

Microsoft

Microsoft Visual C# 2012

John Sharp



ebook+exercises

Step by Step



Microsoft Visual C# 2012 Step by Step

Your hands-on, step-by-step guide to the fundamentals of Visual C# development.

Teach yourself how to build applications with Microsoft Visual C# 2012 and Visual Studio® 2012—one step at a time. Ideal for those with fundamental programming skills, this tutorial provides practical, learn-by-doing exercises for mastering core C# language features and creating working applications and components for Windows®.

Discover how to:

- Work with variables, statements, operators, and methods
- Write robust code that can catch and handle exceptions
- Respond to user input and gestures
- Handle events arising from multiple sources
- Manipulate data sets using arrays and collections
- Establish new data types by using classes, interfaces, and structures
- Use LINQ expressions to enumerate data
- Optimize processing with tasks and asynchronous operations
- Build your first Windows Store app

Your *Step by Step* digital content includes:

- Downloadable practice files
See <http://go.microsoft.com/fwlink/?Linkid=273785>
- Fully searchable online edition of this book—with unlimited access on the web. *Free online account required; see inside book.*

About the Author

John Sharp is an expert on developing applications with the Microsoft .NET Framework and interoperability issues. He has coauthored guides for the Microsoft Patterns and Practices group, and is the author of *Microsoft Visual C# 2010 Step by Step* and *Microsoft Windows Communication Foundation Step by Step*.

DEVELOPER ROADMAP

Start Here!

- Beginner-level instruction
- Easy-to-follow explanations and examples
- Exercises to build your first projects



Step by Step

- For experienced developers learning a new topic
- Focus on fundamental techniques and tools
- Hands-on tutorial with practice files plus eBook



Developer Reference

- Professional developers: intermediate to advanced
- Expertly covers essential tools and techniques
- Features extensive, adaptable code examples



Focused Topics

- For programmers who develop complex or specialized solutions
- Specialized topics, more in-depth coverage
- Features extensive, adaptable code examples



microsoft.com/mspress

U.S.A. \$44.99
Canada \$47.99
(Recommended)

Programming/Microsoft Visual C#

Microsoft

Microsoft® Visual C#® 2012 Step by Step

John Sharp

Published with the authorization of Microsoft Corporation by:
O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, California 95472

Copyright © 2012 by CM Group, Ltd.
All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN: 978-0-7356-6801-0

1 2 3 4 5 6 7 8 9 QG 7 6 5 4 3 2

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, O'Reilly Media, Inc., Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions and Development Editor: Russell Jones

Production Editor: Rachel Steely

Editorial Production: Zyg Group, LLC

Technical Reviewer: John Mueller

Copyeditor: Nicole Flores

Indexer: BIM Indexing Services

Cover Design: Twist Creative

Cover Composition: Zyg Group, LLC

Illustrator: Rebecca Demarest

I dedicate this book to Diana, my wife and fellow Warwickshire supporter, for keeping me sane and giving me the perfect excuse to spend time watching cricket.

—JOHN SHARP

Contents at a Glance

Introduction

xxi

PART I **INTRODUCING MICROSOFT VISUAL C# AND MICROSOFT VISUAL STUDIO 2012**

CHAPTER 1	Welcome to C#	3
CHAPTER 2	Working with Variables, Operators, and Expressions	39
CHAPTER 3	Writing Methods and Applying Scope	67
CHAPTER 4	Using Decision Statements	95
CHAPTER 5	Using Compound Assignment and Iteration Statements	115
CHAPTER 6	Managing Errors and Exceptions	137

PART II **UNDERSTANDING THE C# OBJECT MODEL**

CHAPTER 7	Creating and Managing Classes and Objects	165
CHAPTER 8	Understanding Values and References	189
CHAPTER 9	Creating Value Types with Enumerations and Structures	213
CHAPTER 10	Using Arrays	233
CHAPTER 11	Understanding Parameter Arrays	257
CHAPTER 12	Working with Inheritance	271
CHAPTER 13	Creating Interfaces and Defining Abstract Classes	295
CHAPTER 14	Using Garbage Collection and Resource Management	325

