

Preface

C# (pronounced *see sharp*) is Microsoft's answer to Java. C# is a modern programming language for practitioners, combining the latest concepts in software technology, such as object-orientation, component-orientation, security, versioning support, parallel computing, exception handling, reflection and much more.

But do we really need another programming language? What about Java, C++, Visual Basic and all the rest? The reason Microsoft introduced C# is .NET (*dot net*). .NET is Microsoft's new software platform that does away with many of the problems associated with Windows programming. Although the existing languages can be used with .NET, they cannot take full advantage of the features. That is why Microsoft created C#, its own in-house language that fits perfectly with .NET. Anyone working with .NET will sooner or later have to learn C#, although this is not a daunting task as C# has the same pedigree as Java and C++ and is binary-compatible with most standard languages.

C# is an object-oriented language, which means it provides *classes* and *interfaces* with single code inheritance and multiple interface inheritance. Component-based programming is supported by new elements such as *properties*, *events* and *delegates*. The components known as *assemblies* in .NET are dynamically linked and are given a version number, which the loader uses to ensure that the correct version of the software is being used. And finally, digitally-signed assemblies are designed to make life difficult for viruses.

Content

This book describes the whole of the C# language in concise form. The reader will learn about all the features of C#, from simple data types and statements, through object-oriented and component-based programming, to threads, exceptions, attributes and automatically generated documentation. The individual elements of the language are described using examples. The appendix contains the C# grammar, with the precise syntax of each element.

No modern programming language is released without a class library. This is why the book also includes a review of the comprehensive .NET library, which contains types for every conceivable purpose: simple data structures like lists, sets

and hash tables; input/output classes, network classes, GUI controls; even advanced topics like reflection and web programming.

.NET and C# support Internet programming, including *web services* (business-to-business services on the Internet) and dynamic web pages created using the latest *ASP.NET* technology. The book uses case studies to show how to develop web services and dynamic web pages with ASP.NET. The case studies also deal with graphical user interfaces, including how to create them using *Windows Forms* and the *Visual Studio .NET* development environment.

Although C# and .NET are heralding a new era in Windows programming, they still retain their links to COM and the old Win32 technology. One of the chapters in this book explains how to call COM components from .NET and vice versa.

As a language, C# is under constant development. New elements have been announced for the next version of .NET, including *generic types*, *iterators*, *anonymous methods* and *partial types*. Although they were not yet available at the time of publication, there is a chapter dedicated to these elements because of the contribution they make to the power and ease of use of C#.

The companion CD of this book contains the .NET Framework SDK with the common language runtime, the .NET base class library, the C# compiler, various tools, as well as the complete documentation of C# and .NET. It also contains a variety of other C# development tools such as SharpDevelop, ASP.NET Web Matrix, Webservice Studio and Coco/R.

Target Audience

This book is intended for practitioners wishing to learn about C# and for students using the language in advanced courses on object-orientation, component-orientation or .NET. There are exercises at the end of each chapter, and example solutions can be found at <http://dotnet.jku.at>. The book is therefore suitable for self-study.

I would like to thank my colleagues *Wolfgang Beer*, *Dietrich Birngruber* and *Albrecht Wöbß*, with whom I use .NET and C# in teaching and in projects. Microsoft specialists in Redmond, Cambridge and Vienna were always happy to answer my questions. In particular, I would like to thank *Jim Miller*, *Eric Gunnerson*, *Don Syme*, *Van Eden*, *Ralph Zeller*, *Andreas Schabus* and *Alexander Holy*. My thanks also go to the anonymous reviewers who read through the manuscript of this book with such care.

Hanspeter Mössenböck
May 2003