



Pro C# 10 with .NET 6

Foundational Principles and Practices
in Programming

—

Eleventh Edition

—

Andrew Troelsen
Phil Japikse

Apress®

Pro C# 10 with .NET 6

Foundational Principles and Practices
in Programming

Eleventh Edition



Andrew Troelsen
Phil Japikse

Apress®

Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming

Andrew Troelsen
Chambersburg, PA, USA

Phil Japikse
West Chester, OH, USA

ISBN-13 (pbk): 978-1-4842-7868-0

ISBN-13 (electronic): 978-1-4842-7869-7

<https://doi.org/10.1007/978-1-4842-7869-7>

Copyright © 2022 by Andrew Troelsen, Phil Japikse

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Joan Murray
Development Editor: Laura Berendson
Coordinating Editor: Mark Powers
Copyeditor: Kim Wimpsett

Cover designed by eStudioCalamar

Cover image by Shutterstock (www.shutterstock.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub. For more detailed information, please visit www.apress.com/source-code.

Printed on acid-free paper

*To my family, Amy, Conner, Logan, and Skylar,
thank you for all of the support and patience you have given me.*

*Also, to my father (Cor);
father, husband, inventor, and my center of excellence.*

—Phil

Table of Contents

About the Authors	li
About the Technical Reviewers	liii
Acknowledgments	lv
Introduction	lvii
■ Part I: Introducing C# and .NET 6	1
■ Chapter 1: Introducing C# and .NET 6	3
Exploring Some Key Benefits of the .NET Platform	4
Understanding the .NET Support Lifecycle.....	4
Previewing the Building Blocks of the .NET Platform.....	5
The Role of the Base Class Libraries	5
What C# Brings to the Table.....	6
Managed vs. Unmanaged Code	7
Using Additional .NET–Aware Programming Languages	7
Getting an Overview of .NET Assemblies	8
The Role of the Common Intermediate Language	8
Benefits of CIL	11
Compiling CIL to Platform-Specific Instructions.....	11
The Role of .NET Type Metadata	12
The Role of the Assembly Manifest	13
Understanding the Common Type System.....	13
CTS Class Types.....	14
CTS Interface Types	14

CTS Structure Types	15
CTS Enumeration Types	15
CTS Delegate Types	16
CTS Type Members	16
Intrinsic CTS Data Types	16
Understanding the Common Language Specification	17
Ensuring CLS Compliance.....	19
Understanding the .NET Runtime	19
Distinguishing Between Assembly, Namespace, and Type	19
Accessing a Namespace Programmatically	22
Global Using Statements (New 10.0)	22
File Scoped Namespaces (New 10.0)	24
Referencing External Assemblies	24
Exploring an Assembly Using ildasm.exe	25
Summary.....	25
■ Chapter 2: Building C# Applications.....	27
Installing .NET 6	27
Understanding the .NET Version Numbering Scheme.....	27
Confirming the .NET 6 Install	28
Building .NET Core Applications with Visual Studio.....	29
Installing Visual Studio 2022 (Windows).....	30
Taking Visual Studio 2022 for a Test-Drive	31
Building .NET Core Applications with Visual Studio Code.....	44
Taking Visual Studio Code for a Test-Drive	44
Finding the .NET Core and C# Documentation	47
Summary.....	48
■ Part II: Core C# Programming.....	49
■ Chapter 3: Core C# Programming Constructs, Part 1	51
Breaking Down a Simple C# Program (Updated C# 10).....	51
Using Variations of the Main() Method (Updated 7.1)	53