PART III **DEFINING EXTENSIBLE TYPES WITH C#**

CHAPTER 15	Implementing Properties to Access Fields	349
CHAPTER 16	Using Indexers	371
CHAPTER 17	Introducing Generics	389
CHAPTER 18	Using Collections	419
CHAPTER 19	Enumerating Collections	441
CHAPTER 20	Decoupling Application Logic and Handling Events	457
CHAPTER 21	Querying In-Memory Data by Using Query Expressions	491
CHAPTER 22	Operator Overloading	515

PART IV BUILDING PROFESSIONAL WINDOWS 8 APPLICATIONS WITH C#

CHAPTER 23	Improving Throughput by Using Tasks	541
CHAPTER 24	Improving Response Time by Performing Asynchronous Operations	585
CHAPTER 25	Implementing the User Interface for a Windows Store App	627
CHAPTER 26	Displaying and Searching for Data in a Windows Store App	681
CHAPTER 27	Accessing a Remote Database in a Windows Store App	733
	<i>Index</i>	775

Contents

Introduction *xxi*

**PART I INTRODUCING MICROSOFT VISUAL C#
 AND MICROSOFT VISUAL STUDIO 2012**

Chapter 1	Welcome to C#	3
	Beginning Programming with the Visual Studio 2012 Environment	3
	Writing Your First Program	8
	Using Namespaces	14
	Creating a Graphical Application	18
	Examining the Windows Store App	30
	Examining the WPF Application	33
	Adding Code to the Graphical Application	34
	Summary	37
	Chapter 1 Quick Reference	38
Chapter 2	Working with Variables, Operators, and Expressions	39
	Understanding Statements	39
	Using Identifiers	40
	Identifying Keywords	40
	Using Variables	42
	Naming Variables	42
	Declaring Variables	42

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Working with Primitive Data Types	43
Unassigned Local Variables	44
Displaying Primitive Data Type Values	44
Using Arithmetic Operators	52
Operators and Types	52
Examining Arithmetic Operators	54
Controlling Precedence	59
Using Associativity to Evaluate Expressions	60
Associativity and the Assignment Operator	60
Incrementing and Decrementing Variables	61
Prefix and Postfix	62
Declaring Implicitly Typed Local Variables	63
Summary	64
Chapter 2 Quick Reference	65

Chapter 3 Writing Methods and Applying Scope 67

Creating Methods	67
Declaring a Method	68
Returning Data from a Method	69
Calling Methods	71
Applying Scope	74
Defining Local Scope	74
Defining Class Scope	75
Overloading Methods	76
Writing Methods	76
Using Optional Parameters and Named Arguments	85
Defining Optional Parameters	86
Passing Named Arguments	87
Resolving Ambiguities with Optional Parameters and Named Arguments	87

Summary	92
Chapter 3 Quick Reference	93
Chapter 4 Using Decision Statements	95
Declaring Boolean Variables	95
Using Boolean Operators	96
Understanding Equality and Relational Operators	96
Understanding Conditional Logical Operators	97
Short-Circuiting	98
Summarizing Operator Precedence and Associativity	98
Using if Statements to Make Decisions	99
Understanding if Statement Syntax	99
Using Blocks to Group Statements	100
Cascading if Statements	101
Using switch Statements	107
Understanding switch Statement Syntax	108
Following the switch Statement Rules	109
Summary	113
Chapter 4 Quick Reference	114
Chapter 5 Using Compound Assignment and Iteration Statements	115
Using Compound Assignment Operators	115
Writing while Statements	117
Writing for Statements	123
Understanding for Statement Scope	125
Writing do Statements	125
Summary	134
Chapter 5 Quick Reference	135

