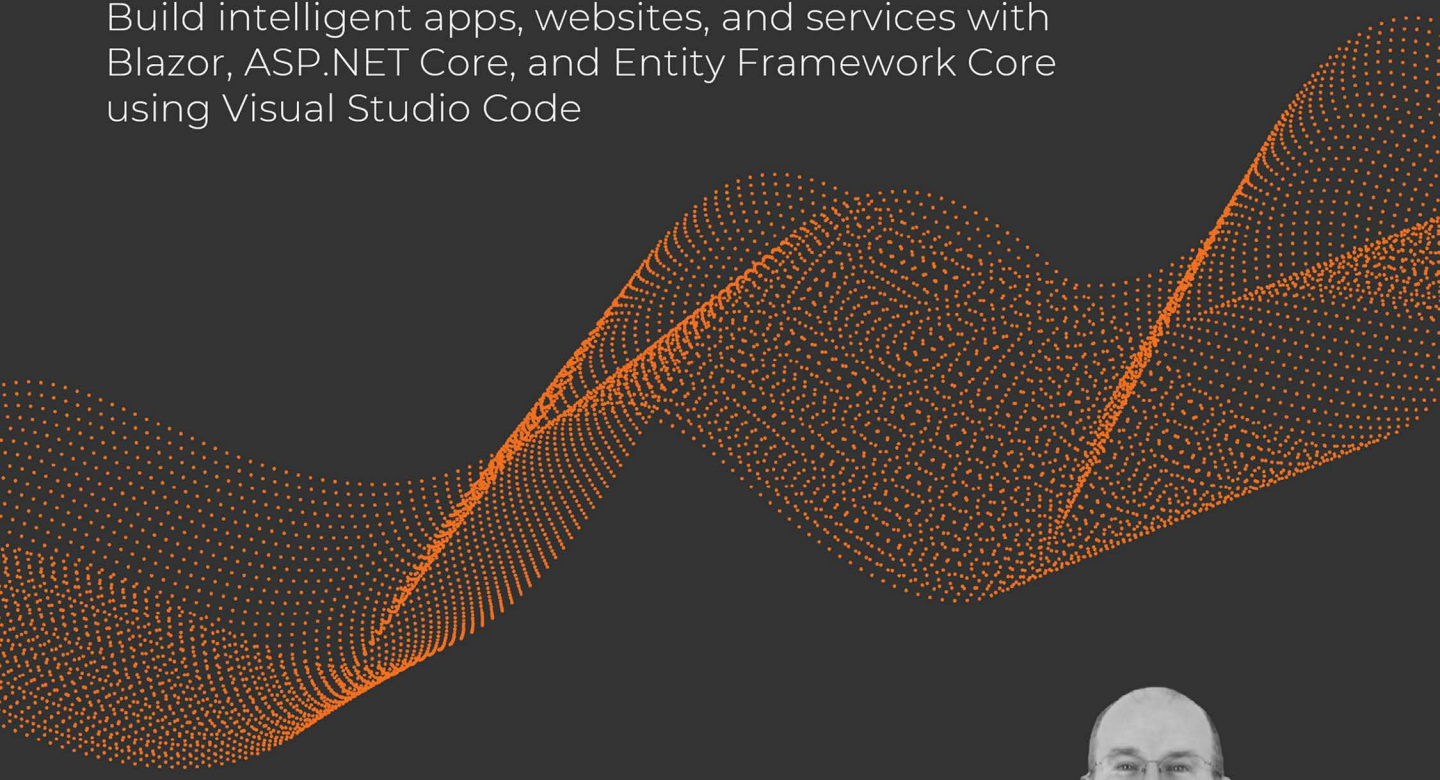
EXPERT INSIGHT

C# 9 and .NET 5

Modern Cross-Platform Development

Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code



Fifth Edition

Mark J. Price

Packt

C# 9 and .NET 5 – Modern Cross-Platform Development

Fifth Edition

Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code

Mark J. Price

BIRMINGHAM - MUMBAI

C# 9 and .NET 5 – Modern Cross-Platform Development

Fifth Edition

Producer: Ben Renow-Clarke
Acquisition Editor – Peer Reviews: Divya Mudaliar
Content Development Editors: Joanne Lovell, Bhavesh Amin
Technical Editor: Aniket Shetty
Project Editor: Radhika Atitkar
Copy Editor: Safis Editing
Proofreader: Safis Editing
Indexer: Rekha Nair
Presentation Designer: Sandip Tadge

First published: March 2016
Second edition: March 2017
Third edition: November 2017
Fourth edition: October 2019
Fifth edition: November 2020

Production reference: 1051120

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-80056-810-5

www.packt.com

packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Learn better with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.Packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.Packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Contributors

About the author

Mark J. Price is a Microsoft Specialist: Programming in C# and architecting Microsoft Azure Solutions, with more than 20 years of educational and programming experience.

