Accelerated C# 2005

Trey Nash

Apress®

Accelerated C# 2005

Copyright © 2006 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): 978-1-59059-717-0

ISBN-10 (pbk): 1-59059-717-6

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: James Huddleston
Technical Reviewer: David Weller
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade
Project Manager: Richard Dal Porto
Copy Edit Manager: Nicole LeClerc
Copy Editor: Nicole Abramowitz
Assistant Production Director: Kari Brooks-Copony
Production Editor: Ellie Fountain
Compositor: Kinetic Publishing Services, LLC
Proofreader: April Eddy
Indexer: Brenda Miller
Artist: Kinetic Publishing Services, LLC
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 section.

Contents

Foreword xv
About the Author xvii
About the Technical Reviewer xix
Acknowledgments xxi
Introduction xxiii

CHAPTER 1 **C# Preview** 1

Differences Between C#, C++, and Java 1
- C# 1
- C++ and Java 2
- CLR Garbage Collection 2

Example of a C# Program 3
Overview of What's New in C# 2.0 4
Summary 5

CHAPTER 2 **C# and the CLR** 7

The JIT Compiler in the CLR 7
Assemblies and the Assembly Loader 9
- Minimizing the Working Set of the Application 9
- Naming Assemblies 10
- Loading Assemblies 10

Metadata 11
Cross-Language Compatibility 12
Summary 12

CHAPTER 3 **C# Syntax Overview** 15

C# Is a Strongly Typed Language 15
Expressions 16
Statements 17

Types and Variables 17
Value Types. 19
Reference Types. 21
Default Variable Initialization. 22
Type Conversion. 23
as and is Operators 24
Generics 26
Namespaces 27
Defining Namespaces 28
Using Namespaces 29
Control Flow 30
if-else, while, do-while, and for 30
switch 30
foreach 31
break, continue, goto, return, and throw 32
Summary 32

CHAPTER 4 Classes, Structs, and Objects 33

Class Definitions. 34
Fields. 35
Constructors 38
Methods 39
Properties 40
Encapsulation. 43
Accessibility 46
Interfaces 47
Inheritance 49
sealed Classes 55
abstract Classes. 56
Nested Classes. 57
Indexers 60
partial Classes 62
static Classes 63
Reserved Member Names 64
Value Type Definitions 65
Constructors. 66
The Meaning of this. 68
Finalizers. 70
Interfaces 71
Boxing and Unboxing 71
When Boxing Occurs 75
Efficiency and Confusion. 76

System.Object 77
Equality and What It Means 79
The IComparable Interface 79
Creating Objects 79
The new Keyword 79
Field Initialization 80
Static (Class) Constructor 82
Instance Constructor and Creation Ordering 84
Destroying Objects 88
Finalizers 88
Deterministic Destruction 90
Exception Handling 90
Disposable Objects 91
The IDisposable Interface 91
The using Keyword 93
Method Parameter Types 94
Value Arguments 95
ref Arguments 95
out Parameters 97
param Arrays 97
Method Overloading 98
Inheritance and Virtual Methods 98
Virtual and Abstract Methods 98
override and new Methods 99
sealed Methods 101
A Final Few Words on C# Virtual Methods 101
Inheritance, Containment, and Delegation 102
Choosing Between Interface and Class Inheritance 102
Delegation and Composition vs. Inheritance 103
Summary 105

CHAPTER 5 **Interfaces and Contracts** 107

Interfaces Define Types 108
Defining Interfaces 109
What Can Be in an Interface? 109
Interface Inheritance and Member Hiding 110
Implementing Interfaces 112
Implicit Interface Implementation 112
Explicit Interface Implementation 113
Overriding Interface Implementations in Derived Classes 114
Beware of Side Effects of Value Types Implementing Interfaces 118

Interface Member Matching Rules 119
Explicit Interface Implementation with Value Types 122
Versioning Considerations. 124
Contracts. 125
Contracts Implemented with Classes 125
Interface Contracts 127
Choosing Between Interfaces and Classes 128
Summary 132

CHAPTER 6 **Overloading Operators** 133

Just Because You Can Doesn’t Mean You Should. 133
Types and Formats of Overloaded Operators 133
Operators Shouldn’t Mutate Their Operands. 134
Does Parameter Order Matter?. 135
Overloading the Addition Operator 135
Operators That Can Be Overloaded 137
Comparison Operators 137
Conversion Operators 140
Boolean Operators. 142
Summary 145