Chapter 6	Managing Errors and Exceptions	137
	Coping with Errors	137
	Trying Code and Catching Exceptions	138
	Unhandled Exceptions	139
	Using Multiple catch Handlers	140
	Catching Multiple Exceptions	141
	Propagating Exceptions	147
	Using Checked and Unchecked Integer Arithmetic	149
	Writing Checked Statements	150
	Writing Checked Expressions	151
	Throwing Exceptions	154
	Using a finally Block	159
	Summary	160
	Chapter 6 Quick Reference	161

PART II UNDERSTANDING THE C# OBJECT MODEL

Chapter 7	Creating and Managing Classes and Objects	165
	Understanding Classification	165
	The Purpose of Encapsulation	166
	Defining and Using a Class	166
	Controlling Accessibility	168
	Working with Constructors	169
	Overloading Constructors	170
	Understanding static Methods and Data	180
	Creating a Shared Field	181
	Creating a static Field by Using the const Keyword	182
	Understanding static Classes	182
	Anonymous Classes	185
	Summary	186
	Chapter 7 Quick Reference	187

Chapter 8 Understanding Values and References 189

Copying Value Type Variables and Classes189

Understanding Null Values and Nullable Types195

 Using Nullable Types196

 Understanding the Properties of Nullable Types197

Using ref and out Parameters198

 Creating ref Parameters199

 Creating out Parameters200

How Computer Memory Is Organized202

 Using the Stack and the Heap203

The System.Object Class204

Boxing205

Unboxing206

Casting Data Safely207

 The is Operator207

 The as Operator208

Summary210

Chapter 8 Quick Reference210

Chapter 9 Creating Value Types with Enumerations and Structures 213

Working with Enumerations213

 Declaring an Enumeration214

 Using an Enumeration214

 Choosing Enumeration Literal Values215

 Choosing an Enumeration's Underlying Type216

Working with Structures218

 Declaring a Structure220

 Understanding Structure and Class Differences221

 Declaring Structure Variables222

 Understanding Structure Initialization223

 Copying Structure Variables227

Summary	231
Chapter 9 Quick Reference	232

Chapter 10 Using Arrays 233

Declaring and Creating an Array	233
Declaring Array Variables	234
Creating an Array Instance	234
Populating and Using an Array	235
Creating an Implicitly Typed Array	236
Accessing an Individual Array Element	237
Iterating Through an Array	238
Passing Arrays as Parameters and Return Values for a Method	239
Copying Arrays	241
Using Multidimensional Arrays	242
Creating Jagged Arrays	243
Summary	254
Chapter 10 Quick Reference	255

Chapter 11 Understanding Parameter Arrays 257

Overloading: A Recap	257
Using Array Arguments	258
Declaring a params Array	260
Using params object[]	262
Using a params Array	263
Comparing Parameter Arrays and Optional Parameters	266
Summary	268
Chapter 11 Quick Reference	269

Chapter 12 Working with Inheritance 271

What Is Inheritance?271

Using Inheritance272

 The System.Object Class Revisited274

 Calling Base Class Constructors274

 Assigning Classes276

 Declaring new Methods277

 Declaring virtual Methods279

 Declaring override Methods280

 Understanding protected Access282

Understanding Extension Methods288

Summary293

Chapter 12 Quick Reference293

Chapter 13 Creating Interfaces and Defining Abstract Classes 295

Understanding Interfaces295

 Defining an Interface296

 Implementing an Interface297

 Referencing a Class Through Its Interface298

 Working with Multiple Interfaces299

 Explicitly Implementing an Interface300

 Interface Restrictions302

 Defining and Using Interfaces302

Abstract Classes312

 Abstract Methods314

Sealed Classes314

 Sealed Methods315

 Implementing and Using an Abstract Class315

Summary322

Chapter 13 Quick Reference323

**Chapter 14 Using Garbage Collection
and Resource Management 325**

The Life and Times of an Object	325
Writing Destructors	326
Why Use the Garbage Collector?	328
How Does the Garbage Collector Work?	330
Recommendations	330
Resource Management	331
Disposal Methods	331
Exception-Safe Disposal	332
The using Statement and the IDisposable Interface	332
Calling the Dispose Method from a Destructor	334
Implementing Exception-Safe Disposal	336
Summary	345
Chapter 14 Quick Reference	345

