



Wrox Programmer to Programmer™

Beginning

# C# 7 Programming with Visual Studio® 2017

Benjamin Perkins, Jacob Vibe Hammer, Jon D. Reid



BEGINNING

# C# 7 Programming with Visual Studio® 2017

Benjamin Perkins  
Jacob Vibe Hammer  
Jon D. Reid



# Beginning C# 7 Programming with Visual Studio® 2017

Published by  
John Wiley & Sons, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
[www.wiley.com](http://www.wiley.com)

Copyright © 2018 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-119-45868-5  
ISBN: 978-1-119-45872-2 (ebk)  
ISBN: 978-1-119-45866-1 (ebk)  
Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2018933383

**Trademarks:** Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Visual Studio is a registered trademark of Microsoft Corporation. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

## ABOUT THE AUTHORS

**BENJAMIN PERKINS** (MBA, MCSD, ITIL) is currently employed at Microsoft in Munich, Germany, as a Senior Escalation Engineer. He has been working professionally in the IT industry for over two decades. He started computer programming with QBasic at the age of 11 on an Atari 1200XL desktop computer. He takes pleasure in the challenges that troubleshooting technical issues has to offer and savors the rewards of a well-written program. After completing high school he joined the United States Army. After successfully completing his military service, he attended Texas A&M University in College Station, Texas, where he received a Bachelor of Business Administration in Management Information Systems.

His roles in the IT industry have spanned the entire spectrum including programmer, system architect, technical support engineer, team leader, and mid-level management. While employed at Hewlett-Packard, he received numerous awards, degrees, and certifications. He has a passion for technology and customer service and looks forward to troubleshooting and writing more world-class technical solutions.

“My approach is to write code with support in mind, and to write it once correctly and completely so we do not have to come back to it again, except to enhance it.”

Benjamin is married to Andrea and has two wonderful children, Lea and Noa.

**JACOB VIBE HAMMER** helps develop solutions for the health care industry as a Senior Software Engineer at Systematic in Denmark. He started programming just about the time when he was able to spell the word “BASIC”—which, incidentally, is the first programming language he ever used. Since then, he has worked with numerous programming languages and solution architectures; however, since the turn of the century, he has worked primarily with the .NET platform. Today, his programming time is spent working primarily with C# and WPF, as well as toying with NoSQL databases. A Danish citizen, Jacob lives in Aarhus, Denmark, with his wife and two sons.

**JON D. REID** is a Product Solution Manager for IFS Field Service Management ([www.IFSWORLD.com](http://www.IFSWORLD.com)). He has coauthored a number of books, including *Beginning Visual C# 2015*, *Fast Track C#*, *Pro Visual Studio .NET*, and many others.

## ABOUT THE TECHNICAL EDITOR

**JOHN MUELLER** is a freelance author and technical editor. He has writing in his blood, having produced 108 books and more than 600 articles to date. The topics range from networking to artificial intelligence and from database management to heads-down programming. Some of his current books include topics such as Python for beginners, Python for data scientists, and Amazon Web Services. He has also written about algorithms and machine learning. His technical editing skills have helped more than 70 authors refine the content of their manuscripts. John has provided technical editing services to a number of computing magazines. Be sure to read John’s blog at <http://blog.johnmuelเลอร์books.com/>.

# CREDITS

**SENIOR ACQUISITIONS EDITOR**

Kenyon Brown

**PROJECT EDITOR**

Tom Dinse

**TECHNICAL EDITOR**

John Mueller

**PRODUCTION EDITOR**

Barath Kumar Rajasekaran

**COPY EDITOR**

First Edition Publishing Services

**PRODUCTION MANAGER**

Katie Wisor

**MANAGER OF CONTENT ENABLEMENT  
AND OPERATIONS**

Pete Gaughan

**MARKETING MANAGER**

Christie Hilbrich

**BUSINESS MANAGER**

Amy Knies

**PROJECT COORDINATOR, COVER**

Brent Savage

**PROOFREADER**

Nancy Bell

**INDEXER**

Johnna VanHoose Dinse

**COVER DESIGNER**

Wiley

**COVER IMAGE**

©Ben Clift Williams/EyeEm/Getty Images

# ACKNOWLEDGMENTS

It takes a lot of work to get content into a presentable format for students and IT professionals to read and get value from. The authors indeed have technical knowledge and experiences to share, but without the technical writers, technical reviewers, developers, editors, publishers, graphic designers, the list goes on, providing their valuable input, a book of high quality could not be written. The rate of change occurs too quickly for an individual to perform all these tasks and still publish a book that is valid before the technology becomes stale. This is why authors worked together with a great team to get all the components of the book together quickly. It was done to ensure that the most up to date information gets to the reader while the features are still fresh and current. I would like to thank Tom Dinse for his great project management and technical review of the content as well as John Mueller for his technical review and suggestions throughout the process. Lastly, I would like to thank all the numerous people behind the scenes who helped get this book together.