Using Top-Level Statements (New 9.0).....	54
Specifying an Application Error Code (Updated 9.0).....	56
Processing Command-Line Arguments (Updated 9.0).....	58
Specifying Command-Line Arguments with Visual Studio 2022.....	60
Additional Members of the System.Environment Class (Updated 10.0).....	61
Using the System.Console Class	62
Performing Basic Input and Output (I/O) with the Console Class	63
Formatting Console Output.....	64
Formatting Numerical Data	65
Formatting Numerical Data Beyond Console Applications	66
Working with System Data Types and Corresponding C# Keywords	67
Understanding Variable Declaration and Initialization	68
Using Intrinsic Data Types and the new Operator (Updated 9.0)	70
Understanding the Data Type Class Hierarchy	71
Understanding the Members of Numerical Data Types	72
Understanding the Members of System.Boolean	73
Understanding the Members of System.Char	73
Parsing Values from String Data.....	74
Using TryParse to Parse Values from String Data	74
Using System.DateTime and System.TimeSpan (Updated 10.0)	75
Working with the System.Numerics Namespace	76
Using Digit Separators (New 7.0).....	77
Using Binary Literals (New 7.0/7.2).....	78
Working with String Data	78
Performing Basic String Manipulation.....	79
Performing String Concatenation	80
Using Escape Characters.....	81
Performing String Interpolation	82
Defining Verbatim Strings (Updated 8.0)	84
Working with Strings and Equality	85

Strings Are Immutable	87
Using the System.Text.StringBuilder Type	89
Narrowing and Widening Data Type Conversions	90
Using the checked Keyword	92
Setting Project-wide Overflow Checking (Project File).....	94
Setting Project-wide Overflow Checking (Visual Studio).....	94
Using the unchecked Keyword	95
Understanding Implicitly Typed Local Variables	96
Declaring Numerics Implicitly.....	97
Understanding Restrictions on Implicitly Typed Variables	98
Implicit Typed Data Is Strongly Typed Data.....	99
Understanding the Usefulness of Implicitly Typed Local Variables.....	100
Working with C# Iteration Constructs.....	100
Using the for Loop	101
Using the foreach Loop.....	101
Using Implicit Typing Within foreach Constructs	102
Using the while and do/while Looping Constructs	102
A Quick Discussion About Scope.....	103
Working with Decision Constructs and the Relational/Equality Operators.....	104
Using the if/else Statement	104
Using Equality and Relational Operators	104
Using if/else with Pattern Matching (New 7.0).....	106
Making Pattern Matching Improvements (New 9.0)	106
Using the Conditional Operator (Updated 7.2, 9.0)	108
Using Logical Operators.....	109
Using the switch Statement	109
Performing switch Statement Pattern Matching (New 7.0, Updated 9.0).....	112
Using switch Expressions (New 8.0).....	115
Summary	117

■ Chapter 4: Core C# Programming Constructs, Part 2	119
Understanding C# Arrays	119
Looking at the C# Array Initialization Syntax	120
Understanding Implicitly Typed Local Arrays	121
Defining an Array of Objects	122
Working with Multidimensional Arrays	122
Using Arrays As Arguments or Return Values	124
Using the System.Array Base Class.....	125
Using Indices and Ranges (New 8.0, Updated 10.0).....	126
Understanding Methods	128
Understanding Expression-Bodied Members	129
Understanding Local Functions (New 7.0, Updated 9.0).....	129
Understanding Static Local Functions (New 8.0).....	131
Understanding Method Parameters	131
Understanding Method Parameter Modifiers.....	131
Understanding the Default Parameter-Passing Behavior	132
Using the out Modifier (Updated 7.0).....	133
Using the ref Modifier	135
Using the in Modifier (New 7.2).....	136
Using the params Modifier	137
Defining Optional Parameters.....	138
Using Named Arguments (Updated 7.2).....	139
Understanding Method Overloading	141
Checking Parameters for Null (Updated 10.0)	144
Understanding the enum Type	144
Controlling the Underlying Storage for an enum	146
Declaring enum Variables	146
Using the System.Enum Type	147
Dynamically Discovering an enum's Name-Value Pairs	148
Using Enums, Flags, and Bitwise Operations	149

