

THE EXPERT'S VOICE® IN .NET

Illustrated C# 2005

C# presented clearly, concisely, and visually

Daniel Solis

Apress®

Illustrated C# 2005



Daniel Solis

Illustrated C# 2005

Copyright © 2006 by Daniel Solis

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-13 (pbk): 978-1-59059-723-1

ISBN-10 (pbk): 1-59059-723-0

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Trademarked names may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, we use the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Lead Editor: Matthew Moodie

Technical Reviewer: Christophe Nasarre

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Matt Wade

Project Manager | Production Director: Grace Wong

Copy Edit Manager: Nicole Flores

Copy Editors: Damon Larson, Sharon Wilkey

Assistant Production Director: Kari Brooks-Copony

Production Editor: Katie Stence

Compositor: Pat Christenson

Proofreader: Lori Bring

Indexer: Michael Brinkman

Cover Designer: Kurt Krames

Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit <http://www.springeronline.com>.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit <http://www.apress.com>.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at <http://www.apress.com> in the Source Code/Download section. You will need to answer questions pertaining to this book in order to successfully download the code.

*I would like to dedicate this book to
my parents, Sal and Amy;
and to Sian and Sue.*

Contents at a Glance

About the Author	xxi
About the Technical Reviewer	xxiii
Acknowledgments	xxv
Introduction	xxvii
■ CHAPTER 1 C# and the .NET Framework	1
■ CHAPTER 2 Overview of C# Programming	15
■ CHAPTER 3 Types, Storage, and Variables	29
■ CHAPTER 4 Classes: The Basics	43
■ CHAPTER 5 Methods	61
■ CHAPTER 6 More About Classes	95
■ CHAPTER 7 Classes and Inheritance	139
■ CHAPTER 8 Expressions and Operators	171
■ CHAPTER 9 Statements	209
■ CHAPTER 10 Namespaces and Assemblies	239
■ CHAPTER 11 Exceptions	265
■ CHAPTER 12 Structs	285
■ CHAPTER 13 Enumerations	295
■ CHAPTER 14 Arrays	307
■ CHAPTER 15 Delegates	333
■ CHAPTER 16 Events	353
■ CHAPTER 17 Interfaces	371
■ CHAPTER 18 Conversions	397
■ CHAPTER 19 Generics	427
■ CHAPTER 20 Enumerators and Iterators	455
■ CHAPTER 21 Attributes	481

■ CHAPTER 22	Preprocessor Directives	503
■ CHAPTER 23	Other Topics	515
■ INDEX		543

Contents

About the Author	xxi
About the Technical Reviewer.....	xxiii
Acknowledgments.....	xxv
Introduction	xxvii
CHAPTER 1 C# and the .NET Framework	1
Before .NET	2
Windows Programming in the Late 1990s	2
Goals for the Next-Generation Platform	2
Enter Microsoft .NET	3
Components of the .NET Framework.....	3
An Improved Programming Environment	4
Compiling to the Common Intermediate Language (CIL).....	7
Compiling to Native Code and Execution	8
Overview of Compilation and Execution	9
The Common Language Runtime (CLR)	10
The Common Language Infrastructure (CLI)	11
Important Parts of the CLI	12
Review of the Acronyms	13
CHAPTER 2 Overview of C# Programming	15
A Simple C# Program.....	16
More About SimpleProgram	17
Identifiers and Keywords	18
Naming Conventions.....	19
Keywords	19
Main: The Starting Point of a Program	20
Whitespace	20
Statements.....	21
Simple Statements	21
Blocks	21

Text Output from a Program	22
Write	22
WriteLine.....	23
The Format String	23
Substituting Values	24
Multiple Markers and Values.....	25
Comments: Annotating the Code	26
More About Comments	27
Documentation Comments	27
Summary of Comment Types	28

■ CHAPTER 3 **Types, Storage, and Variables**

A C# Program Is a Set of Type Declarations	30
A Type Is a Template	31
Instantiating a Type	31
Data Members and Function Members.....	32
Types of Members	32
Predefined Types	33
More About the Predefined Types	34
User-Defined Types	35
The Stack and the Heap	36
The Stack	36
The Heap.....	37
Value Types and Reference Types.....	38
Storing Members of a Reference Type Object	38
Categorizing the C# Types.....	39
Variables.....	40
Variable Declarations	40
Multiple-Variable Declarations	42
Using the Value of a Variable.....	42