# CONTENTS

INTRODUCTION

xxi

## PART I: THE C# LANGUAGE

<b>CHAPTER 1: INTRODUCING C#</b>	<b>3</b>
<b>What Is the .NET Framework?</b>	<b>4</b>
What's in the .NET Framework?	4
.NET Standard and .NET Core	5
Writing Applications Using the .NET Framework and .NET Core	5
CIL and JIT	6
Assemblies	7
Managed Code	7
Garbage Collection	7
Fitting It Together	8
Linking	9
<b>What Is C#?</b>	<b>9</b>
Applications You Can Write with C#	10
C# in this Book	11
<b>Visual Studio 2017</b>	<b>11</b>
Visual Studio 2017 Products	11
Solutions	12
<b>CHAPTER 2: WRITING A C# PROGRAM</b>	<b>15</b>
<b>The Visual Studio 2017 Development Environment</b>	<b>16</b>
<b>Console Applications</b>	<b>21</b>
The Solution Explorer	24
The Properties Window	25
The Error List Window	25
<b>Desktop Applications</b>	<b>26</b>
<b>CHAPTER 3: VARIABLES AND EXPRESSIONS</b>	<b>33</b>
<b>Basic C# Syntax</b>	<b>34</b>
<b>Basic C# Console Application Structure</b>	<b>36</b>
<b>Variables</b>	<b>38</b>
Simple Types	38

Variable Naming	43
Literal Values	43
Binary Literals and Digit Separators	44
String Literals	45
<b>Expressions</b>	<b>46</b>
Mathematical Operators	47
Assignment Operators	52
Operator Precedence	53
Namespaces	54
<b>CHAPTER 4: FLOW CONTROL</b>	<b>59</b>
<b>Boolean Logic</b>	<b>60</b>
Boolean Bitwise and Assignment Operators	62
Operator Precedence Updated	64
<b>Branching</b>	<b>65</b>
The Ternary Operator	65
The if Statement	65
Checking More Conditions Using if Statements	68
The switch Statement	69
<b>Looping</b>	<b>72</b>
do Loops	73
while Loops	75
for Loops	77
Interrupting Loops	78
Infinite Loops	79
<b>CHAPTER 5: MORE ABOUT VARIABLES</b>	<b>83</b>
<b>Type Conversion</b>	<b>84</b>
Implicit Conversions	84
Explicit Conversions	86
Explicit Conversions Using the Convert Commands	88
<b>Complex Variable Types</b>	<b>91</b>
Enumerations	91
Defining Enumerations	92
Structs	96
Defining Structs	96
Arrays	99
Declaring Arrays	99
foreach Loops	102

---

Pattern Matching with switch case expression	102
Multidimensional Arrays	106
Arrays of Arrays	108
String Manipulation	109
<b>CHAPTER 6: FUNCTIONS</b>	<b>117</b>
<hr/>	
Defining and Using Functions	118
Return Values	120
Parameters	122
Parameter Matching	124
Parameter Arrays	124
Reference and Value Parameters	126
Out Parameters	129
Tuples	130
Variable Scope	131
Variable Scope in Other Structures	134
Parameters and Return Values versus Global Data	136
Local Functions	137
The Main() Function	138
Struct Functions	141
Overloading Functions	142
Using Delegates	144
<b>CHAPTER 7: DEBUGGING AND ERROR HANDLING</b>	<b>149</b>
<hr/>	
Debugging in Visual Studio	150
Debugging in Nonbreak (Normal) Mode	150
Outputting Debugging Information	151
Tracepoints	156
Diagnostics Output Versus Tracepoints	158
Debugging in Break Mode	158
Entering Break Mode	158
Monitoring Variable Content	162
Stepping through Code	164
Immediate and Command Windows	166
The Call Stack Window	167
Error Handling	167
try...catch...finally	168
Throw Expressions	175
Listing and Configuring Exceptions	176

---

<b>CHAPTER 8: INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING</b>	<b>179</b>
<b>What Is Object-Oriented Programming?</b>	<b>180</b>
What Is an Object?	181
Properties and Fields	182
Methods	183
Everything's an Object	184
The Life Cycle of an Object	184
Constructors	184
Destructors	185
Static and Instance Class Members	185
Static Constructors	185
Static Classes	186
<b>OOP Techniques</b>	<b>186</b>
Interfaces	187
Disposable Objects	188
Inheritance	188
Polymorphism	191
Interface Polymorphism	192
Relationships between Objects	193
Containment	193
Collections	194
Operator Overloading	194
Events	195
Reference Types versus Value Types	195
<b>OOP in Desktop Applications</b>	<b>196</b>
<b>CHAPTER 9: DEFINING CLASSES</b>	<b>203</b>
<b>Class Definitions in C#</b>	<b>204</b>
Interface Definitions	206
<b>System.Object</b>	<b>209</b>
<b>Constructors and Destructors</b>	<b>211</b>
Constructor Execution Sequence	212
<b>OOP Tools in Visual Studio</b>	<b>216</b>
The Class View Window	216
The Object Browser	218
Adding Classes	219
Class Diagrams	220
<b>Class Library Projects</b>	<b>222</b>

