Beginning C# 2008

From Novice to Professional

SECOND EDITION

Christian Gross

Apress®

Beginning C# 2008: From Novice to Professional, Second Edition

Copyright © 2008 by Christian Gross

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): 978-1-4302-1033-7

ISBN-13 (electronic): 978-1-4302-1034-4

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: Ewan Buckingham
Technical Reviewer: Christian Kenyeres
Editorial Board: Editorial Board: Clay Andres, Steve Anglin, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Matthew Moodie, Joseph Ottinger, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Sofia Marchant
Developmental Editor: Ron Pronk
Copy Editor: Damon Larson
Associate Production Director: Kari Brooks-Copony
Production Editor: Laura Esterman
Compositors: Susan Glinert and Octal Publishing, Inc.
Proofreader: Lisa Hamilton
Indexer: Broccoli Information Management
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.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

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.

Some food for thought when writing software:

"A common mistake that people make when trying to design something completely foolproof is to underestimate the ingenuity of complete fools."

"The major difference between a thing that might go wrong and a thing that cannot possibly go wrong is that when a thing that cannot possibly go wrong goes wrong it usually turns out to be impossible to get at or repair."

—*Douglas Adams,* Mostly Harmless

Contents at a Glance

About the Author xvii
About the Technical Reviewer xix
Introduction xxi

■CHAPTER 1 **Ready, Steady, Go!** 1
■CHAPTER 2 **.NET Number and Value Types** 27
■CHAPTER 3 **String Manipulations** 51
■CHAPTER 4 **Data Structures, Decisions, and Loops** 75
■CHAPTER 5 **C# Exception Handling** 113
■CHAPTER 6 **Object-Oriented Programming Basics** 131
■CHAPTER 7 **Components and Object Hierarchies** 159
■CHAPTER 8 **Component-Oriented Architecture** 193
■CHAPTER 9 **Lists, Delegates, and Lambda Expressions** 225
■CHAPTER 10 **All About Persistence** 251
■CHAPTER 11 **Exploring .NET Generics** 289
■CHAPTER 12 **Application Configuration and Dynamic Loading** 323
■CHAPTER 13 **All About Multithreading** 353
■CHAPTER 14 **Using Relational Databases** 383
■CHAPTER 15 **Larning About LINQ** 409
■CHAPTER 16 **Writing Functional Code in C#** 433
■CHAPTER 17 **C# Odds and Ends** 449

■**INDEX** 465

Contents

About the Author xvii
About the Technical Reviewer xix
Introduction xxi

CHAPTER 1 **Ready, Steady, Go!** 1

Downloading and Installing the Tools 1
- Downloading Visual C# Express 2

Choosing the Application Type 4
Creating Projects and Solutions 4
Creating the Windows Application 5
- Viewing the Source Code 6
- Renaming the Solution 6
- Saving the Solution 7
- Running the Windows Application 7
- Making the Windows Application Say Hello 9
- Adding Comments to the Application 11

Navigating the User Controls of the Solution 13
Creating the Console Application 15
- Adding a Console Application Project to the Solution 16
- Making the Console Application Say Hello 16
- Setting the Startup Project 16
- Running the Console Project 17

Creating the Class Library 17
- Adding a Class Library Project to the Solution 17
- Moving Functionality 18
- Defining References 19
- Calling Class Library Functionality 19
- Using Variables and Constants 21

Understanding How the .NET Framework Works 23
The Important Stuff to Remember 25
Some Things for You to Do 26

CHAPTER 2 .NET Number and Value Types 27

Focusing and Organizing Your Development 27
Organizing the Calculator. 28
Focusing the Calculator 29
Implementing the Class Library 32
Writing the Add() Method 34
Writing Code to Test the Add() Method 36
Understanding Problems with Numeric Values and Types 41
Understanding Numeric and Value Data Types 43
Understanding Value and Reference Types 43
Understanding the CLR Numeric Types 44
Finishing the Calculator. 48
The Important Stuff to Remember 49
Some Things for You to Do 49