■ CHAPTER 4 **Classes: The Basics**

Overview of Classes.....	44
A Class Is an Active Data Structure	44
Programs and Classes: A Quick Example.....	45
Declaring a Class	46

Class Members	47
Fields	47
Methods	49
Creating Variables and Instances of a Class	50
Allocating Memory for the Data	50
Combining the Steps	51
Instance Members	52
Access Modifiers	53
Private and Public Access	53
Accessing Members from Inside the Class	56
Accessing Members from Outside the Class	57
Putting It All Together	58

CHAPTER 5	Methods	61
	The Structure of a Method	62
	Code Execution in the Method Body	63
	Local Variables	64
	Local Variables Inside Nested Blocks	65
	Flow-of-Control	66
	Method Invocations	67
	Return Values	68
	The Return Statement and Void Methods	70
	Parameters	72
	Formal Parameters	72
	Actual Parameters	73
	Value Parameters	75
	Reference Parameters	78
	Output Parameters	81
	Parameter Arrays	84
	Method Invocation	85
	Expanded Form	85
	Arrays As Actual Parameters	88
	Summary of Parameter Types	88
	Stack Frames	89
	Recursion	91
	Method Overloading	93

CHAPTER 6	More About Classes	95
	Class Members	96
	Order of Member Modifiers	96
	Instance Class Members	98
	Static Fields	99
	Accessing Static Members from Outside the Class	100
	Example of a Static Field	100
	Lifetimes of Static Members	101
	Static Function Members	102
	Other Static Class Member Types	103
	Constants	104
	Constants Are Like Statics	105
	Local Constants	106
	Properties	107
	Property Declarations and Accessors	108
	A Property Example	109
	Using a Property	110
	Properties and Associated Fields	111
	Performing Other Calculations	113
	Read-Only and Write-Only Properties	114
	A Computed, Read-Only Property Example	115
	Example of Properties and Databases	116
	Static Properties	117
	Instance Constructors	118
	Constructors with Parameters	119
	Default Constructors	120
	Static Constructors	121
	Example of a Static Constructor	122
	Accessibility of Constructors	122
	Finalizers	123
	Calling the Finalizer	124
	Comparison of Constructors and Finalizers	124
	The readonly Modifier	125
	The this Keyword	126

Indexers	127
What Is an Indexer?	128
Indexers and Properties	128
Declaring an Indexer	129
The set Accessor	130
The get Accessor	131
More About Indexers	131
Declaring the Indexer for the Employee Example	132
An Additional Example of an Indexer	133
Indexer Overloading	134
Access Modifiers on Accessors	135
Partial Classes	136

CHAPTER 7	Classes and Inheritance	139
	Class Inheritance	140
	Accessing the Inherited Members	141
	All Classes are Derived From Class object	142
	Hiding Members of a Base Class	143
	Base Access	145
	Using References to a Base Class	146
	Virtual and Override Methods	148
	Overriding a Method Marked override	150
	Constructor Execution	153
	Constructor Initializers	154
	Class Access Modifiers	156
	Inheritance Between Assemblies	157
	Member Access Modifiers	159
	Regions Accessing a Member	160
	Public Member Accessibility	161
	Private Member Accessibility	161
	Protected Member Accessibility	162
	Internal Member Accessibility	162
	Protected Internal Member Accessibility	163
	Summary of Member Access Modifiers	164
	Abstract Members	165
	Abstract Classes	166
	Example of an Abstract Class and an Abstract Method	167
	Sealed Classes	168
	External Methods	168

CHAPTER 8	Expressions and Operators	171
	Expressions	172
	Literals	173
	Integer Literals	174
	Real Literals	176
	Character Literals	177
	String Literals	178
	Order of Evaluation	180
	Precedence	180
	Associativity	181
	Parenthesized Expressions	181
	Simple Arithmetic Operators	182
	The Remainder Operator	183
	Relational and Equality Comparison Operators	184
	Comparison and Equality Operations	185
	Increment and Decrement Operators	186
	Conditional Logical Operators	188
	Logical Operators	190
	Shift Operators	192
	Assignment Operators	194
	Compound Assignment	195
	The Conditional Operator	196
	Unary Arithmetic Operators	198
	User-Defined Type Conversions	199
	Explicit Conversion and the Cast Operator	201
	Operator Overloading	202
	Restrictions on Operator Overloading	203
	Example of Operator Overloading	203
	The typeof Operator	205
CHAPTER 9	Statements	209
	What Are Statements?	210
	Expression Statements	211
	Flow-of-Control Statements	212
	The if Statement	213
	The if...else Statement	214