---

Interfaces Versus Abstract Classes	226
Struct Types	228
Shallow Copying Versus Deep Copying	230
<b>CHAPTER 10: DEFINING CLASS MEMBERS</b>	<b>233</b>
<hr/>	
<b>Member Definitions</b>	<b>234</b>
Defining Fields	234
Defining Methods	235
Defining Properties	236
Tuple Deconstruction	241
Refactoring Members	242
Automatic Properties	243
<b>Additional Class Member Topics</b>	<b>244</b>
Hiding Base Class Methods	244
Calling Overridden or Hidden Base Class Methods	246
The this Keyword	246
Using Nested Type Definitions	247
<b>Interface Implementation</b>	<b>249</b>
Implementing Interfaces in Classes	250
Explicit Interface Member Implementation	251
Additional Property Accessors	252
<b>Partial Class Definitions</b>	<b>252</b>
<b>Partial Method Definitions</b>	<b>253</b>
<b>Example Application</b>	<b>255</b>
Planning the Application	255
The Card Class	255
The Deck Class	255
Writing the Class Library	256
Adding the Suit and Rank Enumerations	257
Adding the Card Class	259
Adding the Deck Class	260
A Client Application for the Class Library	263
<b>The Call Hierarchy Window</b>	<b>265</b>
<b>CHAPTER 11: COLLECTIONS, COMPARISONS, AND CONVERSIONS</b>	<b>269</b>
<hr/>	
<b>Collections</b>	<b>270</b>
Using Collections	271
Defining Collections	276
Indexers	277

Adding a Cards Collection to CardLib	279
Keyed Collections and IDictionary	282
Iterators	283
Iterators and Collections	288
Deep Copying	289
Adding Deep Copying to CardLib	290
<b>Comparisons</b>	<b>292</b>
Type Comparisons	292
Boxing and Unboxing	292
The is Operator	294
Pattern Matching with the is Operator Pattern Expression	297
Value Comparisons	298
Operator Overloading	298
Adding Operator Overloads to CardLib	302
The IComparable and IComparer Interfaces	308
Sorting Collections	309
<b>Conversions</b>	<b>313</b>
Overloading Conversion Operators	313
The as Operator	315
<b>CHAPTER 12: GENERICS</b>	<b>319</b>
<hr/>	
<b>What Are Generics?</b>	<b>320</b>
<b>Using Generics</b>	<b>321</b>
Nullable Types	321
Operators and Nullable Types	322
The ?? Operator	323
The ?. Operator	324
Working with Nullable Types	325
The System.Collections.Generic Namespace	329
List<T>	330
Sorting and Searching Generic Lists	331
Dictionary<K, V>	337
Modifying CardLib to Use a Generic Collection Class	339
<b>Defining Generic Types</b>	<b>339</b>
Defining Generic Classes	340
The default Keyword	342
Constraining Types	342
Inheriting from Generic Classes	348
Generic Operators	349
Generic Structs	350
Defining Generic Interfaces	350

---

Defining Generic Methods	351
Defining Generic Delegates	352
<b>Variance</b>	<b>353</b>
Covariance	354
Contravariance	354
<b>CHAPTER 13: ADDITIONAL C# TECHNIQUES</b>	<b>359</b>
<hr/>	
<b>The :: Operator and the Global Namespace Qualifier</b>	<b>360</b>
<b>Custom Exceptions</b>	<b>361</b>
Adding Custom Exceptions to CardLib	362
<b>Events</b>	<b>363</b>
What Is an Event?	363
Handling Events	365
Defining Events	368
Multipurpose Event Handlers	371
The EventHandler and Generic EventHandler<T> Types	374
Return Values and Event Handlers	374
Anonymous Methods	375
<b>Expanding and Using CardLib</b>	<b>375</b>
<b>Attributes</b>	<b>384</b>
Reading Attributes	384
Creating Attributes	385
<b>Initializers</b>	<b>386</b>
Object Initializers	387
Collection Initializers	389
<b>Type Inference</b>	<b>392</b>
<b>Anonymous Types</b>	<b>394</b>
<b>Dynamic Lookup</b>	<b>398</b>
The dynamic Type	399
<b>Advanced Method Parameters</b>	<b>402</b>
Optional Parameters	403
Optional Parameter Values	404
The OptionalAttribute Attribute	404
Optional Parameter Order	404
Named Parameters	404
<b>Lambda Expressions</b>	<b>409</b>
Anonymous Methods Recap	409
Lambda Expressions for Anonymous Methods	410
Lambda Expression Parameters	413
Lambda Expression Statement Bodies	414
Lambda Expressions as Delegates and Expression Trees	415
Lambda Expressions and Collections	416

