

Accelerated C# 2008



Trey Nash

Accelerated C# 2008**Copyright © 2007 by Weldon W. Nash, III**

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): 978-1-59059-873-3

ISBN-10 (pbk): 1-59059-873-3

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: Dominic Shakeshaft

Technical Reviewer: Shawn Wildermuth

Editorial Board: Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick,

Jason Gilmore, Kevin Goff, Jonathan Hassell, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper,

Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Project Manager: Sofia Marchant

Copy Editor: Jim Compton

Assistant Production Director: Kari Brooks-Copony

Production Editor: Laura Cheu

Compositor: Jimmie Young

Proofreader: April Eddy

Indexer: Beth Palmer

Artist: April Milne

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 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. 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>.

This book is dedicated to the memory of my grandfather

*Weldon W. Nash, Sr.
December 19, 1912 – April 29, 2007*

*To Svetlana
for believing in me*

Contents at a Glance

Foreword	xvii
About the Author	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
Introduction	xxv
■ CHAPTER 1 C# Preview	1
■ CHAPTER 2 C# and the CLR	9
■ CHAPTER 3 C# Syntax Overview	17
■ CHAPTER 4 Classes, Structs, and Objects	39
■ CHAPTER 5 Interfaces and Contracts	123
■ CHAPTER 6 Overloading Operators	149
■ CHAPTER 7 Exception Handling and Exception Safety	163
■ CHAPTER 8 Working with Strings	195
■ CHAPTER 9 Arrays, Collection Types, and Iterators	221
■ CHAPTER 10 Delegates, Anonymous Functions, and Events	253
■ CHAPTER 11 Generics	279
■ CHAPTER 12 Threading in C#	317
■ CHAPTER 13 In Search of C# Canonical Forms	363
■ CHAPTER 14 Extension Methods	417
■ CHAPTER 15 Lambda Expressions	443
■ CHAPTER 16 LINQ: Language Integrated Query	465
■ APPENDIX References	495
■ INDEX	497

Contents

Foreword	xvii
About the Author	xix
About the Technical Reviewer	xxi
Acknowledgments	xxiii
Introduction	xxv
■ CHAPTER 1 C# Preview	1
Differences Between C# and C++	1
C# 1	1
C++	2
CLR Garbage Collection	2
Example of a C# Program	3
Overview of Features Added in C# 2.0	5
Overview of What's New in C# 3.0	6
Summary	7
■ CHAPTER 2 C# and the CLR	9
The JIT Compiler in the CLR	10
Assemblies and the Assembly Loader	11
Minimizing the Working Set of the Application	11
Naming Assemblies	12
Loading Assemblies	12
Metadata	13
Cross-Language Compatibility	14
Summary	15
■ CHAPTER 3 C# Syntax Overview	17
C# Is a Strongly Typed Language	17
Expressions	18
Statements and Expressions	19

Types and Variables	19
Value Types	21
Reference Types	24
Default Variable Initialization	25
Implicitly Typed Local Variables	25
Type Conversion	27
as and is Operators	29
Generics	31
Namespaces	32
Defining Namespaces	33
Using Namespaces	34
Control Flow	35
if-else, while, do-while, and for	35
switch	35
foreach	36
break, continue, goto, return, and throw	36
Summary	37

CHAPTER 4	Classes, Structs, and Objects	39
	Class Definitions	41
	Fields	41
	Constructors	44
	Methods	45
	Properties	47
	Encapsulation	51
	Accessibility	55
	Interfaces	56
	Inheritance	57
	sealed Classes	64
	abstract Classes	65
	Nested Classes	65
	Indexers	69
	partial Classes	70
	partial Methods	71
	Static Classes	72
	Reserved Member Names	74
	Value Type Definitions	75
	Constructors	76
	The Meaning of this	78
	Finalizers	80
	Interfaces	81
	Anonymous Types	81
	Object Initializers	84

