bibliotheken.
- Je begrijpt de .NET Framework-klassen.
- Goed gestructureerde en gemakkelijk te onderhouden C#-code creëren
- Interfaces definiëren en implementeren.
- Een klassenhiërarchie met behulp van overerving maken.
- Je begrijpt objectgeoriënteerde programmeerconcepten.
- De fundamentele architectuur en kerncomponenten van een computerprogramma implementeren.
- Bestands-I/O en streams, en serialiseer-/deserialiseergegevens in verschillende formaten gebruiken.
- Je begrijpt het gebruik van generieken en generieke gegevensverzamelingen.

## Voorkennis

Voordat jij deze training gaat volgen, raden wij aan dat je programmeerervaring hebt in C, C++, JavaScript, Objective-C, Microsoft Visual Basic of Java en dat je de concepten van objectgeoriënteerd programmeren begrijpt.

## Onderwerpen

### Module 1: C# Syntax

Microsoft .NET 6 provides a comprehensive development platform that you can use to build, deploy, and manage applications and services. By using .NET, you can create visually compelling applications, enable seamless communication across technology boundaries, and provide support for a wide range of business processes.

In this module, you'll learn about some of the core features provided by .NET and Microsoft Visual Studio. You'll also learn about some of the core C# constructs that enable you to start developing .NET applications.

Lessons:

- Lesson 1: Writing Applications in C# and .NET.
- Lesson 2: Types of Data and Expressions.
- Lesson 3: C# Language Constructs.

### Lab 1: Developing the Class Enrolment Application

- Developing the Class Enrolment Application.

After completing this module, students will be able to:

- Write Applications in C# and .NET.
- Explain types of Data and Expressions.
- Understand C# Language Constructs.

### Module 2: C# Language Concepts

Applications often consist of logical units of functionality that perform specific functions, such as providing access to data or triggering some logical processing. C# is an object-orientated language and uses the concept of methods to encapsulate logical units of functionality. Although a good practice is to have methods that do just one thing, they can be as simple or as complex as you like. It is also important to consider what happens to the state of your application when an exception occurs in a method.

Lessons:

- Lesson 1: Methods.
- Lesson 2: Method Overloading.
- Lesson 3: Exception Handling.
- Lesson 4: Monitoring.

### Lab 1: Extending the Class Enrolment Application

- Refactor code to facilitate reusability.
- Write C# code that validates data entered by a user.
- Write C# code that saves changes back to a database.

After completing this module, students will be able to:

- Create and use methods and handle exceptions.
- Use logging and tracing to record the details of any exceptions that occur.

### **Module 3: C# Structures, Collections and Events**

To create effective applications you must first learn some fundamental C# constructs. You need to know how to create simple structures to represent the data items you are working with. You need to know how to organize these structures into collections, so that you can add items, retrieve items, and iterate over your items. Finally, you need to know how to subscribe to events so that you can respond to the actions of your users.

Lessons:

- Lesson 1: Structs.
- Lesson 2: Enums.
- Lesson 3: Built-in Collections.
- Lesson 4: Events.

#### **Lab 1: Building the Grades Prototype Application**

- Structs.
- Enums.
- Built-in Collections.
- Events.

After completing this module, students will be able to:

- Create and use structs and enums.
- Organize data into collections.
- Create and subscribe to events.

### **Module 4: C# Classes**

In this module, you'll learn how to use interfaces and classes to define and create your own custom, reusable types. You'll also learn how to create and use enumerable type-safe collections of any type.

Lessons:

- Lesson 1: Creating Classes.
- Lesson 2: Interfaces.
- Lesson 3: Understanding Generics in C#.

#### **Lab 1: Adding Data Validation to the Application**

- Creating Classes.
- Interfaces.
- Understanding Generics in C#.

After completing this module, students will be able to:

- Use interfaces and classes to define and create custom, reusable types.
- Create and use enumerable type-safe collections of any type.

## **Module 5: C# Inheritance**

In this module, you'll learn how to use inheritance to create class hierarchies and to extend .NET types.

Lessons:

- Lesson 1: Hierarchies of Classes.
- Lesson 2: Polymorphism.
- Lesson 3: Extending Classes.

### **Lab 1: Refactoring**

- Hierarchies of Classes.
- Polymorphism.
- Extending Classes.

After completing this module, students will be able to:

- Use inheritance to factor common functionality into a base class.
- Implement polymorphism by using an abstract method.
- Create a custom exception class.

## **Module 6: Input and Output**

In this module, you'll learn how to read and write data by using transactional filesystem I/O operations, how to serialize and deserialize data to the filesystem, and how to read and write data to the filesystem by using streams.

Lessons:

- Lesson 1: File I/O.
- Lesson 2: Serialization and Deserialization.
- Lesson 3: Streams.

### **Lab 1: Creating the Grades Report**

- File I/O.
- Serialization and Deserialization.
- Streams.

After completing this module, students will be able to:

- Read and write data by using transaction filesystem I/O operations.
- How to searlize and deserialize data to the file system.
- How to read and write data to the filesystem by using streams.