PART III DEFINING EXTENSIBLE TYPES WITH C#

**Chapter 15 Implementing Properties
to Access Fields 349**

Implementing Encapsulation by Using Methods	349
What Are Properties?	351
Using Properties	354
Read-Only Properties	354
Write-Only Properties	355
Property Accessibility	355
Understanding the Property Restrictions	356
Declaring Interface Properties	358
Replacing Methods with Properties	359
Generating Automatic Properties	363
Initializing Objects by Using Properties	365
Summary	369
Chapter 15 Quick Reference	369

Chapter 16 Using Indexers 371

What Is an Indexer?371

 An Example That Doesn't Use Indexers371

 The Same Example Using Indexers373

 Understanding Indexer Accessors.376

 Comparing Indexers and Arrays376

Indexers in Interfaces378

Using Indexers in a Windows Application379

Summary.386

Chapter 16 Quick Reference386

Chapter 17 Introducing Generics 389

The Problem with the object Type.....389

The Generics Solution393

 Generics vs. Generalized Classes395

 Generics and Constraints396

Creating a Generic Class396

 The Theory of Binary Trees.....396

 Building a Binary Tree Class by Using Generics399

Creating a Generic Method.409

 Defining a Generic Method to Build a Binary Tree410

Variance and Generic Interfaces.....412

 Covariant Interfaces.....414

 Contravariant Interfaces.....415

Summary.....417

Chapter 17 Quick Reference418

Chapter 18 Using Collections 419

What Are Collection Classes?419

 The List<T> Collection Class.....421

 The LinkedList<T> Collection Class423

 The Queue<T> Collection Class.....425

 The Stack<T> Collection Class426

The Dictionary<TKey, TValue> Collection Class	427
The SortedList<TKey, TValue> Collection Class	428
The HashSet<T> Collection Class	429
Using Collection Initializers	431
The Find Methods, Predicates, and Lambda Expressions	431
Comparing Arrays and Collections	433
Using Collection Classes to Play Cards	434
Summary	438
Chapter 18 Quick Reference	439
Chapter 19 Enumerating Collections	441
Enumerating the Elements in a Collection	441
Manually Implementing an Enumerator	443
Implementing the IEnumerable Interface	447
Implementing an Enumerator by Using an Iterator	450
A Simple Iterator	450
Defining an Enumerator for the Tree<TItem> Class by Using an Iterator	452
Summary	454
Chapter 19 Quick Reference	455
Chapter 20 Decoupling Application Logic and Handling Events	457
Understanding Delegates	458
Examples of Delegates in the .NET Framework Class Library	459
The Automated Factory Scenario	461
Implementing the Factory Control System Without Using Delegates	461
Implementing the Factory by Using a Delegate	462
Declaring and Using Delegates	465
Lambda Expressions and Delegates	474
Creating a Method Adapter	474
The Forms of Lambda Expressions	475

Enabling Notifications with Events.	476
Declaring an Event.	477
Subscribing to an Event.	478
Unsubscribing from an Event.	478
Raising an Event.	478
Understanding User Interface Events	479
Using Events	480
Summary.	487
Chapter 20 Quick Reference	488

Chapter 21 Querying In-Memory Data by Using Query Expressions 491

What Is Language-Integrated Query?.	491
Using LINQ in a C# Application	492
Selecting Data.	494
Filtering Data	497
Ordering, Grouping, and Aggregating Data	497
Joining Data	500
Using Query Operators.	501
Querying Data in Tree<TItem> Objects	503
LINQ and Deferred Evaluation.	509
Summary.	513
Chapter 21 Quick Reference	513

Chapter 22 Operator Overloading 515

Understanding Operators	515
Operator Constraints.	516
Overloaded Operators	516
Creating Symmetric Operators	518
Understanding Compound Assignment Evaluation	520
Declaring Increment and Decrement Operators.	520
Comparing Operators in Structures and Classes.	521
Defining Operator Pairs.	522

