

# C#

## C# Fundamentals, .NET Fx Intro, Types, Classes, Inheritance, Attributes, Delegates, Generics

C# is the premier development language for the .NET platform. C# was designed from scratch with .NET in mind. Most of the internals of the .NET Framework and Visual Studio are written in it. It has been selected by the majority of application teams creating commercial .NET projects. C# builds on the rich common heritage of languages such as C++ and Java - it avoids their pitfalls and adds certain interesting new concepts, such as LINQ. There are some aspects of C# that developers will already know, there are some they have experienced similar but slightly different syntax in other languages, and there are some that are highly

innovative. C# can be used to develop stand-alone apps, local and distributed components, web services and mobile code. It produces code that can target desktop PCs, mobile devices, servers and IoT devices. It is capable of supporting user interface, Internet, database and security projects. Hence it is an excellent all-round development language for all .NET applications. This intensive course aims to take experienced software engineers rapidly through all the major aspects of C# - using plenty of demo source code and hands-on labs to show it in action. This is an ideal first course for those moving to the C# language and the .NET Framework.

<b>Contents of One-Day Training Course</b>	
<p><b>Target Audience</b> Experienced software engineers wishing to rapidly get up to speed with C#.</p> <p><b>Prerequisites</b> Programming experience with an OO language such as C++, TypeScript or JAVA, along with good exposure to object-oriented design.</p> <p>No previous experience of C# or .NET is needed.</p>	<p><b>C# and the .NET Framework</b> What is the .NET Framework? The Base Class Library The CLR How C# is used with .NET Delivery of C# functionality in assemblies</p> <p><b>A C# Project Walk through</b> Solutions, projects and files Parts of a C# project Structure of code Setting up a solution with a C# app and class library</p> <p><b>Base Types</b> Built-in data types .NET value types and reference types How C# and .NET data types compare Building code in C# that is callable from other languages</p> <p><b>Language Fundamentals</b> Main starting point Flow control, operators Variables, methods Enumerators, bit flags, arrays, indexers Namespaces</p> <p><b>Class Fundamentals</b> Members, constructors, visibility, ref and out, constant fields, structs Fields &amp; properties, methods, nested types</p> <p><b>Inheritance</b> Single inheritance only (for classes) Virtual functions Override and new keywords Designing libraries using inheritance</p>
	<p><b>Delegates And Events</b> Equivalent of function pointers Defining and exposing delegates Registering an interest in a delegate Async info with events Design pattern for event handling</p> <p><b>Interfaces</b> When to use interfaces Multiple inheritance &amp; hierarchies Abstract classes vs. interfaces</p> <p><b>Exception Handling</b> Try .. catch ... finally Detecting and responding to exceptions Strategies for exception handling</p> <p><b>Generics &amp; Constraints</b> Generics (for methods and classes) Constraints Partial types Anonymous methods Type inferencing</p> <p><b>Expression Bodied Members</b> Succinct member definitions Methods, constructors, properties, indexers</p> <p><b>Specialist Features</b> Null conditional operator Auto-property initializer nameof</p> <p><b>Calling C Code</b> Calling out to C code from C# Passing parameters and accepting return val</p> <p><b>C# Project</b> Developing a C# project using a selection of language features</p>



<http://www.clipcode.net/training>

To arrange an on-site presentation anywhere in Europe, please email [training@clipcode.com](mailto:training@clipcode.com)