**PART II: WINDOWS PROGRAMMING**

<b>CHAPTER 14: BASIC DESKTOP PROGRAMMING</b>	<b>425</b>
<b>XAML</b>	<b>426</b>
Separation of Concerns	427
XAML in Action	427
Namespaces	428
Code-Behind Files	429
<b>The Playground</b>	<b>429</b>
WPF Controls	430
Properties	432
Dependency Properties	435
Attached Properties	435
Events	436
Handling Events	437
Routed Events	438
Routed Commands	438
Control Types	441
<b>Control Layout</b>	<b>441</b>
Basic Layout Concepts	441
Stack Order	442
Alignment, Margins, Padding, and Dimensions	442
Border	443
Visual Debugging Tools	443
Layout Panels	444
Canvas	444
DockPanel	446
StackPanel	448
WrapPanel	449
Grid	449
<b>The Game Client</b>	<b>452</b>
The About Window	453
Designing the User Interface	453
The Image Control	454
The Label Control	454
The TextBlock Control	454
The Button Control	455
The Options Window	458
The TextBox Control	459
The CheckBox Control	459
The RadioButton Control	460



---

The ComboBox Control	461
The TabControl	462
Handling Events in the Options Window	465
Data Binding	467
The DataContext	468
Binding to Local Objects	468
Static Binding to External Objects	469
Dynamic Binding to External Objects	470
Starting a Game with the ListBox Control	472
<b>CHAPTER 15: ADVANCED DESKTOP PROGRAMMING</b>	<b>479</b>
<hr/>	
Creating and Styling Controls	480
Styles	480
Templates	481
Triggers	483
Animations	484
WPF User Controls	485
Implementing Dependency Properties	486
The Main Window	499
The Menu Control	499
Routed Commands with Menus	499
Putting It All Together	504
Refactoring the Domain Model	504
The ViewModel	511
Completing the Game	519
<hr/>	
<b>PART III: CLOUD AND CROSS-PLATFORM PROGRAMMING</b>	
<hr/>	
<b>CHAPTER 16: BASIC CLOUD PROGRAMMING</b>	<b>533</b>
<hr/>	
The Cloud, Cloud Computing, and the Cloud Optimized Stack	534
Cloud Patterns and Best Practices	537
Using Microsoft Azure C# Libraries to Create a Storage Container	538
Creating an ASP.NET 4.7 Web Site That Uses the Storage Container	548
<b>CHAPTER 17: ADVANCED CLOUD PROGRAMMING AND DEPLOYMENT</b>	<b>559</b>
<hr/>	
Creating an ASP.NET Web API	560
Deploying and Consuming an ASP.NET Web API on Microsoft Azure	564
Scaling an ASP.NET Web API on Microsoft Azure	572

---

<b>CHAPTER 18: .NET STANDARD AND .NET CORE</b>	<b>579</b>
<b>Cross-Platform Basics and Key “Must Know” Terms</b>	<b>581</b>
<b>What Is .NET Standard, and Why Is It Needed?</b>	<b>583</b>
Shared Project, PCL, and .NET Standard	584
<b>Referencing and Targeting Frameworks</b>	<b>587</b>
<b>What is .NET Core?</b>	<b>588</b>
Cross Platform	590
Open Source	591
Optimized for the Cloud	591
Performance	592
Modular Design	593
Self-Contained Deployment Model	595
<b>Building and Packaging a .NET Standard Library</b>	<b>596</b>
<b>Building a .NET Core Application with Visual Studio</b>	<b>602</b>
<b>Porting from .NET Framework to .NET Core</b>	<b>605</b>
Identifying Third-Party Dependencies	605
Understanding Which Features Are Not Available	606
Upgrading the Current .NET Framework Target	606
Choosing the Platforms to Target for the Program	606
<b>CHAPTER 19: ASP.NET AND ASP.NET CORE</b>	<b>609</b>
<b>Overview of Web Applications</b>	<b>610</b>
<b>Which ASP.NET to Use and Why</b>	<b>611</b>
ASP.NET Web Forms	613
ASP.NET MVC	614
ASP.NET Web API	617
ASP.NET Core	617
ASP.NET Web Site versus ASP.NET Web Application	
Project Types	618
File Structure	619
Compiling	619
Deployment	619
<b>Using ASP.NET Web Forms</b>	<b>620</b>
Server Controls	620
Input Validation	621
State Management	622
Authentication and Authorization	623
<b>Creating ASP.NET Core Web Applications</b>	<b>627</b>
IIS and Kestrel	628
Razor Syntax	628