The switch Statement	215
A Switch Example	217
More on the switch Statement	218
Switch Labels.	219
The while Loop	220
The do Loop.	221
The for Loop	223
Scope of Variables in a for Statement.	225
Multiple Expressions in the Initializer and Iterator	226
Jump Statements	227
The break Statement	227
The continue Statement	228
Labeled Statements	229
Labels	229
The Scope of Labeled Statements.	230
The goto Statement	231
The goto Statement Inside a switch Statement.	231
The using Statement	232
Packaging Use of the Resource	233
Example of the using Statement.	234
Multiple Resources and Nesting.	235
Another Form of the using Statement	236
Other Statements	237

■ CHAPTER 10 **Namespaces and Assemblies** 239

Referencing Other Assemblies	240
The mscorlib Library	243
Namespaces	245
Namespace Names.	247
More About Namespaces.	248
Namespaces Spread Across Files	249
Nesting Namespaces	250
The using Directives.	251
The using Namespace Directive.	251
The using Alias Directive	252
The Structure of an Assembly	253
The Identity of an Assembly	255

Strongly Named Assemblies	257
Creating a Strongly Named Assembly	258
Private Deployment of an Assembly	259
Shared Assemblies and the GAC	260
Installing Assemblies into the GAC	260
Side-by-Side Execution in the GAC	261
Configuration Files	262
Delayed Signing	263

CHAPTER 11 Exceptions	265
What Are Exceptions?	266
The try Statement	267
Handling the Exception	268
The Exception Classes	269
The catch Clause	270
Examples Using Specific catch Clauses	271
The catch Clauses Section	272
The finally Block	273
Finding a Handler for an Exception	274
Searching Further	275
General Algorithm	276
Example of Searching Down the Call Stack	277
Throwing Exceptions	280
Throwing Without an Exception Object	282

CHAPTER 12 Structs	285
What Are Structs?	286
Structs Are Value Types	287
Assigning to a Struct	288
Constructors and Finalizers	289
Instance Constructors	289
Static Constructors	291
Summary of Constructors and Finalizers	291
Field Initializers	292
Inheritance	292
Boxing and Unboxing	292
Structs As Return Values and Parameters	293
Additional Information About Structs	293

CHAPTER 13	Enumerations	295
	Enumerations	296
	Underlying Types and Values	297
	Setting the Underlying Type	298
	Setting Explicit Values for the Members	298
	Implicit Member Numbering	299
	Bit Flags	300
	The Flags Attribute	301
	Example Using Bit Flags	303
	More About Enums	304
CHAPTER 14	Arrays	307
	Arrays	308
	Definitions	308
	Important Details	308
	Types of Arrays	309
	An Array As an Object	310
	One-Dimensional and Rectangular Arrays	311
	Declaring a One-Dimensional Array or a Rectangular Array	311
	Instantiating a One-Dimensional or Rectangular Array	312
	Accessing Array Elements	313
	Initializing an Array	314
	Automatic Initialization	314
	Explicit Initialization of One-Dimensional Arrays	314
	Explicit Initialization of Rectangular Arrays	315
	Syntax Points for Initializing Rectangular Arrays	315
	Shortcut Syntax	316
	Putting It All Together	316
	Jagged Arrays	317
	Declaring a Jagged Array	318
	Shortcut Instantiation	318
	Instantiating a Jagged Array	319
	Sub-Arrays in Jagged Arrays	320
	Comparing Rectangular and Jagged Arrays	321
	The foreach Statement	322
	The Iteration Variable Is Read-Only	323
	The foreach Statement with Multidimensional Arrays	324
	Array Covariance	326