Boxing and Unboxing	87
When Boxing Occurs	91
Efficiency and Confusion	93
System.Object	94
Equality and What It Means	95
The IComparable Interface	95
Creating Objects	95
The new Keyword	96
Field Initialization	97
Static (Class) Constructors	98
Instance Constructor and Creation Ordering	101
Destroying Objects	104
Finalizers	105
Deterministic Destruction	106
Exception Handling	107
Disposable Objects	107
The IDisposable Interface	107
The using Keyword	110
Method Parameter Types	111
Value Arguments	111
ref Arguments	112
out Parameters	113
param Arrays	114
Method Overloading	114
Inheritance and Virtual Methods	115
Virtual and Abstract Methods	115
override and new Methods	115
sealed Methods	118
A Final Few Words on C# Virtual Methods	118
Inheritance, Containment, and Delegation	118
Choosing Between Interface and Class Inheritance	119
Delegation and Composition vs. Inheritance	120
Summary	122

■ CHAPTER 5 **Interfaces and Contracts** 123

Interfaces Define Types	124
Defining Interfaces	125
What Can Be in an Interface?	125
Interface Inheritance and Member Hiding	126

Implementing Interfaces	128
Implicit Interface Implementation	128
Explicit Interface Implementation	129
Overriding Interface Implementations in Derived Classes	131
Beware of Side Effects of Value Types Implementing Interfaces	134
Interface Member Matching Rules	135
Explicit Interface Implementation with Value Types	139
Versioning Considerations	141
Contracts	142
Contracts Implemented with Classes	142
Interface Contracts	144
Choosing Between Interfaces and Classes	144
Summary	148
 CHAPTER 6 Overloading Operators	 149
Just Because You Can Doesn't Mean You Should	149
Types and Formats of Overloaded Operators	149
Operators Shouldn't Mutate Their Operands	150
Does Parameter Order Matter?	151
Overloading the Addition Operator	151
Operators That Can Be Overloaded	153
Comparison Operators	153
Conversion Operators	156
Boolean Operators	158
Summary	161
 CHAPTER 7 Exception Handling and Exception Safety	 163
How the CLR Treats Exceptions	163
Mechanics of Handling Exceptions in C#	164
Throwing Exceptions	164
Changes with Unhandled Exceptions Starting with .NET 2.0	164
Syntax Overview of the try Statement	165
Rethrowing Exceptions and Translating Exceptions	167
Exceptions Thrown in finally Blocks	169
Exceptions Thrown in Finalizers	170
Exceptions Thrown in Static Constructors	171
Who Should Handle Exceptions?	172
Avoid Using Exceptions to Control Flow	173
Achieving Exception Neutrality	173
Basic Structure of Exception-Neutral Code	174
Constrained Execution Regions	179
Critical Finalizers and SafeHandle	181

Creating Custom Exception Classes	185
Working with Allocated Resources and Exceptions	186
Providing Rollback Behavior	190
Summary	193

■ CHAPTER 8 Working with Strings 195

String Overview	195
String Literals	196
Format Specifiers and Globalization	197
Object.ToString, IFormattable, and CultureInfo	198
Creating and Registering Custom CultureInfo Types	199
Format Strings	201
Console.WriteLine and String.Format	202
Examples of String Formatting in Custom Types	203
ICustomFormatter	204
Comparing Strings	206
Working with Strings from Outside Sources	208
StringBuilder	209
Searching Strings with Regular Expressions	211
Searching with Regular Expressions	211
Searching and Grouping	213
Replacing Text with Regex	216
Regex Creation Options	219
Summary	220

■ CHAPTER 9 Arrays, Collection Types, and Iterators 221

Introduction to Arrays	221
Implicitly Typed Arrays	222
Type Convertibility and Covariance	224
Sortability and Searchability	225
Synchronization	225
Vectors vs. Arrays	226
Multidimensional Rectangular Arrays	228
Multidimensional Jagged Arrays	229
Collection Types	231
Comparing ICollection with ICollection	231
Collection Synchronization	232
Lists	233
Dictionaries	234
Sets	235
System.Collections.ObjectModel	235
Efficiency	238

IEnumerable <T>, IEnumerator <T>, IEnumerable, and IEnumerator	239
Types That Produce Collections	242
Iterators	242
Forward, Reverse, and Bidirectional Iterators	247
Collection Initializers	251
Summary	252

■ CHAPTER 10 Delegates, Anonymous Functions, and Events 253

Overview of Delegates.	253
Delegate Creation and Use	254
Single Delegate	255
Delegate Chaining	256
Iterating Through Delegate Chains	258
Unbound (Open Instance) Delegates.	259
Events.	262
Anonymous Methods.	266
Beware the Captured Variable Surprise.	270
Anonymous Methods as Delegate Parameter Binders.	272
The Strategy Pattern	276
Summary	278