CHAPTER 7 **Exception Handling and Exception Safety**. 147

Handling Exceptions 147
Avoid Using Exceptions to Control Flow. 148
Mechanics of Handling Exceptions in C#. 148
Throwing Exceptions. 148
Changes with Unhandled Exceptions in .NET 2.0. 149
Syntax Overview of the try Statement. 150
Rethrowing Exceptions and Translating Exceptions 152
Exceptions Thrown in finally Blocks. 154
Exceptions Thrown in Finalizers 154
Exceptions Thrown in Static Constructors. 155
Achieving Exception Neutrality 157
Basic Structure of Exception-Neutral Code. 157
Constrained Execution Regions 162
Critical Finalizers and SafeHandle 164
Creating Custom Exception Classes. 168
Working with Allocated Resources and Exceptions 169
Providing Rollback Behavior 173
Summary 176

CHAPTER 8 **Working with Strings** 177

String Overview 177
String Literals 178
Format Specifiers and Globalization 179
Object.ToString(), IFormattable, and CultureInfo 180
Creating and Registering Custom CultureInfo Types 181
Format Strings 182
Console.WriteLine() and String.Format() 183
Examples of String Formatting in Custom Types 184
ICustomFormatter 186
Comparing Strings 188
Working with Strings from Outside Sources 189
StringBuilder 191
Searching Strings with Regular Expressions 192
Searching with Regular Expressions 193
Searching and Grouping 194
Replacing Text with Regex 198
Regex Creation Options 200
Summary 201

CHAPTER 9 **Arrays, Collection Types, and Iterators** 203

Introduction to Arrays 203
Type Convertibility and Covariance 204
Sortability and Searchability 205
Synchronization 205
Vectors vs. Arrays 206
Multi-Dimensional Rectangular Arrays 207
Multi-Dimensional Jagged Arrays 209
Collection Types 210
Comparing ICollection<T> with ICollection 211
Collection Synchronization 212
Lists 213
Dictionaries 214
System.Collections.ObjectModel 214
Efficiency 217
IEnumerable<T>, IEnumerator<T>, IEnumerable, IEnumerator 218
Types That Produce Collections 221
Iterators 221
Forward, Reverse, and Bidirectional Iterators 226
Summary 230

CHAPTER 10 **Delegates, Anonymous Functions, and Events** 231

Overview of Delegates. 231

Delegate Creation and Use 232

Single Delegate 233

Delegate Chaining 234

Iterating Through Delegate Chains. 235

Unbound (Open Instance) Delegates 237

Events 239

Anonymous Methods. 243

Beware the Captured Variable Surprise. 247

Anonymous Methods As Delegate Parameter Binders 250

Strategy Pattern. 253

Summary 255

CHAPTER 11 **Generics**. 257

Introduction to Generics 257

Difference Between Generics and C++ Templates 258

Efficiency and Type Safety of Generics 259

Generic Type Placeholder Naming Conventions 260

Generic Type Definitions and Constructed Types 260

Generic Classes and Structs. 261

Generic Interfaces 263

Generic Methods 264

Generic Delegates 266

Generic Type Conversion. 269

Default Value Expression. 269

Nullable Types 271

Constructed Types Control Accessibility 273

Generics and Inheritance 273

Constraints 274

Constraints on Nonclass Types. 279

Generic System Collections 279

Generic System Interfaces 281

Select Problems and Solutions. 282

Conversion and Operators Within Generic Types 282

Creating Constructed Types Dynamically 291

Summary 293

CHAPTER 12 Threading in C#. 295

Threading in C# and .NET 295
Starting Threads. 296
The IOU Pattern and Asynchronous Method Calls 298
States of a Thread 298
Terminating Threads 301
Halting Threads and Waking Sleeping Threads. 302
Waiting for a Thread to Exit. 303
Foreground and Background Threads 304
Thread-Local Storage 305
How Unmanaged Threads and COM Apartments Fit In 308
Synchronizing Work Between Threads. 309
Lightweight Synchronization with the Interlocked Class 310
Monitor Class 315
Locking Objects 322
Events 326
Win32 Synchronization Objects and WaitHandle 327
Using ThreadPool 329
Asynchronous Method Calls 330
Timers. 336
Summary 338

CHAPTER 13 In Search of C# Canonical Forms 339

Reference Type Canonical Form. 339
Default to sealed Classes 340
Use the NVI Pattern 341
Is the Object Cloneable? 343
Is the Object Disposable? 349
Does the Object Need a Finalizer? 351
What Does Equality Mean for This Object? 357
If You Override Equals(), Override GetHashCode() Too 364
Does the Object Support Ordering? 366
Is the Object Formattable? 369
Is the Object Convertible? 372
Prefer Type Safety at All Times. 374
Using Immutable Reference Types. 378

Value Type Canonical Form 380
Override Equals() for Better Performance 381
Do Values of This Type Support Any Interfaces? 385
Implement Type-Safe Forms of Interface Members and Derived Methods 386
Summary 388
Checklist for Reference Types 389
Checklist for Value Types 390

APPENDIX **References** 391

Blogs 392

INDEX 393