Microsoft CERTIFIED Solutions Developer App Builder

Microsoft Specialist Programming in C#

Episerver CMS Certified Developer

Since 1993, Mark has passed more than 80 Microsoft programming exams and specializes in preparing others to pass them too. His students range from professionals with decades of experience to 16-year-old apprentices with none. He successfully guides all of them by combining educational skills with real-world experience in consulting and developing systems for enterprises worldwide.

Between 2001 and 2003, Mark was employed full time to write official courseware for Microsoft in Redmond, USA. His team wrote the first training courses for C# while it was still an early alpha version. While with Microsoft, he taught "train-the-trainer" classes to get other MCTs up to speed on C# and .NET.

Currently, Mark creates and delivers training courses for Episerver's Digital Experience Platform, including Content Cloud, Commerce Cloud, and Intelligence Cloud.

In 2010, Mark studied for a Postgraduate Certificate in Education (PGCE). He taught GCSE and A-Level mathematics in two London secondary schools. He holds a Computer Science BSc. Hons. degree from the University of Bristol, UK.

Thank you to my parents, Pamela and Ian, for raising me to be polite, hardworking, and curious about the world. Thank you to my sisters, Emily and Juliet, for loving me despite being their awkward older brother. Thank you to my friends and colleagues who inspire me technically and creatively. Lastly, thanks to all the students I have taught over the years for motivating me to be the best teacher that I can be.

About the reviewer

Damir Arh has many years of experience with software development and maintenance; from complex enterprise software projects to modern consumer-oriented mobile applications. Although he has worked with a wide spectrum of different languages, his favorite language remains C#. In his drive toward better development processes he is a proponent of test-driven development, continuous integration, and continuous deployment. He shares his knowledge by speaking at local user groups and conferences, blogging, and writing articles. He has received the prestigious Microsoft MVP award for developer technologies 9 times in a row. In his spare time, he's always on the move: hiking, geocaching, running, and rock climbing.

I'd like to thank my family and friends for their patience and understanding during the weekends and evenings I spent on my computer to help make this book better for everyone.

Table of Contents

Preface **xxi**

Chapter 1: Hello, C#! Welcome, .NET! **1**

Setting up your development environment **2**
Using Visual Studio Code for cross-platform development 2
Using GitHub Codespaces for development in the cloud 3
Using Visual Studio 2019 for Windows app development 4
Using Visual Studio for Mac for mobile development 4
Recommended tools for chapters 4
Deploying cross-platform 5
Understanding Microsoft Visual Studio Code versions 5
Downloading and installing Visual Studio Code 7
Installing other extensions 8

Understanding .NET **8**
Understanding .NET Framework 8
Understanding the Mono and Xamarin projects 9
Understanding .NET Core 9
Understanding .NET 5 and the journey to one .NET 10
Understanding .NET support 11
Understanding .NET Runtime and .NET SDK versions 12
Removing old versions of .NET 13
What is different about .NET Core and .NET 5? 14
Understanding .NET Standard 15
.NET platforms and tools used by the book editions 16
Understanding intermediate language 17
Comparing .NET technologies 17

Building console apps using Visual Studio Code **17**
Writing code using Visual Studio Code 18
Compiling and running code using the dotnet CLI 20
Writing top-level programs 20

