C# 7.1 and .NET Core 2.0 – Modern Cross-Platform Development

Table of Contents

Preface
- What this book covers
  - Part 1 – C# 7.1
  - Part 2 – .NET Core 2.0 and .NET Standard 2.0
  - Part 3 – App Models
- What you need for this book
- Who this book is for
- Conventions
- Reader feedback
- Customer support
  - Downloading the example code
  - Downloading the color images of this book
  - Errata
  - Piracy
  - Questions

1. Hello, C#! Welcome, .NET Core!
   - Setting up your development environment
     - Using alternative C# IDEs
     - Deploying cross-platform
     - Installing Microsoft Visual Studio 2017
       - Choosing workloads
       - Choosing additional components
     - Installing Microsoft Visual Studio Code
       - Installing Microsoft Visual Studio Code for macOS
       - Installing .NET Core SDK for macOS
       - Installing Node Package Manager for macOS
       - Installing the Visual Studio Code extension for C#
     - Installing Visual Studio for Mac
       - Installing Xcode
       - Downloading and installing Visual Studio for Mac
   - Understanding .NET
     - Understanding .NET Framework
     - Understanding the Mono and Xamarin projects
     - Understanding .NET Core
     - Understanding .NET Standard
     - Understanding .NET Native

        - Comparing .NET technologies
    - Writing and compiling code using the .NET Core CLI tool
        - Writing code using a simple text editor
            - If you are using Windows Notepad
            - If you are using macOS TextEdit
        - Creating and compiling apps using the .NET Core CLI tool
            - Creating a console application at Command Prompt
            - Restoring packages, compiling code, and running the application
        - Fixing compiler errors
        - Understanding intermediate language
    - Writing and compiling code using Visual Studio 2017
        - Writing code using Microsoft Visual Studio 2017
        - Compiling code using Visual Studio 2017
        - Fixing mistakes with the error list
        - Adding existing projects to Visual Studio 2017
        - Autoformatting code
        - Experimenting with C# Interactive
        - Other useful windows
    - Writing and compiling code using Visual Studio Code
        - Writing code using Visual Studio Code
        - Compiling code using Visual Studio Code
        - Autoformatting code
    - Writing and compiling code using Visual Studio for Mac
        - Next steps
    - Managing source code with GitHub
        - Using Git with Visual Studio 2017
            - Using the Team Explorer window
            - Cloning a GitHub repository
            - Managing a GitHub repository
        - Using Git with Visual Studio Code
            - Configuring Git at the command line
            - Managing Git with Visual Studio Code
    - Practicing and exploring
        - Exercise 1.1 – Test your knowledge
        - Exercise 1.2 – Practice C# anywhere
        - Exercise 1.3 – Explore topics
    - Summary
2. Part 1, C# 7.1
3. Speaking C#
    - Understanding C# basics

- Using Visual Studio 2017
- Using Visual Studio Code on macOS, Linux, or Windows
- C# grammar
  - Statements
  - Comments
  - Blocks
- C# vocabulary
- Help for writing correct code
- Verbs are methods
- Nouns are types, fields, and variables
- Revealing the extent of the C# vocabulary
  - Building and running with Visual Studio 2017
  - Building and running with Visual Studio Code
  - Adding more types with Visual Studio 2017 and Visual Studio Code

Declaring variables

- Naming variables
- Literal values
- Storing text
- Storing numbers
  - Storing whole numbers
  - C# 7 improvements
  - Storing real numbers
    - Using Visual Studio 2017
    - Using Visual Studio Code
  - Writing code to explore numbers
    - Comparing double and decimal types
- Storing Booleans
- The object type
- The dynamic type
- Local variables
  - Specifying the type of a local variable
  - Inferring the type of a local variable
- Making a value type nullable
- Understanding nullable reference types
  - The billion-dollar mistake
  - Changing the defaults for nullable types in C# 8.0
  - Checking for null
- Storing multiple values in an array