---

Input Validation	629
State Management	630
Authentication and Authorization	631
Dependency Injection	632

## **PART IV: DATA ACCESS**

---

### **CHAPTER 20: FILES** **641**

---

<b>File Classes for Input and Output</b>	<b>642</b>
The File and Directory Classes	643
The FileInfo Class	644
The DirectoryInfo Class	646
Path Names and Relative Paths	646
<b>Streams</b>	<b>647</b>
Classes for Using Streams	647
The FileStream Object	648
File Position	650
Reading Data	650
Writing Data	653
The StreamWriter Object	655
The StreamReader Object	658
Reading Data	660
Asynchronous File Access	660
Reading and Writing Compressed Files	661
<b>Monitoring the File System</b>	<b>664</b>

---

### **CHAPTER 21: XML AND JSON** **673**

---

<b>XML Basics</b>	<b>674</b>
<b>JSON Basics</b>	<b>674</b>
<b>XML Schemas</b>	<b>675</b>
<b>XML Document Object Model</b>	<b>677</b>
The XmlDocument Class	678
The XmlElement Class	678
Changing the Values of Nodes	683
Inserting New Nodes	684
Deleting Nodes	687
Selecting Nodes	689
<b>Converting XML to JSON</b>	<b>689</b>
<b>Searching XML with XPath</b>	<b>691</b>

---

<b>CHAPTER 22: LINQ</b>	<b>697</b>
<b>LINQ to XML</b>	<b>698</b>
LINQ to XML Functional Constructors	698
Working with XML Fragments	701
<b>LINQ Providers</b>	<b>704</b>
<b>LINQ Query Syntax</b>	<b>705</b>
Declaring a Variable for Results Using the var Keyword	706
Specifying the Data Source: from Clause	707
Specify Condition: where Clause	707
Selecting Items: select Clause	708
Finishing Up: Using the foreach Loop	708
Deferred Query Execution	708
<b>LINQ Method Syntax</b>	<b>709</b>
LINQ Extension Methods	709
Query Syntax versus Method Syntax	709
Lambda Expressions	710
<b>Ordering Query Results</b>	<b>712</b>
<b>Understanding the orderby Clause</b>	<b>713</b>
<b>Querying a Large Data Set</b>	<b>714</b>
<b>Using Aggregate Operators</b>	<b>717</b>
<b>Using the Select Distinct Query</b>	<b>720</b>
<b>Ordering by Multiple Levels</b>	<b>723</b>
<b>Using Group Queries</b>	<b>725</b>
<b>Using Joins</b>	<b>727</b>
<b>CHAPTER 23: DATABASES</b>	<b>731</b>
<b>Using Databases</b>	<b>731</b>
<b>Installing SQL Server Express</b>	<b>732</b>
<b>Entity Framework</b>	<b>732</b>
<b>A Code First Database</b>	<b>733</b>
<b>But Where Is My Database?</b>	<b>740</b>
<b>Navigating Database Relationships</b>	<b>742</b>
<b>Handling Migrations</b>	<b>749</b>
<b>Creating and Querying XML from an Existing Database</b>	<b>750</b>
<b>PART V: ADDITIONAL TECHNIQUES</b>	
<b>CHAPTER 24: WINDOWS COMMUNICATION FOUNDATION</b>	<b>761</b>
<b>What Is WCF?</b>	<b>762</b>
<b>WCF Concepts</b>	<b>763</b>
WCF Communication Protocols	763
Addresses, Endpoints, and Bindings	764

---

Contracts	766
Message Patterns	767
Behaviors	767
Hosting	768
<b>WCF Programming</b>	<b>768</b>
The WCF Test Client	774
Defining WCF Service Contracts	777
Data Contracts	777
Service Contracts	778
Operation Contracts	778
Message Contracts	779
Fault Contracts	779
Self-Hosted WCF Services	784
<b>CHAPTER 25: UNIVERSAL APPS</b>	<b>793</b>
<hr/>	
Getting Started	794
Windows Universal Apps	795
App Concepts and Design	796
Screen Orientation	796
Menus and Toolbars	796
Tiles and Badges	796
App Lifetime	797
Lock Screen Apps	797
App Development	797
Adaptive Displays	797
Relative Panel	798
Adaptive Triggers	798
FlipView	800
Sandboxed Apps	805
Disk Access	806
Serialization, Streams, and Async Programming	806
Navigation between Pages	810
The CommandBar Control	812
Managing State	814
Common Elements of Windows Store Apps	816
The Windows Store	818
Packaging an App	818
Creating the Package	819
<b>APPENDIX: EXERCISE SOLUTIONS</b>	<b>821</b>
<hr/>	
<i>INDEX</i>	865



# INTRODUCTION

**THE C# LANGUAGE WAS UNVEILED TO THE WORLD** when Microsoft announced the first version of its .NET Framework in July 2000. Since then its popularity has rocketed, and it has arguably become the language of choice for desktop, web, cloud, and cross-platform developers who use the .NET Framework. Part of the appeal of C# comes from its clear syntax, which derives from C/C++ but simplifies some things that have previously discouraged some programmers. Despite this simplification, C# has retained the power of C++, and there is now no reason not to move into C#. The language is not difficult and it's a great one to learn elementary programming techniques with. This ease of learning, combined with the capabilities of the .NET Framework, make C# an excellent way to start your programming career.