Downloading solution code from the GitHub repository **21**

Using Git with Visual Studio Code 21
Cloning the book solution code repository 22
Looking for help 22
Reading Microsoft documentation 22
Getting help for the dotnet tool 22
Getting definitions of types and their members 23
Looking for answers on Stack Overflow 25
Searching for answers using Google 26
Subscribing to the official .NET blog 26
Scott Hanselman's videos 26
Practicing and exploring 27
Exercise 1.1 – Test your knowledge 27
Exercise 1.2 – Practice C# anywhere 27
Exercise 1.3 – Explore topics 27
Summary 28
Chapter 2: Speaking C# 29
Introducing C# 29
Understanding language versions and features 30
C# 1.0 30
C# 2.0 30
C# 3.0 30
C# 4.0 31
C# 5.0 31
C# 6.0 31
C# 7.0 31
C# 7.1 32
C# 7.2 32
C# 7.3 32
C# 8.0 32
C# 9.0 33
Discovering your C# compiler versions 33
Enabling a specific language version compiler 35
Understanding C# basics 36
Understanding C# grammar 37
Statements 37
Comments 37
Blocks 38
Understanding C# vocabulary 38
Changing the color scheme for syntax 38
Comparing programming languages to human languages 39
Help for writing correct code 39
Verbs are methods 40
Nouns are types, fields, and variables 40
Revealing the extent of the C# vocabulary 41
Working with variables 43
Naming things and assigning values 44
Literal values 44
Storing text 44

Understanding verbatim strings 45
Storing numbers 46
Storing whole numbers 46
Storing real numbers 48
Writing code to explore number sizes 48
Comparing double and decimal types 49
Storing Booleans 51
Using Visual Studio Code workspaces 51
Storing any type of object 52
Storing dynamic types 53
Declaring local variables 54
Specifying and inferring the type of a local variable 54
Using target-typed new to instantiate objects 55
Getting default values for types 55
Storing multiple values 56
Working with null values 57
Making a value type nullable 57
Understanding nullable reference types 58
Enabling nullable and non-nullable reference types 59
Declaring non-nullable variables and parameters 59
Checking for null 61
Exploring console applications further 62
Displaying output to the user 62
Formatting using numbered positional arguments 62
Formatting using interpolated strings 63
Understanding format strings 63
Getting text input from the user 65
Importing a namespace 65
Simplifying the usage of the console 66
Getting key input from the user 66
Getting arguments 67
Setting options with arguments 69
Handling platforms that do not support an API 70
Practicing and exploring 71
Exercise 2.1 – Test your knowledge 71
Exercise 2.2 – Practice number sizes and ranges 71
Exercise 2.3 – Explore topics 72
Summary 72
Chapter 3: Controlling Flow and Converting Types 73
Operating on variables 73
Unary operators 74
Binary arithmetic operators 75
Assignment operators 76
Logical operators 76
Conditional logical operators 78
Bitwise and binary shift operators 79

Miscellaneous operators 80
Understanding selection statements 81
Branching with the if statement 81
Why you should always use braces with if statements 82
Pattern matching with the if statement 83
Branching with the switch statement 83
Pattern matching with the switch statement 85
Simplifying switch statements with switch expressions 87
Understanding iteration statements 88
Looping with the while statement 88
Looping with the do statement 88
Looping with the for statement 89
Looping with the foreach statement 90
Understanding how foreach works internally 90
Casting and converting between types 91
Casting numbers implicitly and explicitly 91
Converting with the System.Convert type 93
Rounding numbers 94
Understanding the default rounding rules 94
Taking control of rounding rules 95
Converting from any type to a string 95
Converting from a binary object to a string 96
Parsing from strings to numbers or dates and times 97
Avoiding exceptions using the TryParse method 98
Handling exceptions when converting types 99
Wrapping error-prone code in a try block 99
Catching all exceptions 100
Catching specific exceptions 101
Checking for overflow 102
Throwing overflow exceptions with the checked statement 103
Disabling compiler overflow checks with the unchecked statement 104
Practicing and exploring 105
Exercise 3.1 – Test your knowledge 106
Exercise 3.2 – Explore loops and overflow 106
Exercise 3.3 – Practice loops and operators 106
Exercise 3.4 – Practice exception handling 107
Exercise 3.5 – Test your knowledge of operators 108
Exercise 3.6 – Explore topics 108
Summary 108
Chapter 4: Writing, Debugging, and Testing Functions 109
Writing functions 109
Writing a times table function 110
Writing a function that returns a value 112
Writing mathematical functions 114
Converting numbers from cardinal to ordinal 114
Calculating factorials with recursion 116