Exploring console applications further

Displaying output to the user

Getting input from the user

Importing a namespace

Simplifying the usage of the console
Reading arguments and working with arrays

Passing arguments with Visual Studio 2017

Passing arguments with Visual Studio Code

Viewing the output
Enumerating arguments

Running on Windows

Running on macOS

Handling platforms that do not support an API
Operating on variables

Experimenting with unary operators

Experimenting with arithmetic operators

Comparison and Boolean operators
Practicing and exploring

Exercise 2.1 – Test your knowledge

Exercise 2.2 – Practice number sizes and ranges

Exercise 2.3 – Explore topics

Summary

4. Controlling the Flow and Converting Types
Selection statements

Using Visual Studio 2017

Using Visual Studio Code on macOS, Linux, or Windows
The if statement

The code

Pattern matching with the if statement
The switch statement

The code

Pattern matching with the switch statement
Iteration statements

The while statement

The do statement

The for statement

The foreach statement
Casting and converting between types
Casting from numbers to numbers

Casting numbers implicitly

Casting numbers explicitly

- Using the convert type
- Rounding numbers
- Converting from any type to a string
- Converting from a binary object to a string
- Parsing from strings to numbers or dates and times

Handling exceptions when converting types
- The try statement
- Catching all exceptions
- Catching specific exceptions

Checking for overflow
- The checked statement
- The unchecked statement

Looking for help
- Microsoft Docs and MSDN
- Go to definition
- Stack Overflow
- Google
- Subscribing to blogs
- Design patterns
  - Singleton pattern

Practicing and exploring
- Exercise 3.1 – Test your knowledge
- Exercise 3.2 – Explore loops and overflow
- Exercise 3.3 – Practice loops and operators
- Exercise 3.4 – Practice exception handling
- Exercise 3.5 – Explore topics

Summary

5. Writing, Debugging, and Testing Functions

Writing functions
- Writing a times table function
- Writing a function that returns a value
- Writing mathematical functions
  - Formatting numbers for output
  - Calculating factorials with recursion

Debugging an application during development
- Creating an application with a deliberate bug
- Setting a breakpoint
- The debugging toolbar
- Debugging windows

- Stepping through code
- Customizing breakpoints
- Logging during development and runtime
  - Instrumenting with Debug and Trace
    - Writing to the default trace listener
    - Configuring trace listeners
  - Switching trace levels
- Unit testing functions
  - Creating a class library that needs testing with Visual Studio 2017
  - Creating a unit test project with Visual Studio 2017
  - Creating a class library that needs testing with Visual Studio Code
  - Writing unit tests
  - Running unit tests with Visual Studio 2017
  - Running unit tests with Visual Studio Code
- Practicing and exploring
  - Exercise 4.1 – Test your knowledge
  - Exercise 4.2 – Practice writing functions with debugging and unit test ing
  - Exercise 4.3 – Explore topics
- Summary

6. Building Your Own Types with Object-Oriented Programming
- Talking about OOP
- Building class libraries
  - Creating a class library with Visual Studio 2017
  - Creating a class library with Visual Studio Code
  - Defining a class
  - Instantiating a class
    - Referencing an assembly using Visual Studio 2017
    - Referencing an assembly using Visual Studio Code
    - Importing a namespace
  - Managing multiple projects with Visual Studio Code
  - Inheriting from System.Object
- Storing data with fields
  - Defining fields
    - Understanding access modifiers
  - Storing a value using the enum keyword
  - Storing multiple values using collections
  - Making a field static
  - Making a field constant

- Making a field read-only
- Initializing fields with constructors
- Setting fields with default literal

Writing and calling methods
- Combining multiple values with tuples
  - Defining methods with tuples
  - Naming the fields of a tuple
  - Inferring tuple names
  - Deconstructing tuples
- Defining and passing parameters to methods
- Overloading methods
- Optional parameters and named arguments