Useful Inherited Array Members	327
The Clone Method	329
Comparing Array Types	331
CHAPTER 15 Delegates	333
What Is a Delegate?	334
Declaring the Delegate Type	335
Creating the Delegate Object	336
Assigning Delegates	338
Combining Delegates	339
Adding Methods to Delegates	340
Deleting Methods from a Delegate	341
Invoking a Delegate	342
Delegate Example	343
Invoking Delegates with Return Values	344
Invoking Delegates with Reference Parameters	345
Anonymous Methods	346
Using Anonymous Methods	347
Syntax of Anonymous Methods	348
Scope of Variables and Parameters	351
CHAPTER 16 Events	353
Events Are Like Delegates	354
An Event Has a Private Delegate	355
Overview of Source Code Components	356
Declaring an Event	357
An Event Is a Member	358
The Delegate Type and EventHandler	358
Raising an Event	359
Subscribing to an Event	360
Removing Event Handlers	362
Adding Anonymous Method Event Handlers	363
Standard Event Usage	364
Using the EventArgs Class	364
Passing Data by Extending EventArgs	364
Using the Custom Delegate	365
The MyTimerClass Code	367
Event Accessors	369

CHAPTER 17	Interfaces	371
	What Is an Interface?	372
	Example Using the Comparable Interface	373
	Declaring an Interface	376
	Implementing an Interface	378
	Example with a Simple Interface	379
	An Interface Is a Reference Type	380
	Implementing Multiple Interfaces	382
	Implementing Interfaces with Duplicate Members	383
	References to Multiple Interfaces	384
	An Inherited Member As an Implementation	386
	Explicit Interface Member Implementations	388
	Accessing Explicit Interface Member Implementations	391
	Interfaces Can Inherit Interfaces	392
	Using the as Operator with Interfaces	393
	Example of Different Classes Implementing an Interface	394
CHAPTER 18	Conversions	397
	What Are Conversions?	398
	Implicit Conversions	399
	Explicit Conversions and Casting	400
	Casting	400
	Types of Conversions	402
	Numeric Conversions	402
	Implicit Numeric Conversions	403
	Overflow Checking Context	404
	Explicit Numeric Conversions	406
	Reference Conversions	410
	Implicit Reference Conversions	411
	Explicit Reference Conversions	413
	Valid Explicit Reference Conversions	414
	Boxing Conversions	416
	Boxing Creates a Copy	417
	Unboxing Conversions	418
	The Unboxing Conversions	419

User-Defined Conversions	420
Constraints on User-Defined Conversions	420
Example of a User-Defined Conversion	421
Evaluating User-Defined Conversions	423
Example of a Multi-Step User-Defined Conversion	423
The is Operator	425
The as Operator	426
 CHAPTER 19 Generics	 427
What Are Generics?	428
A Stack Example	428
Generics in C#	430
Continuing with the Example	431
Generic Classes	432
Declaring a Generic Class	433
Creating a Constructed Type	433
Creating Variables and Instances	435
The Stack Example Using Generics	437
Comparing the Generic and Non-Generic Stack	439
Constraints on Type Parameters	440
Where Clauses	441
Constraint Types and Order	442
Generic Structs	443
Generic Interfaces	444
An Example Using Generic Interfaces	445
Generic Interface Implementations Must Be Unique	446
Generic Delegates	447
Generic Methods	449
Declaring a Generic Method	449
Invoking a Generic Method	450
Example of a Generic Method	452
 CHAPTER 20 Enumerators and Iterators	 455
Enumerators and Enumerable Types	456
Using the foreach Statement	456
Types of Enumerators	457

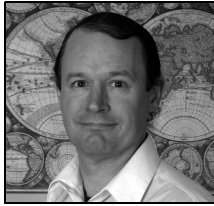
Using the IEnumerator Interface	458
Declaring an IEnumerator Enumerator	461
The IEnumerable Interface	463
Example Using IEnumerable and IEnumerator	464
The Non-Interface Enumerator	466
The Generic Enumeration Interfaces	468
The IEnumerator<T> Interface	469
The IEnumerable<T> Interface	472
Iterators	474
Iterator Blocks	475
More about Iterators	476
Producing Enumerables and Enumerators	477
Using an Iterator to Produce an Enumerable	478
Using an Iterator to Produce an Enumerator	479