Documenting functions with XML comments 119
Using lambdas in function implementations 120
Debugging during development 123
Creating code with a deliberate bug 123
Setting a breakpoint 124
Navigating with the debugging toolbar 125
Debugging windows 126
Stepping through code 126
Customizing breakpoints 127
Logging during development and runtime 128
Instrumenting with Debug and Trace 129
Writing to the default trace listener 130
Configuring trace listeners 131
Switching trace levels 132
Unit testing functions 135
Creating a class library that needs testing 135
Writing unit tests 137
Running unit tests 138
Practicing and exploring 139
Exercise 4.1 – Test your knowledge 139
Exercise 4.2 – Practice writing functions with debugging and unit testing 140
Exercise 4.3 – Explore topics 140
Summary 140
Chapter 5: Building Your Own Types with Object-Oriented Programming 141
Talking about object-oriented programming 141
Building class libraries 142
Creating a class library 142
Defining a class 143
Understanding members 144
Instantiating a class 145
Referencing an assembly 145
Importing a namespace to use a type 146
Managing multiple files 146
Understanding objects 147
Inheriting from System.Object 147
Storing data within fields 148
Defining fields 148
Understanding access modifiers 149
Setting and outputting field values 149
Storing a value using an enum type 150
Storing multiple values using an enum type 152
Storing multiple values using collections 153
Making a field static 154
Making a field constant 155
Making a field read-only 156

Initializing fields with constructors 157
Setting fields with default literals 158
Writing and calling methods 160
Returning values from methods 160
Combining multiple returned values using tuples 161
Naming the fields of a tuple 162
Inferring tuple names 163
Deconstructing tuples 163
Defining and passing parameters to methods 164
Overloading methods 164
Passing optional parameters and naming arguments 165
Controlling how parameters are passed 167
Understanding ref returns 168
Splitting classes using partial 168
Controlling access with properties and indexers 169
Defining read-only properties 169
Defining settable properties 171
Defining indexers 172
Pattern matching with objects 173
Creating and referencing a .NET 5 class library 173
Defining flight passengers 174
Enhancements to pattern matching in C# 9 176
Working with records 177
Init-only properties 177
Understanding records 178
Simplifying data members 179
Positional records 179
Practicing and exploring 180
Exercise 5.1 – Test your knowledge 180
Exercise 5.2 – Explore topics 181
Summary 181
Chapter 6: Implementing Interfaces and Inheriting Classes 183
Setting up a class library and console application 184
Simplifying methods 186
Implementing functionality using methods 186
Implementing functionality using operators 188
Implementing functionality using local functions 189
Raising and handling events 190
Calling methods using delegates 190
Defining and handling delegates 191
Defining and handling events 193
Implementing interfaces 194
Common interfaces 194
Comparing objects when sorting 195
Comparing objects using a separate class 197

Defining interfaces with default implementations 198
Making types safely reusable with generics 200
Working with generic types 202
Working with generic methods 203
Managing memory with reference and value types 204
Working with struct types 205
Releasing unmanaged resources 207
Ensuring that Dispose is called 209
Inheriting from classes 210
Extending classes to add functionality 210
Hiding members 211
Overriding members 212
Preventing inheritance and overriding 213
Understanding polymorphism 214
Casting within inheritance hierarchies 215
Implicit casting 215
Explicit casting 215
Avoiding casting exceptions 216
Inheriting and extending .NET types 217
Inheriting exceptions 217
Extending types when you can't inherit 219
Using static methods to reuse functionality 219
Using extension methods to reuse functionality 220
Practicing and exploring 221
Exercise 6.1 – Test your knowledge 221
Exercise 6.2 – Practice creating an inheritance hierarchy 222
Exercise 6.3 – Explore topics 222
Summary 223
Chapter 7: Understanding and Packaging .NET Types 225
Introducing .NET 5 225
.NET Core 1.0 226
.NET Core 1.1 227
.NET Core 2.0 227
.NET Core 2.1 227
.NET Core 2.2 228
.NET Core 3.0 228
.NET 5.0 228
Improving performance from .NET Core 2.0 to .NET 5 229
Understanding .NET components 229
Understanding assemblies, packages, and namespaces 230
Understanding dependent assemblies 230
Understanding the Microsoft .NET project SDKs 231
Understanding NuGet packages 232
Understanding frameworks 232
Importing a namespace to use a type 233