Controlling how parameters are passed

Splitting classes using partial

Controlling access with properties and indexers
- Defining read-only properties
- Defining settable properties
- Defining indexers

Practicing and exploring
- Exercise 5.1 – Test your knowledge
- Exercise 5.2 – Explore topics

Summary

7. Implementing Interfaces and Inheriting Classes

Setting up a class library and console application
- Using Visual Studio 2017
- Using Visual Studio Code
- Defining the classes

Simplifying methods with operators
- Implementing some functionality with a method
- Implementing some functionality with an operator

Defining local functions

Raising and handling events
- Calling methods using delegates
- Defining events
  - Using Visual Studio 2017
  - Using Visual Studio Code
  - Using Visual Studio 2017 or Visual Studio Code

Implementing interfaces
- Common interfaces
- Comparing objects when sorting

Attempting to sort objects without a method to compare

Defining a method to compare

Defining a separate comparer

Making types more reusable with generics

Making a generic type

Making a generic method

Managing memory with reference and value types

Defining a struct type

Releasing unmanaged resources

Ensuring that dispose is called

Inheriting from classes

Extending classes

Hiding members

Overriding members

Using Visual Studio 2017

Using Visual Studio 2017 or Visual Studio Code

Preventing inheritance and overriding

Polymorphism

Casting within inheritance hierarchies

Implicit casting

Explicit casting

Handling casting exceptions

Inheriting and extending .NET types

Inheriting from an exception

Extending types when you can't inherit

Using static methods to reuse functionality

Using extension methods to reuse functionality

Practicing and exploring

Exercise 6.1 – Test your knowledge

Exercise 6.2 – Practice creating an inheritance hierarchy

Exercise 6.3 – Explore topics

Summary

8. Part 2 – .NET Core 2.0 and .NET Standard 2.0

9. Understanding and Packaging .NET Standard Types

Understanding assemblies and namespaces

Base Class Libraries and CoreFX

Assemblies, NuGet packages, and platforms

Namespaces

Understanding dependent assemblies

Using Visual Studio 2017

            Using Visual Studio Code

            Using Visual Studio 2017 and Visual Studio Code
        Relating assemblies and namespaces

            Browsing assemblies with Visual Studio 2017

            Using Visual Studio 2017 or Visual Studio Code

            Importing a namespace

        Relating C# keywords to .NET types
    Sharing code cross-platform with .NET Standard 2.0 class libraries
        Creating a .NET Standard 2.0 class library

            Using Visual Studio 2017

            Using Visual Studio Code
    Understanding NuGet packages

        Understanding metapackages

        Understanding frameworks

        Fixing dependencies
    Publishing your applications for deployment

        Creating a console application to publish

        Publishing with Visual Studio 2017 on Windows

        Publishing with Visual Studio Code on macOS
    Packaging your libraries for NuGet distribution

        Understanding dotnet commands
        Adding a package reference

            Using Visual Studio Code

            Using Visual Studio 2017

        Packaging a library for NuGet
        Testing your package

            Using Visual Studio Code

            Using Visual Studio 2017

            Using Visual Studio 2017 and Visual Studio Code
    Porting from .NET Framework to .NET Core

        Could you port?

        Should you port?

        Differences between .NET Framework and .NET Core

        Understanding the .NET Portability Analyzer

        Using non-.NET Standard libraries
    Practicing and exploring

        Exercise 7.1 – Test your knowledge

        Exercise 7.2 – Explore topics

    Summary
10. Using Common .NET Standard Types

Working with numbers
- Working with big integers
- Working with complex numbers

Working with text
- Getting the length of a string
- Getting the characters of a string
- Splitting a string
- Getting part of a string
- Checking a string for content
- Other string members
- Building strings efficiently
- Pattern matching with regular expressions
  - The syntax of a regular expression
  - Examples of regular expressions

Working with collections
- Common features of all collections
- Understanding collections
  - Lists
  - Dictionaries
  - Stacks
  - Queues
  - Sets
- Working with lists
- Working with dictionaries
- Sorting collections
- Using specialized collections
- Using immutable collections