CHAPTER 3 String Manipulations 51

Organizing the Translation Application 51
Building the Translator Application 52
Creating the Translator Class 53
Translating Hello 53
Creating the Test Application 54
Answering the Question of Responsibility 55
Investigating the String Type. 55
Solving the Extra Whitespace Problem. 61
Quoting Strings 65
Character Mapping 66
Dealing with Languages and Cultures. 67
Setting Culture and Language in Windows 67
Parsing and Processing Numbers. 68
Working with Cultures 71
The Important Stuff to Remember 73
Some Things for You to Do 74

CHAPTER 4 Data Structures, Decisions, and Loops 75

Understanding the Depth-First Search Algorithm 75
Implementing User-Defined Types 78
Declaring Structs and Classes 78
Value Type Constraints. 79

Organizing the Search Algorithm 85
Writing the Depth-First Search Code 87
Defining and Implementing the Data Structure 87
Defining the Algorithm Test 97
Implementing the Depth-First Search Algorithm 101
Running the Depth-First Search Algorithm 109
The Important Stuff to Remember 110
Some Things for You to Do 111

CHAPTER 5 C# Exception Handling 113

Understanding Errors, Exceptions, and Exception Handling 113
Running the Debugger 114
Handling Exceptions 115
Catching Exceptions 115
Implementing Exception Handlers 118
Safeguarding Against Stack Unwinding 121
Filtering Exceptions 122
Writing Exception-Safe Code 125
Writing Defensive Code 125
Using Default State 127
Processing Errors That Are Warnings 128
The Important Stuff to Remember 129
Some Things for You to Do 130

CHAPTER 6 Object-Oriented Programming Basics 131

Understanding Currency Spreads 132
Organizing the Currency Exchange Application 133
Writing Tests for the Currency Exchange Application 133
Getting Started with Structural Code 133
Understanding Base Classes 134
Understanding Inheritance 135
Using C# Properties 137
Understanding Inheritance and Scope Modifiers 140
Handling Verification 143
Finishing the Base Class 146
Writing the Active Trader and Hotel Trader Currency Converters 147
Implementing ActiveCurrencyTrader 147
Implementing HotelCurrencyTrader 150

Learning More About Preprocessor Directives, Properties, and Abstract Methods 152
More Preprocessor Directive Details 152
More Property Scope Details 154
More abstract Keyword Details 154
The Important Stuff to Remember 156
Some Things for You to Do 157

CHAPTER 7 Components and Object Hierarchies 159

Understanding Some Basic Tax Concepts 159
Organizing the Tax Application 160
Programming Using Ideas 161
Representing Ideas Using C# Interfaces 162
Understanding How Inheritance and Components Work 164
Implementing a Tax Calculation Engine 170
Defining the Interfaces 170
Implementing a Base Class Tax Calculation Engine 171
Using Default Implementations 176
Implementing a Base Tax Account 177
Using the Base Functionality of the Tax Calculation Engine 179
Implementing a Tax Calculation Engine and Tax Account 179
Using the Tax Calculation Engine 184
More About Inheritance and Type Casting 184
More Inheritance Details 184
More Type-Casting Details 190
The Important Stuff to Remember 191
Some Things for You to Do 192

CHAPTER 8 Component-Oriented Architecture 193

Understanding Kernels 193
Organizing the Lighting Application 194
Building the Kernel 195
Defining the Interfaces 196
Implementing the Kernel 200
Defining the Kernel As an Interface Instead of a Class 216
Building a Complete Application 218
Defining Some Rooms 218
Instantiating PublicRoom and PrivateRoom 219

Learning More About Private Classes and Object Initialization 221
Private Classes 221
Object Initialization with Nested Data Types 221
The Important Stuff to Remember 223
Some Things for You to Do 223

CHAPTER 9 Lists, Delegates, and Lambda Expressions 225

Managing Collections 225
Managing a Collection Before C# 2.0 226
Managing a Collection After C# 2.0 230
The Case of the Code That Feels Wrong 231
Using Delegates 235
Using Anonymous Methods 239
Multicasting with Delegates 240
Using Lambda Expressions 241
Understanding Lambda Expressions 243
Creating the Algorithm 244
Implementing the Algorithm Using Lambda Expressions 245
More About Collection Types 246
Using a Plain-Vanilla List 246
Using a Key/Value Pair List 247
Using a Stack 248
Using a Queue 248
The Important Stuff to Remember 249
Some Things for You to Do 249