Understanding the Structure	151
Creating Structure Variables	153
Structure Constructors (Updated 10.0)	153
Using Field Initializers (New 10.0)	154
Using Read-Only Structs (New 7.2)	155
Using Read-Only Members (New 8.0)	156
Using ref Structs (New 7.2)	156
Using Disposable ref Structs (New 8.0)	157
Understanding Value Types and Reference Types	158
Using Value Types, Reference Types, and the Assignment Operator	159
Using Value Types Containing Reference Types	160
Passing Reference Types by Value	162
Passing Reference Types by Reference	163
Final Details Regarding Value Types and Reference Types	164
Understanding C# Nullable Types	165
Using Nullable Value Types	166
Using Nullable Reference Types (New 8.0, Updated 10.0)	167
Operating on Nullable Types	171
Understanding Tuples (New/Updated 7.0)	173
Getting Started with Tuples	173
Using Inferred Variable Names (Updated 7.1)	174
Understanding Tuple Equality/Inequality (New 7.3)	175
Understanding Tuples as Method Return Values	175
Understanding Discards with Tuples	176
Understanding Tuple Pattern Matching switch Expressions (New 8.0)	176
Deconstructing Tuples (Updated 10.0)	177
Summary	179
■ Part III: Object Oriented Programming with C#	181
■ Chapter 5: Understanding Encapsulation	183
Introducing the C# Class Type	183
Allocating Objects with the new Keyword	185

Understanding Constructors	186
Understanding the Role of the Default Constructor	186
Defining Custom Constructors.....	187
Understanding the Default Constructor Revisited	188
Understanding the Role of the this Keyword	190
Chaining Constructor Calls Using this	191
Observing Constructor Flow	193
Revisiting Optional Arguments	195
Understanding the static Keyword	196
Defining Static Field Data	197
Defining Static Methods	199
Defining Static Constructors.....	200
Defining Static Classes.....	202
Importing Static Members via the C# using Keyword	203
Defining the Pillars of OOP	204
Understanding the Role of Encapsulation.....	204
Understanding the Role of Inheritance	205
Understanding the Role of Polymorphism	206
Understanding C# Access Modifiers (Updated 7.2)	207
Using the Default Access Modifiers	208
Using Access Modifiers and Nested Types	209
Understanding the First Pillar: C#'s Encapsulation Services	209
Encapsulation Using Traditional Accessors and Mutators	210
Encapsulation Using Properties.....	212
Using Properties Within a Class Definition.....	216
Read-Only Properties.....	217
Write-Only Properties	218
Mixing Private and Public Get/Set Methods on Properties	218
Revisiting the static Keyword: Defining Static Properties	218
Pattern Matching with Property Patterns (New 8.0).....	219
Extended Property Patterns (New 10.0).....	221

Understanding Automatic Properties	222
Interacting with Automatic Properties	223
Automatic Properties and Default Values	224
Initializing Automatic Properties.....	225
Understanding Object Initialization	226
Looking at the Object Initialization Syntax	226
Using init-Only Setters (New 9.0)	227
Calling Custom Constructors with Initialization Syntax	228
Initializing Data with Initialization Syntax.....	230
Working with Constant and Read-Only Field Data	231
Understanding Constant Field Data	231
Understanding Read-Only Fields	232
Understanding Static Read-Only Fields	233
Understanding Partial Classes	234
Records (New 9.0)	235
Immutable Record Types with Standard Property Syntax.....	237
Immutable Record Types with Positional Syntax	238
Mutable Record Types	239
Value Equality with Record Types	239
Copying Record Types Using with Expressions.....	240
Record Structs (New 10.0)	241
Mutable Record Structs	241
Immutable Record Structs.....	242
Deconstructing Record Structs.....	243
Summary	243
■ Chapter 6: Understanding Inheritance and Polymorphism	245
Understanding the Basic Mechanics of Inheritance	245
Specifying the Parent Class of an Existing Class.....	246
Regarding Multiple Base Classes	248
Using the sealed Keyword	248