Working with network resources
- Working with URIs, DNS, and IP addresses
- Pinging a server

Working with types and attributes
- Versioning of assemblies
- Reading assembly metadata
- Creating custom attributes
- Doing more with reflection

Internationalizing your code
- Globalizing an application

Practicing and exploring
- Exercise 8.1 – Test your knowledge

Exercise 8.2 – Practice regular expressions

Exercise 8.3 – Practice writing extension methods

Exercise 8.4 – Explore topics

Summary

11. Working with Files, Streams, and Serialization

Managing the filesystem

Handling cross-platform environments and filesystems

Using Windows 10

Using macOS

Managing drives

Managing directories

Managing files

Managing paths

Getting file information

Controlling files

Reading and writing with streams

Writing to text and XML streams

Writing to text streams

Writing to XML streams

Disposing of file resources

Implementing disposal with try statement

Simplifying disposal with the using statement

Compressing streams

Encoding text

Encoding strings as byte arrays

Encoding and decoding text in files

Serializing object graphs

Serializing with XML

Deserializing with XML

Customizing the XML

Serializing with JSON

Serializing with other formats

Practicing and exploring

Exercise 9.1 – Test your knowledge

Exercise 9.2 – Practice serializing as XML

Exercise 9.3 – Explore topics

Summary

12. Protecting Your Data and Applications

Understanding the vocabulary of protection

Keys and key sizes
IVs and block sizes
Salts
Generating keys and IVs
Encrypting and decrypting data
Encrypting symmetrically with AES
Using Visual Studio 2017
Using Visual Studio Code
Creating the Protector class
Hashing data
Hashing with the commonly used SHA256
Signing data
Signing with SHA256 and RSA
Testing the signing and validating
Generating random numbers
Generating random numbers for games
Generating random numbers for cryptography
Testing the random key or IV generation
Authenticating and authorizing users
Implementing authentication and authorization
Testing authentication and authorization
Protecting application functionality
Practicing and exploring
Exercise 10.1 – Test your knowledge
Exercise 10.2 – Practice protecting data with encryption and hashing
Exercise 10.3 – Practice protecting data with decryption
Exercise 10.4 – Explore topics
Summary

13. Working with Databases Using Entity Framework Core
Understanding modern databases
Using a sample relational database
Using Microsoft SQL Server
Connecting to SQL Server
Creating the Northwind sample database for SQL Server
Managing the Northwind sample database with Server Explorer
Using SQLite
Creating the Northwind sample database for SQLite
Managing the Northwind sample database with SQLiteStudio
Setting up Entity Framework Core
Choosing an EF Core data provider
Connecting to the database

- Using Visual Studio 2017
- Using Visual Studio Code

Defining Entity Framework Core models

- EF Core conventions
- EF Core annotation attributes
- EF Core Fluent API

Building an EF Core model

- Defining the Category entity class
- Defining the Product entity class
- Defining the Northwind database context class

Querying an EF Core model

- Logging EF Core
- Pattern matching with Like
- Defining global filters

Loading patterns with EF Core

- Eager and lazy loading entities
- Explicit loading entities

Manipulating data with EF Core

- Inserting entities
- Updating entities
- Deleting entities
- Pooling database contexts
- Transactions
- Defining an explicit transaction

Practicing and exploring

- Exercise 11.1 – Test your knowledge
- Exercise 11.2 – Practice exporting data using different serialization formats
- Exercise 11.3 – Explore the EF Core documentation

Summary

14. Querying and Manipulating Data Using LINQ

Writing LINQ queries

- Extending sequences with the enumerable class

Filtering entities with Where

- Targeting a named method
- Simplifying the code by removing the explicit delegate instantiation
- Targeting a lambda expression