Implementing Operators	523
Understanding Conversion Operators	530
Providing Built-in Conversions.	530
Implementing User-Defined Conversion Operators	531
Creating Symmetric Operators, Revisited	532
Writing Conversion Operators.	533
Summary.	535
Chapter 22 Quick Reference	536

**PART IV BUILDING PROFESSIONAL WINDOWS 8 APPLICATIONS
WITH C#**

Chapter 23 Improving Throughput by Using Tasks 541

Why Perform Multitasking by Using Parallel Processing?	541
The Rise of the Multicore Processor	542
Implementing Multitasking with the .NET Framework.	544
Tasks, Threads, and the ThreadPool	544
Creating, Running, and Controlling Tasks	545
Using the Task Class to Implement Parallelism.	548
Abstracting Tasks by Using the Parallel Class	559
When Not to Use the Parallel Class.	564
Canceling Tasks and Handling Exceptions	566
The Mechanics of Cooperative Cancellation.	566
Using Continuations with Canceled and Faulted Tasks	581
Summary.	582
Chapter 23 Quick Reference	583

**Chapter 24 Improving Response Time by Performing
Asynchronous Operations 585**

Implementing Asynchronous Methods.	586
Defining Asynchronous Methods: The Problem.	586
Defining Asynchronous Methods: The Solution.	589
Defining Asynchronous Methods That Return Values.	595
Asynchronous Methods and the Windows Runtime APIs.	596

Using PLINQ to Parallelize Declarative Data Access	599
Using PLINQ to Improve Performance While Iterating Through a Collection	600
Canceling a PLINQ Query	604
Synchronizing Concurrent Access to Data	605
Locking Data	608
Synchronization Primitives for Coordinating Tasks	608
Cancelling Synchronization	611
The Concurrent Collection Classes	612
Using a Concurrent Collection and a Lock to Implement Thread-Safe Data Access	612
Summary	623
Chapter 24 Quick Reference	624

Chapter 25 Implementing the User Interface for a Windows Store App 627

What Is a Windows Store App?	628
Using the Blank App Template to Build a Windows Store App	632
Implementing a Scalable User Interface	634
Applying Styles to a User Interface	669
Summary	679
Chapter 25 Quick Reference	679

Chapter 26 Displaying and Searching for Data in a Windows Store App 681

Implementing the Model-View-ViewModel Pattern	681
Displaying Data by Using Data Binding	682
Modifying Data by Using Data Binding	689

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Using Data Binding with a ComboBox Control.	693
Creating a ViewModel.	695
Adding Commands to a ViewModel.	699
Windows 8 Contracts	711
Implementing the Search Contract.	712
Navigating to a Selected Item.	722
Starting an Application from the Search Charm.	725
Summary.	729
Chapter 26 Quick Reference	732

Chapter 27 Accessing a Remote Database in a Windows Store App 733

Retrieving Data from a Database.	733
Creating an Entity Model	735
Creating and Using a Data Service	741
Inserting, Updating, and Deleting Data in a Database.	754
Performing Insert, Update, and Delete Operations Through a WCF Data Service.	754
Reporting Errors and Updating the User Interface	764
Summary.	772
Chapter 27 Quick Reference	773
<i>Index</i>	775

Introduction

Microsoft Visual C# is a powerful but simple language aimed primarily at developers creating applications by using the Microsoft .NET Framework. It inherits many of the best features of C++ and Microsoft Visual Basic, but few of the inconsistencies and anachronisms, resulting in a cleaner and more logical language. C# 1.0 made its public debut in 2001. The advent of C# 2.0 with Visual Studio 2005 saw several important new features added to the language, including generics, iterators, and anonymous methods. C# 3.0, which was released with Visual Studio 2008, added extension methods, lambda expressions, and most famously of all, the Language-Integrated Query facility, or LINQ. C# 4.0, released in 2010, provided further enhancements that improved its interoperability with other languages and technologies. These features included support for named and optional arguments, and the *dynamic* type, which indicates that the language runtime should implement late binding for an object. An important addition in the .NET Framework released concurrently with C# 4.0 was the classes and types that constitute the Task Parallel Library (TPL). Using the TPL, you can build highly scalable applications that can take full advantage of multicore processors quickly and easily. C# 5.0 adds native support for asynchronous task-based processing through the *async* method modifier and the *await* operator.

