C++/CLI

The Visual C++ Language for .NET

Gordon Hogenson

Apress®

C++/CLI: The Visual C++ Language for .NET

Copyright © 2006 by Gordon Hogenson

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: 978-1-59059-705-7

ISBN-10: 1-59059-705-2

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 Editors: Ewan Buckingham, James Huddleston
Technical Reviewer: Damien Watkins
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: Julie M. Smith
Copy Edit Manager: Nicole Flores
Copy Editor: Ami Knox
Assistant Production Director: Kari Brooks-Copony
Production Editor: Laura Cheu
Compositor: Susan Glinert Stevens
Proofreader: Elizabeth Berry
Indexer: John Collin
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/Download section.

Contents

Foreword by Stanley B. Lippman xv

Foreword by Herb Sutter xvii

About the Author xxv

About the Technical Reviewer xxvii

Acknowledgments xxix

Introduction xxxi

CHAPTER 1 **Introducing C++/CLI** 1

Garbage Collection and Handles 1

The /clr Compiler Option 3

The Virtual Machine 3

The Common Type System 3

Reference Types and Value Types 4

The CLI and the .NET Framework 5

“Hello, World” 5

Summary 10

CHAPTER 2 **A Quick Tour of the C++/CLI Language Features** 11

Primitive Types 11

Aggregate Types 12

Reference Classes 14

Value Classes 15

Enumeration Classes 17

Interface Classes 19

Elements Modeling the “has-a” Relationship 21

Properties 21

Delegates and Events. 23

Generics 27

Summary 28

CHAPTER 3 Building C++/CLI Programs for the .NET Developer Platform with Visual C++ 29

Targeting the .NET Developer Platform with Visual C++ 2005 29
Visual C++ 2005 Compilation Modes 30
Safe Mode (/clr:safe Compiler Option) 30
Pure Mode (/clr:pure Compiler Option). 30
Mixed Mode (/clr Compiler Option) 31
Managed Extensions Syntax (/clr:oldSyntax Compiler Option) 32
None of the Above 32
Caveats When Upgrading Code to Visual C++ 2005 32
Architecture Dependence and 64-bit Programming 32
Assemblies and Modules 33
The Assembly Manifest 33
Viewing Metadata with ILDasm.exe 34
The #using Directive 37
Referencing Assemblies and Access Control 39
Friend Assemblies 39
Assembly Attributes 40
The Linker and the Assembly Linker 40
Resources and Assemblies 41
Signed Assemblies 41
Multifile Assemblies 41
Summary 41

CHAPTER 4 Object Semantics in C++/CLI 43

Object Semantics for Reference Types 43
Object Semantics for Value Types 44
Implications of the Unified Type System 44
Implicit Boxing and Unboxing 45
Stack vs. Heap Semantics 47
Pitfalls of Delete and Stack Semantics. 51
The Unary % Operator and Tracking References 52
Dereferencing Handles 54
Copy Constructors 55
Lvalues, GC-lvalues, Rvalues, and GC-rvalues 56
auto_handle 58

Parameter Passing 60
Passing Reference Types by Value 63
Passing Value Types by Reference 65
Temporary Handles 66
Passing Value Types As Handles 68
Summary of Parameter-Passing Semantics 70
Do’s and Don’ts of Returning Values 70
Summary 73

CHAPTER 5 **Fundamental Types: Strings, Arrays, and Enums** 75

Strings 75
String Operators 79
Comparing Strings 80
Formatting Strings 81
Numeric String Formatting 82
StringBuilder 84
Conversions Between Strings and Other Data Types 85
Input/Output 86
Basic Output 86
Out, Error, and In 87
Basic Input with Console::ReadLine 87
Reading and Writing Files 87
Reading and Writing Strings 89
System::String and Other I/O Systems 90
Arrays 92
Initializing. 93
Array Length 95
Navigating Arrays 97
Differences Between Native and Managed Arrays 100
Arrays As Parameters. 101
Copying an Array. 102
Managed Array Class Members 103
Array Equality 106
Parameter Arrays 107
Arrays in Classes 108
Beyond Arrays: ArrayList 108

Enumerated Types 110
The Enum Class 111
Enumerated Types and Conversions 112
The Underlying Type of an Enum 112
The Flags Attribute 113
Enum Values As Strings 114
Summary 116