CHAPTER 10 All About Persistence 251

Organizing the Lottery-Prediction System 251
Piping Data Using a Console 252
Reading Data from the Console 252
Building a Shell 254
Implementing the TextProcessor Application 263
Piping Binary Data 272
Defining the Interfaces and Implementing the Shell 273
Defining the Type 276
Converting a Text Stream to a Binary Stream 278
Converting a Binary Stream to a Text Stream 279

Tweaking Serialization 280
Performing Custom Serialization. 281
Declaring a Data Member As Nonserializable 282
Separating Data Objects from Action Objects 282
Completing Custom Types 283
Implementing GetHashCode() 283
Implementing Equals() 286
The Important Stuff to Remember 287
Some Things for You to Do 288

CHAPTER 11 **Exploring .NET Generics** 289

Why Use .NET Generics? 289
The Theory of a Server-Side Spreadsheet 292
Architecting a Server-Side Spreadsheet 295
Designing the Architecture 295
Defining the Server Spreadsheet Interfaces 296
Implementing the Server Spreadsheet 305
Using Lambda Expressions in the Spreadsheet 310
Assigning State Without Knowing the Type 314
Overriding the ToString() Functionality 316
Iterating Data Using Enumerators 317
Using the Spreadsheet 318
Calculating an Average 318
Understanding Why the Calculations Work 320
The Important Stuff to Remember 321
Some Things for You to Do 322

CHAPTER 12 **Application Configuration and Dynamic Loading** 323

Convention over Configuration 323
Decoupling Using a Configuration Architecture. 325
Decoupling Using a Convention Architecture. 325
Setting Up the Dynamic Loading Projects 326
Signing an Assembly 327
Setting the Output Path 328
Defining and Processing a Configuration File 329
Creating an XML-Based Configuration File 330
Adding the Dynamic Loading Configuration Items 332
Reading a Configuration File 332

Dynamically Loading an Assembly 333
Dynamically Instantiating a Type 333
Enhancing the Configuration File 337
Loading a Strongly Named Assembly 342
Relocating a Strongly Named Assembly to the GAC 343
Using Version Numbers 345
Implementing a Convention-Based Architecture 348
Dynamically Loading Base Class or Interface Types 350
The Important Stuff to Remember 350
Some Things for You to Do 351

CHAPTER 13 **All About Multithreading** 353

Understanding Multitasking 353
Preemptive Multitasking 354
Understanding Time Slicing. 354
Using Threads 357
Creating a New Thread 357
Waiting for the Thread to End 358
Creating a Thread with State. 359
Synchronizing Between Threads. 361
How to Prevent Your Code from Deadlocking 366
Implementing a Reader/Writer Threaded Architecture 370
Implementing a Producer/Consumer Architecture. 374
Using a Hidden Producer/Consumer Implementation. 374
Implementing a Generic Producer/Consumer Architecture 376
Using an Asynchronous Approach 378
The Important Stuff to Remember 380
Some Things for You to Do 381

CHAPTER 14 **Using Relational Databases** 383

Understanding Relational Databases. 383
Relational Database Tables 384
Database Relations. 385
Accessing Relational Databases 388
Designing a Database Using Visual C# Express 389
Configuring the Data Source 390
Adding the Tables. 393

Accessing the Database Using ADO.NET 396
Connecting to a Database 397
Adding Table Data 397
Selecting Data from a Table 399
Deleting Data from the Database 400
Closing a Database Connection 401
Recapping ADO.NET Usage 401
Using the Dataset Designer 401
Building Relations Between Tables. 402
Using the Generated Code 406
The Important Stuff to Remember 407
Some Things for You to Do 408

CHAPTER 15 Learning About LINQ 409

Finding the Frequency of Winning Numbers. 409
Extending the Lottery-Prediction System. 410
Implementing a Frequency Solution 413
Learning More LINQ Tricks 419
Selecting and Altering Data 423
Selecting with Anonymous Types 424
Processing Multiple Streams. 425
Sorting the Results 426
Splitting Results into Groups 427
Performing Set Operations. 428
Using LINQ in Other Contexts 430
The Important Stuff to Remember 432
Some Things for You to Do 432