Another key event for Microsoft has been the launch of Windows 8. This new version of Windows supports highly interactive applications that can share data and collaborate with each other as well as connect to services running in the cloud. The development environment provided by Microsoft Visual Studio 2012 makes all these powerful features easy to use, and the many new wizards and enhancements included in Visual Studio 2012 can greatly improve your productivity as a developer. The combination of Visual Studio 2012, Windows 8, and C# 5.0 provides a comprehensive platform and toolset for building the next generation of powerful, intuitive, and portable applications. However, even if you are not using Windows 8, Visual Studio 2012 and C# 5.0 have much to offer, and they form an invaluable partnership for helping you to build great solutions.

Who Should Read This Book

This book assumes that you are a developer who wants to learn the fundamentals of programming with C# by using Visual Studio 2012 and the .NET Framework version 4.5. By the time you complete this book, you will have a thorough understanding of C# and

will have used it to build responsive and scalable Windows Presentation Foundation (WPF) applications that can run on both Windows 7 and Windows 8.

You can build and run C# 5.0 applications on Windows 7 and Windows 8, although the user interfaces provided by these two operating systems have some significant differences. Consequently, Parts I to III of this book provide exercises and worked examples that will run in both environments. Part IV focuses on the application development model used by Windows 8, and the material in this section provides an introduction to building interactive applications for this new platform.

Who Should Not Read This Book

This book is aimed at developers new to C#, and as such, it concentrates primarily on the C# language. This book is not intended to provide detailed coverage of the multitude of technologies available for building enterprise-level applications for Windows, such as ADO.NET, ASP.NET, Windows Communication Foundation, or Workflow Foundation. If you require more information on any of these items, you might consider reading some of the other titles in the Step by Step for Developers series available from Microsoft Press, such as *Microsoft ASP.NET 4 Step by Step*, *Microsoft ADO.NET 4 Step by Step*, and *Microsoft Windows Communication Foundation 4 Step by Step*.

Organization of This Book

This book is divided into four sections:

- Part I, "Introducing Microsoft Visual C# and Microsoft Visual Studio 2012," provides an introduction to the core syntax of the C# language and the Visual Studio programming environment.
- Part II, "Understanding the C# Object Model," goes into detail on how to create and manage new types by using C#, and how to manage the resources referenced by these types.
- Part III, "Defining Extensible Types with C#," includes extended coverage of the elements that C# provides for building types that you can reuse across multiple applications.
- Part IV, "Building Professional Window 8 Applications with C#," describes the Windows 8 programming model, and how you can use C# to build interactive applications for this new model.

- [read By Heresies Distressed online](#)
- [download online *Le Roman vrai*](#)
- [read online *Interaction Flow Modeling Language: Model-Driven UI Engineering of Web and Mobile Apps with IFML \(The MK/OMG Press Series\)*](#)
- [read The Moral Foundation of Economic Behavior](#)
- [read online Comment c'est pdf, azw \(kindle\), epub, doc, mobi](#)

- <http://korplast.gr/lib/By-Heresies-Distressed.pdf>
- <http://betsy.wesleychapelcomputerrepair.com/library/A-Book-of-Abstract-Algebra--2nd-Edition---Dover-Books-on-Mathematics-.pdf>
- <http://betsy.wesleychapelcomputerrepair.com/library/Interaction-Flow-Modeling-Language--Model-Driven-UI-Engineering-of-Web-and-Mobile-Apps-with-IFML--The-MK-OMG-Pr>
- <http://www.experienceolvera.co.uk/library/La-conciencia-uncida-a-la-carne--Diarios-de-madurez-1964-1980--Journals-of-Susan-Sontag--Book-2-.pdf>
- <http://www.1973vision.com/?library/Comment-c-est.pdf>