Sorting entities

        - Sorting by a single property using OrderBy
        - Sorting by a subsequent property using ThenBy
    - Filtering by type
- Working with sets
- Using LINQ with EF Core
    - Projecting entities with Select
    - Building an EF Core model
    - Joining and grouping
    - Aggregating sequences
- Sweetening the syntax with syntactic sugar
- Using multiple threads with parallel LINQ
- Creating your own LINQ extension methods
- Working with LINQ to XML
    - Generating XML using LINQ to XML
    - Reading XML using LINQ to XML
- Practicing and exploring
    - Exercise 12.1 – Test your knowledge
    - Exercise 12.2 – Practice querying with LINQ
    - Exercise 12.3 – Explore topics
- Summary

15. Improving Performance and Scalability Using Multitasking
- Monitoring performance and resource usage
    - Evaluating the efficiency of types
    - Monitoring performance and memory use
        - Using Visual Studio 2017
        - Using Visual Studio Code
        - Creating the Recorder class
        - Measuring the efficiency of processing strings
- Understanding processes, threads, and tasks
- Running tasks asynchronously
    - Running multiple actions synchronously
    - Running multiple actions asynchronously using tasks
    - Waiting for tasks
    - Continuing with another task
    - Nested and child tasks
- Synchronizing access to shared resources
    - Accessing a resource from multiple threads
    - Applying a mutually exclusive lock to a resource

Understanding the lock statement

Making operations atomic

Applying other types of synchronization

Understanding async and await

Improving responsiveness for console apps

Improving responsiveness for GUI apps

Improving scalability for web applications and web services

Common types that support multitasking

await in catch blocks

Practicing and exploring

Exercise 13.1 – Test your knowledge

Exercise 13.2 – Explore topics

Summary

16. Part 3 – App Models

17. Building Web Sites Using ASP.NET Core Razor Pages

Understanding web development

Understanding HTTP

Client-side web development

Understanding ASP.NET Core

Classic ASP.NET versus modern ASP.NET Core

Creating an ASP.NET Core project with Visual Studio 2017

Creating an ASP.NET Core project with Visual Studio Code

Reviewing the ASP.NET Core Empty project template

Testing the empty website

Enabling static files

Enabling default files

Exploring Razor Pages

Enabling Razor Pages

Defining a Razor Page

Using shared layouts with Razor Pages

Setting a shared layout

Defining a shared layout

Using code-behind files with Razor Pages

Using Entity Framework Core with ASP.NET Core

Creating Entity models for Northwind

Creating a class library for the Northwind entity classes

Defining the entity classes

Creating a class library for Northwind database context

Using Visual Studio 2017

Using Visual Studio Code

Defining the database context class

Creating the Northwind database in the website

Configure Entity Framework Core as a service

Manipulating data
Practicing and exploring

Exercise 14.1 – Practice building a data-driven website

Exercise 14.2 – Explore topics

Summary

18. Building Web Sites Using ASP.NET Core MVC
Setting up an ASP.NET Core MVC website
Creating an ASP.NET Core MVC website

Using Visual Studio 2017

Using Visual Studio Code

Reviewing the ASP.NET Core MVC project template
Performing database migrations

Using Visual Studio 2017

Using Visual Studio Code

Testing the ASP.NET MVC website

Reviewing authentication with ASP.NET Identity
Understanding an ASP.NET Core MVC website