The latest release of C# is C# 7 (included with version 4.7 of the .NET Framework), which builds on the existing successes and adds even more attractive features. The latest release of Visual Studio (Visual Studio 2017) and the Visual Studio Code 2017 line of development tools also bring many tweaks and improvements to make your life easier and to dramatically increase your productivity.

This book is intended to teach you about all aspects of C# programming, including the language itself, desktop, cloud, and cross-platform programming, making use of data sources, and some new and advanced techniques. You'll also learn about the capabilities of Visual Studio 2017 and all the ways that this product can aid your application development.

The book is written in a friendly, mentor-style fashion, with each chapter building on previous ones, and every effort is made to ease you into advanced techniques painlessly. At no point will technical terms appear from nowhere to discourage you from continuing; every concept is introduced and discussed as required. Technical jargon is kept to a minimum, but where it is necessary, it, too, is properly defined and laid out in context.

The authors of this book are all experts in their field and are all enthusiastic in their passion for both the C# language and the .NET Framework. Nowhere will you find a group of people better qualified to take you under their collective wing and nurture your understanding of C# from first principles to advanced techniques. Along with the fundamental knowledge it provides, this book is packed full of helpful hints, tips, exercises, and full-fledged example code (available for download on this book's web page at [www.wrox.com](http://www.wrox.com) and at <https://github.com/benperk/BeginningCSharp7>) that you will find yourself returning to repeatedly as your career progresses.

We pass this knowledge on without begrudging it and hope that you will be able to use it to become the best programmer you can be. Good luck, and all the best!

## WHO THIS BOOK IS FOR

This book is for everyone who wants to learn how to program in C# using the .NET Framework. It is for absolute beginners who want to give programming a try by learning a clean, modern, elegant programming language. But it is also for people familiar with other programming languages who want to explore the .NET platform, as well as for existing .NET developers who want to give Microsoft's .NET flagship language a try.

## WHAT THIS BOOK COVERS

The early chapters cover the language itself, assuming no prior programming experience. If you have programmed in other languages before, much of the material in these chapters will be familiar. Many aspects of C# syntax are shared with other languages, and many structures are common to practically all programming languages (such as looping and branching structures). However, even if you are an experienced programmer, you will benefit from looking through these chapters to learn the specifics of how these techniques apply to C#.

If you are new to programming, you should start from the beginning, where you will learn basic programming concepts and become acquainted with both C# and the .NET platform that underpins it. If you are new to the .NET Framework but know how to program, you should read Chapter 1 and then skim through the next few chapters before continuing with the application of the C# language. If you know how to program but haven't encountered an object-oriented programming language before, you should read the chapters from Chapter 8 onward.

Alternatively, if you already know the C# language, you might want to concentrate on the chapters dealing with the most recent .NET Framework and C# language developments, specifically the chapters on collections, generics, and C# language enhancements (Chapters 11 and 12).

The chapters in this book have been written with a dual purpose in mind: They can be read sequentially to provide a complete tutorial in the C# language, and they can be dipped into as required reference material.

In addition to the core material, starting with Chapter 3 most chapters also include a selection of exercises at the end, which you can work through to ensure that you have understood the material. The exercises range from simple multiple choice or true/false questions to more complex exercises that require you to modify or build applications. The answers to all the exercises are provided in the Appendix. You can also find these exercises as part of the wrox.com code downloads on this book's page at [www.wrox.com](http://www.wrox.com).

This book also gives plenty of love and attention to coincide with the release of C# 7 and .NET 4.7. Every chapter received an overhaul, with less relevant material removed, and new material added. All of the code has been tested against the latest version of the development tools used, and all of



the screenshots have been retaken in Windows 10 to provide the most current windows and dialog boxes. New highlights of this edition include the following:

- Additional and improved code examples for you to try out
- Coverage of everything that's new in C# 7 and .NET 4.7
- Examples of programming .NET Core and ASP.NET Core for running cross-platform
- Examples of programming cloud applications and using Azure SDK to create and access cloud resources

## HOW THIS BOOK IS STRUCTURED

This book is divided into six sections:

- **Introduction**—Purpose and general outline of the book's contents
- **The C# Language**—Covers all aspects of the C# language, from the fundamentals to object-oriented techniques
- **Windows Programming**—How to write and deploy desktop applications with the Windows Presentation Foundation library (WPF)
- **Cloud and Cross-Platform Programming**—Cloud and cross-platform application development and deployment, including the creation and consumption of a Web API
- **Data Access**—How to use data in your applications, including data stored in files on your hard disk, data stored in XML format, and data in databases
- **Additional Techniques**—An examination of some extra ways to use C# and the .NET Framework, including Windows Communication Foundation (WCF) and Universal Windows Applications

The following sections describe the chapters in the five major parts of this book.