■ CHAPTER 11 Generics 279

Difference Between Generics and C++ Templates	280
Efficiency and Type Safety of Generics.	281
Generic Type Definitions and Constructed Types	282
Generic Classes and Structs.	283
Generic Interfaces	285
Generic Methods	286
Generic Delegates	288
Generic Type Conversion	291
Default Value Expression	292
Nullable Types	293
Constructed Types Control Accessibility	295
Generics and Inheritance	295
Constraints	297
Constraints on Nonclass Types	301
Generic System Collections	302
Generic System Interfaces.	303
Select Problems and Solutions	305
Conversion and Operators Within Generic Types	305
Creating Constructed Types Dynamically	314
Summary	316

CHAPTER 12	Threading in C#	317
	Threading in C# and .NET	317
	Starting Threads	318
	The IOU Pattern and Asynchronous Method Calls	320
	States of a Thread	320
	Terminating Threads	323
	Halting Threads and Waking Sleeping Threads	324
	Waiting for a Thread to Exit	325
	Foreground and Background Threads	326
	Thread-Local Storage	327
	How Unmanaged Threads and COM Apartments Fit In	330
	Synchronizing Work Between Threads	331
	Lightweight Synchronization with the Interlocked Class	332
	Monitor Class	337
	Locking Objects	345
	Semaphore	350
	Events	350
	Win32 Synchronization Objects and WaitHandle	351
	Using ThreadPool	353
	Asynchronous Method Calls	354
	Timers	361
	Summary	362
 CHAPTER 13	 In Search of C# Canonical Forms	 363
	Reference Type Canonical Forms	363
	Default to sealed Classes	364
	Use the Non-Virtual Interface (NVI) Pattern	365
	Is the Object Cloneable?	368
	Is the Object Disposable?	373
	Does the Object Need a Finalizer?	375
	What Does Equality Mean for This Object?	382
	If You Override Equals, Override GetHashCode Too	388
	Does the Object Support Ordering?	391
	Is the Object Formattable?	393
	Is the Object Convertible?	397
	Prefer Type Safety at All Times	399
	Using Immutable Reference Types	403
	Value Type Canonical Forms	405
	Override Equals for Better Performance	406
	Do Values of This Type Support Any Interfaces?	410
	Implement Type-Safe Forms of Interface Members and Derived Methods	411

Summary	414
Checklist for Reference Types	414
Checklist for Value Types	415
 CHAPTER 14 Extension Methods	417
Introduction to Extension Methods	417
How Does the Compiler Find Extension Methods?	418
Under the Covers	421
Code Readability vs. Code Understandability	422
Recommendations for Use	423
Consider Extension Methods Over Inheritance	423
Isolate Extension Methods in Separate Namespace	424
Changing a Type's Contract Can Break Extension Methods	425
Transforms	425
Operation Chaining	429
Custom Iterators	430
Borrowing from Functional Programming	432
The Visitor Pattern	437
Summary	441
 CHAPTER 15 Lambda Expressions	443
Introduction to Lambda Expressions	443
Lambda Expressions	444
Lambda Statements	448
Expression Trees	449
Operating on Expressions	451
Functions As Data	452
Useful Applications of Lambda Expressions	453
Iterators and Generators Revisited	453
Closures (Variable Capture) and Memoization	456
Currying	460
Anonymous Recursion	462
Summary	464
 CHAPTER 16 LINQ: Language Integrated Query	465
A Bridge to Data	465
Query Expressions	466
Extension Methods and Lambda Expressions Revisited	468
Standard Query Operators	469

C# Query Keywords	470
The from Clause and Range Variables	470
The join Clause	472
The where Clause and Filters	474
The orderby Clause	474
The select Clause and Projection	475
The let Clause	476
The group Clause	478
The into Clause and Continuations	480
The Virtues of Being Lazy	482
C# Iterators Foster Laziness	482
Subverting Laziness	483
Executing Queries Immediately	485
Expression Trees Revisited	485
Techniques from Functional Programming	486
Custom Standard Query Operators and Lazy Evaluation	486
Replacing foreach Statements	493
Summary	494
 APPENDIX	
References	495
Blogs	496
 INDEX	497