Revisiting Visual Studio Class Diagrams	249
Understanding the Second Pillar of OOP: The Details of Inheritance	251
Calling Base Class Constructors with the base Keyword	252
Keeping Family Secrets: The protected Keyword	254
Adding a sealed Class	255
Understanding Inheritance with Record Types (New 9.0).....	256
Programming for Containment/Delegation.....	260
Understanding Nested Type Definitions.....	261
Understanding the Third Pillar of OOP: C#'s Polymorphic Support.....	263
Using the virtual and override Keywords.....	264
Overriding Virtual Members with Visual Studio/Visual Studio Code	267
Sealing Virtual Members (Updated 10.0).....	267
Understanding Abstract Classes.....	268
Understanding the Polymorphic Interface	270
Understanding Member Shadowing	274
Understanding Base Class/Derived Class Casting Rules.....	275
Using the C# as Keyword.....	277
Using the C# is Keyword (Updated 7.0, 9.0)	279
Revisiting Pattern Matching (New 7.0).....	281
Understanding the Super Parent Class: System.Object.....	282
Overriding System.Object.ToString()	285
Overriding System.Object.Equals()	285
Overriding System.Object.GetHashCode()	286
Testing Your Modified Person Class.....	287
Using the Static Members of System.Object	288
Summary.....	288
■ Chapter 7: Understanding Structured Exception Handling	289
Ode to Errors, Bugs, and Exceptions	289
The Role of .NET Exception Handling	290
The Building Blocks of .NET Exception Handling	291
The System.Exception Base Class.....	291

The Simplest Possible Example	292
Throwing a General Exception.....	295
Catching Exceptions	296
Throw As Expression (New 7.0).....	297
Configuring the State of an Exception.....	298
The TargetSite Property	298
The StackTrace Property.....	299
The HelpLink Property	299
The Data Property	300
System-Level Exceptions (System.SystemException).....	302
Application-Level Exceptions (System.ApplicationException).....	302
Building Custom Exceptions, Take 1	303
Building Custom Exceptions, Take 2.....	305
Building Custom Exceptions, Take 3.....	305
Processing Multiple Exceptions	306
General catch Statements	309
Rethrowing Exceptions.....	309
Inner Exceptions	310
The finally Block	311
Exception Filters	312
Debugging Unhandled Exceptions Using Visual Studio	312
Summary.....	314
■ Chapter 8: Working with Interfaces.....	315
Understanding Interface Types.....	315
Interface Types vs. Abstract Base Classes.....	316
Defining Custom Interfaces.....	319
Implementing an Interface	320
Invoking Interface Members at the Object Level.....	322
Obtaining Interface References: The as Keyword.....	323
Obtaining Interface References: The is Keyword (Updated 7.0)	324

Default Implementations (New 8.0).....	324
Static Constructors and Members (New 8.0)	325
Interfaces as Parameters	326
Interfaces as Return Values.....	328
Arrays of Interface Types.....	328
Implementing Interfaces Using Visual Studio or Visual Studio Code	330
Explicit Interface Implementation	331
Designing Interface Hierarchies	334
Interface Hierarchies with Default Implementations (New 8.0).....	335
Multiple Inheritance with Interface Types.....	337
The IEnumerable and IEnumerator Interfaces	339
Building Iterator Methods with the yield Keyword.....	342
Building a Named Iterator.....	344
The ICloneable Interface.....	345
A More Elaborate Cloning Example.....	347
The IComparable Interface	350
Specifying Multiple Sort Orders with IComparer	353
Custom Properties and Custom Sort Types.....	354
Summary.....	355
■ Chapter 9: Understanding Object Lifetime.....	357
Classes, Objects, and References	357
The Basics of Object Lifetime.....	359
The CIL of new	359
Setting Object References to null	361
Determining If an Object Is Live	361
Understanding Object Generations.....	363
Ephemeral Generations and Segments	365
Garbage Collection Types	365
Background Garbage Collection	365