### The C# Language (Chapters 1–13)

Chapter 1 introduces you to C# and how it fits into the .NET landscape. You'll learn the fundamentals of programming in this environment and how Visual Studio 2017 (VS) fits in.

Chapter 2 starts you off with writing C# applications. You'll look at the syntax of C# and put the language to use with sample command-line and Windows applications. These examples demonstrate just how quick and easy it can be to get up and running, and along the way you'll be introduced to the Visual Studio development environment and the basic windows and tools that you'll be using throughout the book.

Next, you'll learn more about the basics of the C# language. You'll learn what variables are and how to manipulate them in Chapter 3. You'll enhance the structure of your applications with flow control (looping and branching) in Chapter 4, and you'll see some more advanced variable types such as arrays in Chapter 5. In Chapter 6 you'll start to encapsulate your code in the form of functions, which makes it much easier to perform repetitive operations and makes your code much more readable.

By the beginning of Chapter 7 you'll have a handle on the fundamentals of the C# language, and you will focus on debugging your applications. This involves looking at outputting trace information as your applications are executed, and at how Visual Studio can be used to trap errors and lead you to solutions for them with its powerful debugging environment.

From Chapter 8 onward you'll learn about object-oriented programming (OOP), starting with a look at what this term means and an answer to the eternal question, "What is an object?" OOP can seem quite difficult at first. The whole of Chapter 8 is devoted to demystifying it and explaining what makes it so great, and you won't actually deal with much C# code until the very end of the chapter.

Everything changes in Chapter 9, when you put theory into practice and start using OOP in your C# applications. This is where the true power of C# lies. You'll start by looking at how to define classes and interfaces, and then move on to class members (including fields, properties, and methods) in Chapter 10. At the end of that chapter you'll start to assemble a card game application, which is developed over several chapters and will help to illustrate OOP.

Once you've learned how OOP works in C#, Chapter 11 moves on to look at common OOP scenarios, including dealing with collections of objects, and comparing and converting objects. Chapter 12 takes a look at a very useful feature of C# that was introduced in .NET 2.0: generics, which enable you to create very flexible classes. Next, Chapter 13 continues the discussion of the C# language and OOP with some additional techniques, notably events, which become very important in, for example, Windows programming. Chapter 13 wraps up the fundamentals by focusing on C# language features that were introduced with versions 3.0, 4, 5, and 6 of the language.

## **Windows Programming (Chapters 14–15)**

Chapter 14 starts by introducing you to what is meant by Windows programming and looks at how this is achieved in Visual Studio. It focuses on WPF as a tool that enables you to build desktop applications in a graphical way and assemble advanced applications with the minimum of effort and time. You'll start with the basics of WPF programming and build up your knowledge in both this chapter and Chapter 15, which demonstrates how you can use the wealth of controls supplied by the .NET Framework in your applications.

## **Cloud and Cross-Platform Programming (Chapters 16–19)**

Chapter 16 starts by describing what cloud programming is and discusses the cloud-optimized stack. The cloud environment is not identical to the way programs have been traditionally coded, so

a few cloud programming patterns are discussed and defined. To complete this chapter, you require an Azure account, which is free, so that you can create an App Service Web App. Then, using the Azure SDK with C#, you create and access a storage account from an ASP.NET 4.7 web application.

In Chapter 17, you learn how to create and deploy an ASP.NET Web API to the cloud and then consume the Web API from a similar ASP.NET 4.7 web application. The chapter ends discussing two of the most valuable features in the cloud, scaling and the optimal utilization of hardware resources.

Chapter 18 introduces .NET Standard and .NET Core, which are tools used for targeting any application type—for example WPF, Windows, and ASP.NET. An emerging application is one that can run cross-platform such as on Linux or MacOS. The chapter provides instructions for installing .NET Core 2.0 and creating and implementing a .NET Standard library.

Chapter 19 describes ASP.NET and its many different types (e.g., ASP.NET Webforms, ASP.NET MVC, and ASP.NET Core). The exercises in this chapter utilize the .NET Standard library created in Chapter 18 from both an ASP.NET Webpage and an ASP.NET Core application.