CHAPTER 21 **Attributes** 481

What Is an Attribute?	482
A Quick Preview	483
The Stages of an Attribute	484
Applying an Attribute	485
Multiple Attributes	486
Other Targets	486
Global Attributes	487
Predefined Attributes	488
Using Custom Attributes	489
Declaring a Custom Attribute	489
Using Attribute Constructors	490
Specifying the Constructor	490
Using the Constructor	491
Positional and Named Parameters in Constructors	492
Restricting the Usage of an Attribute	494
Using Suggested Practices for Custom Attributes	496
Accessing an Attribute	496
Using the IsDefined Method	497
Using the GetCustomAttributes Method	498
Using Reserved Attributes	499
The Obsolete Attribute	499
The Conditional Attribute	499

CHAPTER 22	Preprocessor Directives	503
	What Are Preprocessor Directives?	504
	General Rules	504
	The #define and #undef Directives	506
	Conditional Compilation	507
	The Conditional Compilation Constructs	508
	Diagnostic Directives	511
	Line Number Directives	512
	Region Directives	513
	The #pragma warning Directive	514
CHAPTER 23	Other Topics	515
	Miscellaneous Topics	516
	Strings	516
	Using Class StringBuilder	517
	Formatting Numeric Strings	518
	Parsing Strings to Data Values	523
	Nullable Types	524
	Creating a Nullable Type	524
	Using Nullable User-Defined Types	529
	Method Main	531
	Documentation Comments	533
	Inserting Documentation Comments	534
	Nested Types	537
	Example of a Nested Class	538
	Visibility and Nested Types	539
INDEX		543

About the Author



DAN SOLIS holds a Bachelor of Arts in biology and English, and initially worked in research on the structure of metal crystals, until he found that he enjoyed programming much more than working in a lab. He also holds a Master of Science degree in computer science from the University of California at Santa Barbara, where he concentrated on programming languages and compiler design.

Dan has been programming professionally for more than 20 years, with more than half that time working as a consultant and contract programmer, including several projects for Microsoft Consulting Services. His consulting projects have ranged from programs for mutual fund analysis and supply chain management to systems for missile tracking. He has also taught courses on various programming languages, Windows programming, UNIX internals, and a number of other topics, in both the United States and Europe.

Dan's first programming language was C, but he soon became intrigued by the journal articles about a new language called "C with Classes." Eventually that language was renamed C++ and released to the world. He began using C++ as soon as he could get access to a compiler, and eventually started teaching training seminars on the language as well.

With the advent of C# and .NET, he has moved on to enjoying the myriad advantages of the new language and platform, and has been working with them enthusiastically ever since.

About the Technical Reviewer



■ **CHRISTOPHE NASARRE** is a development architect for Business Objects, a company that develops desktop and web-based business intelligence solutions. In his spare time, Christophe writes articles for *MSDN* magazine, *MSDN/Vista*, and *ASPToday*; and has been reviewing books on Win32, COM, MFC, .NET, and WPF since 1996.

Acknowledgments

I want to thank Sian for supporting and encouraging me on a daily basis, and I also want to thank my parents and family for their continued love and support.

I also want to express my sincere gratitude to the people at Apress who have worked with me to bring this book to fruition. I really appreciate that they understood and appreciated what I was trying to do, and worked with me to achieve it. Thanks to all of you.

Introduction

The purpose of this book is to teach you the fundamentals and mechanics of the C# programming language. Most books teach programming primarily using text. That's great for novels, but many of the important concepts of programming languages can best be understood through a combination of words, figures, and tables.

Many of us think visually, and figures and tables can help clarify and crystallize our understanding of a concept. In several years of teaching programming languages, I have found that the pictures I drew on the whiteboards were the things that most quickly helped the students understand the concepts I was trying to convey.

Illustrations alone, however, are not sufficient to explain a programming language and platform. The goal of this book is to find the best combination of words and illustrations to give you a thorough understanding of the language, and to allow the book to serve as a reference resource as well.

This book is written for anyone who wants an introduction to the C# programming language—from the novice to the seasoned programmer. For those just getting started in programming, I have included the basics. For seasoned programmers, the content is laid out succinctly, and in a form that will allow you to go directly to the information required without having to wade through oceans of words. For both sets of programmers, the content itself is presented graphically, in a form that should make the language easy to learn. Enjoy!