ASP.NET Core startup

Understanding the default route

Understanding ASP.NET Core MVC controllers
Understanding ASP.NET Core MVC models

Configuring an EF Core entity data model

Creating view models for requests

Fetch the model in the controller
Understanding ASP.NET Core MVC views

Rendering the Home controller's views

Sharing layouts between views

Defining custom styles

Defining a typed view

Passing parameters using a route value

Passing parameters using a query string
Practicing and exploring

Exercise 15.1 – Practice improving scalability by understanding and im

plementing async action methods

Exercise 15.2 – Explore topics

Summary
19. Building Web Services and Applications Using ASP.NET Core
Building web services using ASP.NET Core Web API
Understanding ASP.NET Core controllers
Creating an ASP.NET Core Web API project
Using Visual Studio 2017
Using Visual Studio Code
Using Visual Studio 2017 and Visual Studio Code
Creating a web service for the Northwind database
Using Visual Studio 2017
Using Visual Studio Code
Using Visual Studio 2017 and Visual Studio Code
Creating data repositories for entities
Configuring and registering the customers repository
Creating the Web API controller
Documenting and testing web services using Swagger
Testing GET requests with any browser
Testing POST, PUT, and DELETE requests with Swagger
Installing a Swagger package
Using Visual Studio 2017
Using Visual Studio Code
Using Visual Studio 2017 and Visual Studio Code
Testing GET requests with Swagger UI
Testing POST requests with Swagger UI
Building SPAs using Angular
Understanding the Angular project template
Using Visual Studio 2017
Using Visual Studio Code
Using Visual Studio 2017 and Visual Studio Code
Calling NorthwindService
Using Visual Studio 2017
Using Visual Studio Code
Using Visual Studio 2017 and Visual Studio Code
Modifying the home component to call NorthwindService
Testing the Angular component calling the service
Using Visual Studio 2017
Using Visual Studio Code
Using Visual Studio 2017 and Visual Studio Code
Using other project templates
Installing additional template packs
Practicing and exploring

Exercise 16.1 – Practice with React and Redux

Exercise 16.2 – Explore topics

Summary

20. Building Windows Apps Using XAML and Fluent Design
Understanding the modern Windows platform

Understanding Universal Windows Platform
Understanding Fluent Design System

Filling user interface elements with acrylic brushes

Connecting user interface elements with animations

Parallax views and Reveal lighting
Understanding XAML Standard 1.0

Simplifying code using XAML

Choosing common controls
Creating a modern Windows app

Enabling developer mode

Creating a UWP project

Exploring common controls and acrylic brushes

Exploring Reveal

Installing more controls
Using resources and templates

Sharing resources

Replacing a control template
Data binding

Binding to elements
Binding to data sources

Modifying the NorthwindService

Creating the Northwind app
Building apps using Windows Template Studio

Installing Windows Template Studio

Selecting project types, frameworks, pages, and features

Retargeting the project

Customizing some views

Testing the app's functionality
Practicing and exploring

Exercise 17.1 – Explore topics

Summary

21. Building Mobile Apps Using XAML and Xamarin.Forms
Understanding Xamarin and Xamarin.Forms

How Xamarin.Forms extends Xamarin

Mobile first, cloud first

Building mobile apps using Xamarin.Forms

Adding Android SDKs

Creating a Xamarin.Forms solution

Creating a model
Creating an interface for dialing phone numbers

Implement the phone dialer for iOS

Implement the phone dialer for Android
Creating views for the customers list and customer details

Creating the view for the list of customers

Creating the view for the customer details

Testing the mobile app with iOS

Adding NuGet packages for calling a REST service

Getting customers from the service
Practicing and exploring

Exercise 18.1 - Explore topics

Summary

22. Summary

Good luck!

23. Answers to the Test Your Knowledge Questions

Chapter 1 – Hello, C#! Welcome, .NET Core!

Chapter 2 – Speaking C#

Chapter 3 – Controlling the Flow and Converting Types

Chapter 4 – Writing, Debugging, and Testing Functions

Chapter 5 – Building Your Own Types with Object-Oriented Programming

Chapter 6 – Implementing Interfaces and Inheriting Classes

Chapter 7 – Understanding and Packaging .NET Standard Types

Chapter 8 – Using Common .NET Standard Types

Chapter 9 – Working with Files, Streams, and Serialization

Chapter 10 – Protecting Your Data and Applications

Chapter 11 – Working with Databases Using Entity Framework Core

Chapter 12 – Querying and Manipulating Data Using LINQ

Chapter 13 – Improving Performance and Scalability Using Multitasking