CHAPTER 16 Writing Functional Code in C# 433

Why Functional Programming? 433
The Essence of Functional Programming 435
Higher-Order Functions 436
Pure Functions 439
Function Evaluation 444
Recursion. 446
The Important Stuff to Remember 447
Some Things for You to Do 448

CHAPTER 17 C# Odds and Ends 449

Operators 449
Using Arithmetic Operators 449
Overloading the Operators. 454
The goto Statement. 456
Understanding .NET Generics Constraints 457
Using the type Constraint. 457
Using the new Constraint. 459
Using the class Constraint. 460
Nullable Types 460
Partial Classes and Methods 462
The Important Stuff to Remember 464
Some Things for You to Do. 464

INDEX 465

About the Author

Many people say that by looking at a person's dog, you can tell what the person is like. Well, the picture is of my dog Louys, an English bulldog. And yes, my English bulldog and I have many common characteristics.

But what about the biography of the author, **CHRISTIAN GROSS**? It's pretty simple: I'm a guy who has spent oodles of time strapped to a chair debugging and taking apart code. In fact, I really enjoy this business we call software development. I have loved it ever since I learned how to peek and poke my first bytes. I have written various books, including *Ajax and REST Recipes: A Problem-Solution Approach, Foundations of Object-Oriented Programming Using .NET 2.0 Patterns*, and *A Programmer's Introduction to Windows DNA*, all available from Apress.

These days, I enjoy coding and experimenting with .NET, as it is a fascinating environment. .NET makes me feel like a kid opening a present on Christmas morning. You had an idea what the gift was, but you were not completely sure. And with .NET, there is no relative giving you socks or a sweater. It's excitement all the way!

About the Technical Reviewer

CHRISTIAN KENYERES, principal architect for Collaborative Consulting, is a visionary technology professional with more than 15 years of extensive information technology experience. He has served numerous high-profile clients as an enterprise architect and boasts a broad range of technical and business knowledge.

Prior to joining Collaborative, Christian performed consulting for various companies such as Compaq, EMC, Fidelity Investments, Liberty Mutual Insurance, and John Hancock. He holds B.S. and M.S. degrees in Computer Science from the University of Massachusetts and Boston University, respectively.

Introduction

The first computer programming book I read was entitled *Programming Windows 3.0* by Charles Petzold. This was around the time when Microsoft Windows 3.0 (circa 1992) once and for all showed the industry that Microsoft was a company with a future. Writing code for Windows back then was complicated by many things: lack of documentation, 16-bit architecture, and the necessity of buying a compiler separate from the software development kit (SDK). Charles's book tied everything together and solved the problem of how to write a program for Windows.

Now the problems are quite the opposite: we have too much documentation, we have 64-bit architectures, and everything including the kitchen sink is thrown into a development environment. Now we need to figure out what we actually need. We have too many options—too many ways to solve the same problem. What I am trying to do with this book is the same thing that Charles did for me when I first started out, and that was to help me figure out what I needed to write code.

This book is about explaining the C# programming language in the context of solving problems. C# has become a sophisticated programming language that can achieve many goals, but you are left wondering what techniques to use when. This book is here to answer your questions.

This book is not a reference to all of the features of the C# programming language. I don't explain the esoteric C# features. I stick to the C# programming features that you will use day in and day out. That does not mean that you will be missing certain C# programming language constructs, because I have covered all of the major features.

To get the full benefit of this book, I suggest that you do the exercises at the end of the chapters. The answers are available on the Apress web site (`http://www.apress.com`), and you can cheat and not do the exercises, but I advise against that.

If you are a beginning programmer who has no clue about C#, and you read this book and do the exercises, I am almost entirely sure that you will be a solid and knowledgeable C# programmer by the end of the book. If that sounds like a big promise, well, yes it is. The chapter text is intended to get you acquainted with the C# programming language and how to apply its features. The exercises are intended to make sure you actually understand the C# programming language and its features.

The chapter exercises are challenging. They cannot be solved within a few minutes. In fact, when I did all of the exercises, it took me five working-hour days to do all of them!

If you have any questions, such as, "So what was he trying to get at with that exercise?" I am available on Skype with the user ID `christianhgross`. Please don't just ring me. First chat using text, and if necessary, we can have a voice conversation. Also, you can send e-mail to me at `christianhgross@gmail.com`.

Thanks and good luck.