CHAPTER 6 Classes and Structs 117

Constructors and Initialization 118
Static Constructors 119
Copy Constructors for Reference and Value Types 121
Literal Fields 121
initonly Fields 124
Const Correctness 126
Properties, Events, and Operators 127
Example: A Scrabble Game 127
The this Pointer 153
Access Levels for Classes 156
Native and Managed Classes 157
Using a Native Object in a Managed Type 157
Class Destruction and Cleanup 160
Finalizers 161
Pitfalls of Finalizers. 168
Summary 171

CHAPTER 7 Features of a .NET Class 173

Properties 173
Using Indexed Properties 177
Delegates and Events 184
Asynchronous Delegates 188
Events 191
Event Receivers and Senders 199
Using the EventArgs Class 201
Reserved Names. 203
Operator Overloading 203
Static Operators 203
Conversion Operators and Casts. 206
Summary 210

CHAPTER 8 Inheritance 211

Name Collisions in Inheritance Hierarchies 212
Using the new Keyword on Virtual Functions 214
Using the override Keyword on Virtual Methods 215
Abstract Classes 219
Sealed Classes 220
Abstract and Sealed 221
Virtual Properties 222
Special Member Functions and Inheritance 225
Constructors 226
Virtual Functions in the Constructor 228
Destructors and Inheritance 231
Finalizers and Inheritance 232
Casting in Inheritance Hierarchies 233
Summary 234

CHAPTER 9 Interfaces 235

Interfaces vs. Abstract Classes 235
Declaring Interfaces 236
Interfaces Implementing Other Interfaces 237
Interfaces with Properties and Events 240
Interface Name Collisions 240
Interfaces and Access Control 244
Interfaces and Static Members 245
Literals in Interfaces 246
Commonly Used .NET Framework Interfaces 246
IComparable 246
IEnumerable and IEnumerator 248
Interfaces and Dynamically Loaded Types 255
Summary 257

CHAPTER 10 Exceptions, Attributes, and Reflection 259

Exceptions 259
The Exception Hierarchy 260
What’s in an Exception? 260
Creating Exception Classes 262
Using the Finally Block 263
Dealing with Exceptions in Constructors 265

Throwing Nonexception Types 266
Unsupported Features 268
Exception-Handling Best Practices 268
Exceptions and Errors from Native Code 269
Attributes 270
How Attributes Work 270
The Attribute Class 271
Attribute Parameters 271
Some Useful Attributes 271
Assembly and Module Attributes 276
Creating Your Own Attributes 277
Reflection 279
Application Domains 283
Summary 284

CHAPTER 11 Parameterized Functions and Types 285

Generics 285
Type Parameters 285
Generic Functions 286
Generic Types 288
Generic Collections 290
Using Constraints 296
Interface Constraints 296
Class Constraints 297
Reference Types and Value Types As Type Parameters 298
The gcnew Constraint 300
Value Type Constraints 301
Reference Type Constraints 303
Multiple Constraints 303
.NET Framework Container Types 304
Generic vs. Nongeneric Container Classes 304
Using the Collection Class Interfaces 305
ArrayList 305
Dictionaries 308
Managed Templates 309
Summary 316

CHAPTER 12 Interoperability 317

The Many Faces of Interop 317
Interoperating with Other .NET Languages 319
Using Native Libraries with Platform Invoke 322
Data Marshaling 327
Interop with COM 328
Using Native Libraries Without P/Invoke 329
Recompiling a Native Library As Managed Code 332
Interior Pointers 339
Pinning Pointers 340
Native Objects and Managed Objects 341
Using a Managed Object in a Native Class 342
Using a Native Object in a Managed Type 343
Native and Managed Entry Points 347
How to Avoid Double Thunking 348
Managed and Native Exceptions 348
Interop with Structured Exceptions (__try/__except). 348
Interop with Win32 Error Codes 351
Interop with C++ Exceptions. 352
Interop with COM HRESULTs. 354
Summary 354

APPENDIX Quick Reference 355

Keywords and Contextual Keywords 355
Whitespaced Keywords 356
Keywords As Identifiers 357
Detecting CLR Compilation 358
XML Documentation 359
Summary of Compilation Modes 362
Syntax Summary 363

INDEX 377