Relating C# keywords to .NET types 234
Sharing code with legacy platforms using .NET Standard 235
Creating a .NET Standard 2.0 class library 236
Publishing your applications for deployment 237
Creating a console application to publish 237
Understanding dotnet commands 238
Creating new projects 238
Managing projects 239
Publishing a self-contained app 239
Publishing a single-file app 240
Reducing the size of apps using app trimming 242
Decompiling assemblies 243
Packaging your libraries for NuGet distribution 246
Referencing a NuGet package 246
Fixing dependencies 247
Packaging a library for NuGet 247
Testing your package 250
Porting from .NET Framework to .NET 5 252
Could you port? 252
Should you port? 252
Differences between .NET Framework and .NET 5 253
Understanding the .NET Portability Analyzer 253
Using non-.NET Standard libraries 253
Practicing and exploring 255
Exercise 7.1 – Test your knowledge 255
Exercise 7.2 – Explore topics 256
Summary 256
Chapter 8: Working with Common .NET Types 257
Working with numbers 258
Working with big integers 258
Working with complex numbers 259
Working with text 260
Getting the length of a string 260
Getting the characters of a string 260
Splitting a string 261
Getting part of a string 261
Checking a string for content 262
Joining, formatting, and other string members 262
Building strings efficiently 264
Pattern matching with regular expressions 264
Checking for digits entered as text 264
Understanding the syntax of a regular expression 266
Examples of regular expressions 266
Splitting a complex comma-separated string 267
Regular expression performance improvements 268

Storing multiple objects in collections **269**
Common features of all collections 270
Understanding collection choices 271
Lists 272
Dictionaries 272
Stacks 273
Queues 273
Sets 273
Working with lists 273
Working with dictionaries 275
Sorting collections 275
Using specialized collections 276
Using immutable collections 276
Working with spans, indexes, and ranges **277**
Using memory efficiently using spans 277
Identifying positions with the Index type 278
Identifying ranges with the Range type 278
Using indexes and ranges 279
Working with network resources **280**
Working with URIs, DNS, and IP addresses 280
Pinging a server 281
Working with types and attributes **282**
Versioning of assemblies 283
Reading assembly metadata 284
Creating custom attributes 286
Doing more with reflection 288
Working with images **289**
Internationalizing your code **290**
Detecting and changing the current culture 291
Handling time zones 293
Practicing and exploring **293**
Exercise 8.1 – Test your knowledge 293
Exercise 8.2 – Practice regular expressions 294
Exercise 8.3 – Practice writing extension methods 294
Exercise 8.4 – Explore topics 294
Summary **295**
Chapter 9: Working with Files, Streams, and Serialization **297**
Managing the filesystem **297**
Handling cross-platform environments and filesystems 297
Managing drives 299
Managing directories 300
Managing files 303
Managing paths 304
Getting file information 305
Controlling how you work with files 307

Reading and writing with streams **307**
Writing to text streams 309
Writing to XML streams 310
Disposing of file resources 312
Compressing streams 314
Compressing with the Brotli algorithm 316
High-performance streams using pipelines 318
Asynchronous streams 318
Encoding and decoding text **319**
Encoding strings as byte arrays 319
Encoding and decoding text in files 322
Serializing object graphs **322**
Serializing as XML 322
Generating compact XML 325
Deserializing XML files 326
Serializing with JSON 327
High-performance JSON processing 328
Practicing and exploring **330**
Exercise 9.1 – Test your knowledge 330
Exercise 9.2 – Practice serializing as XML 331
Exercise 9.3 – Explore topics 332
Summary **332**
Chapter 10: Protecting Your Data and Applications **333**
Understanding the vocabulary of protection **334**
Keys and key sizes 334
IVs and block sizes 335
Salts 335
Generating keys and IVs 336
Encrypting and decrypting data **336**
Encrypting symmetrically with AES 337
Hashing data **341**
Hashing with the commonly used SHA256 342
Signing data **345**
Signing with SHA256 and RSA 346
Generating random numbers **349**
Generating random numbers for games 349
Generating random numbers for cryptography 350
What's new in cryptography? **351**
Authenticating and authorizing users **352**
Implementing authentication and authorization 354
Protecting application functionality 357
Practicing and exploring **358**
Exercise 10.1 – Test your knowledge 358
Exercise 10.2 – Practice protecting data with encryption and hashing 358

