

# Kurs-Dokumentation



## Zentrum für Informatik ZFI AG

### Programming in C# with Visual Studio 2012 (NPCV)

<http://www.zfi.ch/NPCV>

Weitere Infos finden Sie unter [www.zfi.ch](http://www.zfi.ch) oder via Adresse:

Zentrum für Informatik ZFI AG  
Zentralsekretariat  
Rütistrasse 28  
CH-8952 Zürich-Schlieren  
Telefon: 044 732 40 00  
Telefax: 041 530 31 68

Zürich, Basel, Bern, ZÄ¼rich, Schweiz

<b>Titel</b>	<b>Programming in C# with Visual Studio 2012</b>
<b>Untertitel</b>	<b>for experienced developers with programming experience in OO languages</b>
<b>Einleitung</b>	<p>This training course teaches developers the programming skills that are required for developers to create Windows applications using the C# language. During their five days in the classroom students review the basics of C# program structure, language syntax, and implementation details, and then consolidate their knowledge throughout the week as they build an application that incorporates several features of the .NET Framework 4.5.</p> <p>The course introduces many of the techniques and technologies employed by modern desktop and enterprise applications, including:</p> <ul style="list-style-type: none"><li>Building new data types</li><li>Handling events</li><li>Programming the user interface</li><li>Accessing a database</li><li>Using remote data</li><li>Performing operations asynchronously</li><li>Integrating with unmanaged code</li><li>Creating custom attributes</li><li>Encrypting and decrypting data</li></ul> <p>At the end of the course, students should leave the class with a solid knowledge of C# and how to use it to develop .NET Framework 4.5 applications.</p> <p>This course uses Visual Studio 2012, running on Windows 8.</p>
<b>Ihr Nutzen</b>	<p>After completing this course, students will be able to:</p> <ul style="list-style-type: none"><li>- Describe the core syntax and features of C#</li><li>- Create and call methods, catch and handle exceptions, and describe the monitoring requirements of large-scale applications</li><li>- Implement the basic structure and essential elements of a typical desktop application</li><li>- Create classes, define and implement interfaces, and create and use generic collections</li><li>- Use inheritance to create a class hierarchy, extend a .NET Framework class, and create generic classes and methods</li></ul>

- Read and write data by using file input/output and streams, and serialize and deserialize data in different formats
- Create and use an entity data model for accessing a database and use LINQ to query and update data
- Use the types in the System.Net namespace and WCF Data Services to access and query remote data
- Build a graphical user interface by using XAML
- Improve the throughput and response time of applications by using tasks and asynchronous operations
- Integrate unmanaged libraries and dynamic components into a C# application
- Examine the metadata of types by using reflection, create and use custom attributes, generate code at runtime, and manage assembly versions
- Encrypt and decrypt data by using symmetric and asymmetric encryption

#### Voraussetzungen

Developers attending this course should already have gained some limited experience using C# to complete basic programming tasks. More specifically, students should have hands-on experience using C# that demonstrates their understanding of the following:

- How to name, declare, initialize and assign values to variables within an application
- How to use:
  - arithmetic operators to perform arithmetic calculations involving one or more variables

- relational operators to test the relationship between two variables or expressions
  
- logical operators to combine expressions that contain relational operators
  
- How to create the code syntax for simple programming statements using C# language keywords and recognize syntax errors using the Visual Studio IDE
  
- How to create a simple branching structure using an IF statement
  
- How to create a simple looping structure using a For statement to iterate through a data array
  
- How to use the Visual Studio IDE to locate simple logic errors
  
- How to create a Function that accepts arguments (parameters and returns a value of a specified type
  
- How to design and build a simple user interface using standard controls from the Visual Studio toolbox
  
- How to connect to a SQL Server database and the basics of how to retrieve and store data
  
- How to sort data in a loop

**Teilnehmerkreis**

- How to recognize the classes and methods used in a program

This course is intended for experienced developers who already have programming experience in C, C++, JavaScript, Objective-C, Microsoft Visual Basic, or Java and understand the concepts of object-oriented programming.

This course is not designed for students who are new to programming; it is targeted at professional developers with at least one month of experience programming in an object-oriented environment.

**Unterlagen****Folgekurse****Inhalt**

**Module 1: Review of C# Syntax  
Overview of Writing Applications using C#**

**Datatypes, Operators, and Expressions**

**C# Programming Language Constructs**

**Module 2: Creating Methods, Handling Exceptions, and Monitoring Applications  
Creating and Invoking Methods**

**Creating Overloaded Methods and Using Optional and Output Parameters**

**Handling Exceptions**

**Monitoring Applications**

**Module 3: Developing the Code for a Graphical Application  
Implementing Structs and Enums**

**Organizing Data into Collections**

**Handling Events**

**Module 4: Creating Classes and Implementing Type-safe Collections**

**Creating Classes****Defining and Implementing Interfaces****Implementing Type-safe Collections****Module 5: Creating a Class Hierarchy by Using Inheritance**  
**Creating Class Hierarchies****Extending .NET Framework Classes****Creating Generic Types****Module 6: Reading and Writing Local Data**  
**Reading and Writing Files****Serializing and Deserializing Data****Performing I/O Using Streams****Module 7: Accessing a Database**  
**Creating and Using Entity Data Models****Querying Data by Using LINQ****Updating Data by Using LINQ****Module 8: Accessing Remote Data**  
**Accessing Data Across the Web****Accessing Data in the Cloud**

## **Module 9: Designing the User Interface for a Graphical Application Using XAML to Design a User Interface**

**Binding Controls to Data**

**Styling a User Interface**

## **Module 10: Improving Application Performance and Responsiveness Implementing Multitasking by using Tasks and Lambda Expressions**

**Performing Operations Asynchronously**

**Synchronizing Concurrent Access to Data**

## **Module 11: Integrating with Unmanaged Code Creating and Using Dynamic Objects**

**Managing the Lifetime of Objects and Controlling Unmanaged Resources**

## **Module 12: Creating Reusable Types and Assemblies Examining Object Metadata**

**Creating and Using Custom Attributes**

**Generating Managed Code**

**Versioning, Signing and Deploying Assemblies**

## **Module 13: Encrypting and Decrypting Data Implementing Symmetric Encryption**

**Implementing Asymmetric Encryption**

**Beitrag**

**Der Teilnehmerbeitrag versteht sich rein netto. Das ZFI ist (gemäss MwSt-Gesetz) nicht Mehrwertsteuerpflichtig und erhebt somit keine**

**MwSt. Bei länger als einen Monat dauernden Lehrgängen ist die Zahlung des Teilnehmerbeitrages in mehreren Raten möglich (pro rata temporis).**