The System.GC Type	366
Forcing a Garbage Collection.....	367
Building Finalizable Objects	370
Overriding System.Object.Finalize().....	371
Detailing the Finalization Process	373
Building Disposable Objects.....	373
Reusing the C# using Keyword.....	375
Using Declarations (New 8.0)	377
Building Finalizable and Disposable Types.....	378
A Formalized Disposal Pattern	378
Understanding Lazy Object Instantiation.....	380
Customizing the Creation of the Lazy Data.....	383
Summary.....	384
■ Part IV: Advanced C# Programming	385
■ Chapter 10: Collections and Generics.....	387
The Motivation for Collection Classes	387
The System.Collections Namespace	389
A Survey of System.Collections.Specialized Namespace.....	390
The Problems of Nongeneric Collections	391
The Issue of Performance.....	392
The Issue of Type Safety.....	395
A First Look at Generic CollectionsT	398
The Role of Generic Type Parameters	399
Specifying Type Parameters for Generic Classes/Structures	400
Specifying Type Parameters for Generic Members.....	402
Specifying Type Parameters for Generic Interfaces.....	402
The System.Collections.Generic Namespace	403
Understanding Collection Initialization Syntax	405
Working with the List<T> Class	406
Working with the Stack<T> Class.....	408

Working with the Queue<T> Class.....	409
Working with the PriorityQueue<TElement, TPriority> Class (New 10).....	410
Working with the SortedSet<T> Class	411
Working with the Dictionary<TKey, TValue> Class.....	413
The System.Collections.ObjectModel Namespace	414
Working with ObservableCollection<T>.....	414
Creating Custom Generic Methods.....	416
Inference of Type Parameters.....	418
Creating Custom Generic Structures and Classes.....	419
Default Value Expressions with Generics	420
Default Literal Expressions (New 7.1).....	422
Pattern Matching with Generics (New 7.1).....	422
Constraining Type Parameters.....	423
Examples of Using the where Keyword	423
The Lack of Operator Constraints	425
Summary.....	426
■ Chapter 11: Advanced C# Language Features.....	427
Understanding Indexer Methods	427
Indexing Data Using String Values.....	429
Overloading Indexer Methods.....	430
Indexers with Multiple Dimensions	431
Indexer Definitions on Interface Types.....	432
Understanding Operator Overloading	432
Overloading Binary Operators.....	433
What of the += and -= Operators?	435
Overloading Unary Operators.....	436
Overloading Equality Operators	437
Overloading Comparison Operators.....	437
Final Thoughts Regarding Operator Overloading.....	438

Understanding Custom Type Conversions	439
Recall: Numerical Conversions	439
Recall: Conversions Among Related Class Types	439
Creating Custom Conversion Routines	440
Additional Explicit Conversions for the Square Type	443
Defining Implicit Conversion Routines	443
Understanding Extension Methods	444
Defining Extension Methods	445
Invoking Extension Methods	446
Importing Extension Methods	447
Extending Types Implementing Specific Interfaces	447
Extension Method GetEnumerator Support (New 9.0)	449
Understanding Anonymous Types	450
Defining an Anonymous Type	451
The Internal Representation of Anonymous Types	452
The Implementation of ToString() and GetHashCode()	453
The Semantics of Equality for Anonymous Types	454
Anonymous Types Containing Anonymous Types	456
Working with Pointer Types	456
The unsafe Keyword	459
Working with the * and & Operators	461
An Unsafe (and Safe) Swap Function	462
Field Access via Pointers (the -> Operator)	463
The stackalloc Keyword	463
Pinning a Type via the fixed Keyword	464
The sizeof Keyword	465
Summary	465
■ Chapter 12: Delegates, Events, and Lambda Expressions	467
Understanding the Delegate Type	468
Defining a Delegate Type in C#	468
The System.MulticastDelegate and System.Delegate Base Classes	470

The Simplest Possible Delegate Example 472
 Investigating a Delegate Object..... 474

Sending Object State Notifications Using Delegates 475
 Enabling Multicasting 478
 Removing Targets from a Delegate's Invocation List..... 480
 Method Group Conversion Syntax 481

Understanding Generic Delegates 482
 The Generic Action<> and Func<> Delegates 483