Exercise 10.3 – Practice protecting data with decryption 359
Exercise 10.4 – Explore topics 359
Summary 359
Chapter 11: Working with Databases Using Entity Framework Core 361
Understanding modern databases 361
Understanding legacy Entity Framework 362
Understanding Entity Framework Core 363
Using a sample relational database 363
Setting up SQLite for macOS 364
Setting up SQLite for Windows 365
Creating the Northwind sample database for SQLite 365
Managing the Northwind sample database with SQLiteStudio 366
Setting up EF Core 367
Choosing an EF Core database provider 367
Setting up the dotnet-ef tool 368
Connecting to the database 369
Defining EF Core models 369
EF Core conventions 370
EF Core annotation attributes 370
EF Core Fluent API 371
Understanding data seeding 372
Building an EF Core model 372
Defining the Category and Product entity classes 373
Defining the Northwind database context class 375
Scaffolding models using an existing database 377
Querying EF Core models 381
Filtering included entities 383
Filtering and sorting products 384
Getting the generated SQL 386
Logging EF Core 387
Logging with query tags 391
Pattern matching with Like 391
Defining global filters 392
Loading patterns with EF Core 393
Eager loading entities 393
Enabling lazy loading 394
Explicit loading entities 395
Manipulating data with EF Core 397
Inserting entities 397
Updating entities 399
Deleting entities 400
Pooling database contexts 401
Transactions 401
Defining an explicit transaction 402
Practicing and exploring 403

Exercise 11.1 – Test your knowledge 403
Exercise 11.2 – Practice exporting data using different serialization formats 403
Exercise 11.3 – Explore the EF Core documentation 404
Summary 404
Chapter 12: Querying and Manipulating Data Using LINQ 405
Writing LINQ queries 405
Extending sequences with the Enumerable class 406
Filtering entities with Where 407
Targeting a named method 409
Simplifying the code by removing the explicit delegate instantiation 410
Targeting a lambda expression 410
Sorting entities 410
Sorting by a single property using OrderBy 411
Sorting by a subsequent property using ThenBy 411
Filtering by type 412
Working with sets and bags using LINQ 413
Using LINQ with EF Core 415
Building an EF Core model 415
Filtering and sorting sequences 418
Projecting sequences into new types 419
Joining and grouping sequences 420
Aggregating sequences 424
Sweetening LINQ syntax with syntactic sugar 425
Using multiple threads with parallel LINQ 426
Creating an app that benefits from multiple threads 426
Using Windows 10 427
Using macOS 427
For all operating systems 428
Creating your own LINQ extension methods 429
Working with LINQ to XML 433
Generating XML using LINQ to XML 433
Reading XML using LINQ to XML 433
Practicing and exploring 434
Exercise 12.1 – Test your knowledge 435
Exercise 12.2 – Practice querying with LINQ 435
Exercise 12.3 – Explore topics 436
Summary 436
Chapter 13: Improving Performance and Scalability Using Multitasking 437
Understanding processes, threads, and tasks 437
Monitoring performance and resource usage 439
Evaluating the efficiency of types 439
Monitoring performance and memory use 440
Implementing the Recorder class 441
Measuring the efficiency of processing strings 443
Running tasks asynchronously 445

Running multiple actions synchronously 445
Running multiple actions asynchronously using tasks 446
Waiting for tasks 448
Continuing with another task 449
Nested and child tasks 450
Synchronizing access to shared resources 452
Accessing a resource from multiple threads 452
Applying a mutually exclusive lock to a resource 454
Understanding the lock statement and avoiding deadlocks 454
Synchronizing events 456
Making CPU operations atomic 457
Applying other types of synchronization 458
Understanding async and await 458
Improving responsiveness for console apps 459
Improving responsiveness for GUI apps 460
Improving scalability for web applications and web services 460
Common types that support multitasking 461
Using await in catch blocks 461
Working with async streams 461
Practicing and exploring 462
Exercise 13.1 – Test your knowledge 462
Exercise 13.2 – Explore topics 463
Summary 463
Chapter 14: Introducing Practical Applications of C# and .NET 465
Understanding app models for C# and .NET 465
Building websites using ASP.NET Core 466
Building websites using a web content management system 466
Understanding web applications 467
Building and consuming web services 468
Building intelligent apps 468
New features in ASP.NET Core 468
ASP.NET Core 1.0 468
ASP.NET Core 1.1 469
ASP.NET Core 2.0 469
ASP.NET Core 2.1 469
ASP.NET Core 2.2 470
ASP.NET Core 3.0 470
ASP.NET Core 3.1 471
Blazor WebAssembly 3.2 471
ASP.NET Core 5.0 471
Understanding SignalR 472
Understanding Blazor 474
JavaScript and friends 474
Silverlight – C# and .NET using a plugin 474