## Data Access (Chapters 20–23)

Chapter 20 looks at how your applications can save and retrieve data to disk, both as simple text files and as more complex representations of data. You'll also learn how to compress data and how to monitor and act on file system changes.

In Chapter 21 you'll learn about the de facto standard for data exchange—namely XML—and a rapidly emerging format called JSON. By this point in the book, you'll have touched on XML several times in preceding chapters, but this chapter lays out the ground rules and shows you what all the excitement is about.

The remainder of this part looks at LINQ, which is a query language built in to the latest versions of the .NET Framework. You start in Chapter 22 with a general introduction to LINQ, and then you will use LINQ to access a database and other data in Chapter 23.

## Additional Techniques (Chapters 24–25)

Chapter 24 is an introduction to Windows Communication Foundation (WCF), which provides you with the tools you need for enterprise-level programmatic access to information and capabilities across local networks and the Internet. You will see how you can use WCF to expose complex data and functionality to web and desktop applications in a platform-independent way.

Chapter 25 shows you how you can create Universal Windows Apps, which are new to Windows. This chapter builds on the foundation of Chapters 14 and 15 to show you how to create Windows Apps that can run on all windows platforms.

## WHAT YOU NEED TO USE THIS BOOK

The code and descriptions of C# and the .NET Framework in this book apply to C# 7 and .NET 4.7. You don't need anything other than the Framework to understand this aspect of the book, but many of the examples require a development tool. This book uses Visual Studio Community 2017 as its primary development tool. Use Visual Studio Community 2017 to create Windows, cloud, and cross-platform applications as well as SQL Server Express for applications that access databases. Some functionality is available only in Visual Studio 2017, but this won't stop you from working through any of the examples in this book.

The source code for the samples is available for download from this book's page on [www.wrox.com](http://www.wrox.com) and at <https://github.com/benperk/BeginningCSharp7>.

## CONVENTIONS

To help you get the most from the text and keep track of what's happening, we've used a number of conventions throughout the book.

### TRY IT OUT

The *Try It Out* is an exercise you should work through, following the text in the book.

1. These exercises usually consist of a set of steps.
2. Each step has a number.
3. Follow the steps through with your copy of the database.

### *How It Works*

After each *Try It Out*, the code you've typed will be explained in detail.

**WARNING** Warnings hold important, not-to-be-forgotten information that is directly relevant to the surrounding text.

**NOTE** Shaded boxes like this hold notes, tips, hints, tricks, or asides to the current discussion.

As for styles in the text:

- We *italicize* new terms and important words when we introduce them.
- We show keyboard strokes like this: Ctrl+A.
- We show file names, URLs, and code within the text like so:

```
persistence.properties
```

We present code in two different ways:

We use a monofont type with no highlighting for most code examples.

We use bold to emphasize code that is particularly important in the present context or to show changes from a previous code snippet.

## SOURCE CODE

As you work through the examples in this book, you may choose either to type in all the code manually, or to use the source code files that accompany the book. All the source code used in this book is available for download on this book's page at [www.wrox.com](http://www.wrox.com) and at <https://github.com/benperk/BeginningCSharp7>.

You can also search for the book at [www.wrox.com](http://www.wrox.com) by ISBN (the ISBN for this book is 978-1-119-45868-5) to find the code. A complete list of code downloads for all current Wrox books is available at [www.wrox.com/dynamic/books/download.aspx](http://www.wrox.com/dynamic/books/download.aspx).

Most of the code on [www.wrox.com](http://www.wrox.com) is compressed in a .ZIP, .RAR archive, or similar archive format appropriate to the platform. Once you download the code, just decompress it with an appropriate compression tool.

**NOTE** *Because many books have similar titles, you may find it easiest to search by ISBN; this book's ISBN is 978-1-119-45868-5.*

*Alternatively, as just mentioned, you can also go to the main Wrox code download page at [www.wrox.com/dynamic/books/download.aspx](http://www.wrox.com/dynamic/books/download.aspx) to see the code available for this book and all other Wrox books.*

## ERRATA

We make every effort to ensure that there are no errors in the text or in the code. However, no one is perfect, and mistakes do occur. If you find an error in one of our books, like a spelling mistake or faulty piece of code, we would be very grateful for your feedback. By sending in errata, you may save another reader hours of frustration, and at the same time, you will be helping us provide even higher quality information.

To find the errata page for this book, go to this book's page at [www.wrox.com](http://www.wrox.com) and click the Errata link. On this page you can view all errata that has been submitted for this book and posted by Wrox editors.

If you don't spot "your" error on the Book Errata page, go to [www.wrox.com/contact/techsupport.shtml](http://www.wrox.com/contact/techsupport.shtml) and complete the form there to send us the error you have found. We'll check the information and, if appropriate, post a message to the book's errata page and fix the problem in subsequent editions of the book.