Understanding C# Events 485
 The C# event Keyword 486
 Events Under the Hood 488
 Listening to Incoming Events..... 489
 Simplifying Event Registration Using Visual Studio 490
 Creating Custom Event Arguments 491
 The Generic EventHandler<T> Delegate 493

Understanding C# Anonymous Methods 494
 Accessing Local Variables 495
 Using static with Anonymous Methods (New 9.0) 496
 Discards with Anonymous Methods (New 9.0) 497

Understanding Lambda Expressions 498
 Dissecting a Lambda Expression..... 500
 Processing Arguments Within Multiple Statements..... 501
 Lambda Expressions with Multiple (or Zero) Parameters..... 502
 Using static with Lambda Expressions (New 9.0)..... 504
 Discards with Lambda Expressions (New 9.0) 504
 Retrofitting the CarEvents Example Using Lambda Expressions 505
 Lambdas and Expression-Bodied Members (Updated 7.0)..... 505

Summary 506

■ Chapter 13: LINQ to Objects	509
LINQ-Specific Programming Constructs	509
Implicit Typing of Local Variables	510
Object and Collection Initialization Syntax.....	511
Lambda Expressions.....	511
Extension Methods	512
Anonymous Types	513
Understanding the Role of LINQ	513
LINQ Expressions Are Strongly Typed	514
The Core LINQ Assemblies.....	514
Applying LINQ Queries to Primitive Arrays	514
Once Again, Using Extension Methods	516
Once Again, Without LINQ	517
Reflecting Over a LINQ Result Set	517
LINQ and Implicitly Typed Local Variables	519
LINQ and Extension Methods.....	520
The Role of Deferred Execution	521
The Role of Immediate Execution	523
Returning the Result of a LINQ Query	525
Returning LINQ Results via Immediate Execution	526
Applying LINQ Queries to Collection Objects	527
Accessing Contained Subobjects	528
Applying LINQ Queries to Nongeneric Collections	528
Filtering Data Using <code>OfType<T>()</code>	529
Investigating the C# LINQ Query Operators	530
Basic Selection Syntax	531
Obtaining Subsets of Data	532
Paging Data	533
Paging Data with Ranges (New 10.0)	534
Paging Data with Chunks (New 10.0)	535
Projecting New Data Types	535

Projecting to Different Data Types	537
Obtaining Counts Using Enumerable	537
Obtaining Nonenumerated Counts (New 10.0)	538
Reversing Result Sets.....	539
Sorting Expressions.....	539
LINQ As a Better Venn Diagramming Tool	539
Removing Duplicates.....	543
LINQ Aggregation Operations.....	544
The Internal Representation of LINQ Query Statements.....	544
Building Query Expressions with Query Operators (Revisited)	545
Building Query Expressions Using the Enumerable Type and Lambda Expressions.....	546
Building Query Expressions Using the Enumerable Type and Anonymous Methods	547
Building Query Expressions Using the Enumerable Type and Raw Delegates.....	548
Summary.....	549
■ Chapter 14: Processes, AppDomains, and Load Contexts	551
The Role of a Windows Process	551
The Role of Threads.....	552
Interacting with Processes Using .NET Core	553
Enumerating Running Processes.....	555
Investigating a Specific Process.....	556
Investigating a Process's Thread Set.....	557
Investigating a Process's Module Set.....	558
Starting and Stopping Processes Programmatically	560
Controlling Process Startup Using the ProcessStartInfo Class.....	561
Leveraging OS Verbs with ProcessStartInfo	562
Understanding .NET Application Domains.....	563
The System.AppDomain Class.....	564
Interacting with the Default Application Domain	564
Enumerating Loaded Assemblies	565
Assembly Isolation with Application Load Contexts	566

Summarizing Processes, AppDomains, and Load Contexts.....	569
Summary.....	570
■ Chapter 15: Multithreaded, Parallel, and Async Programming	571
The Process/AppDomain/Context/Thread Relationship.....	571
The Problem of Concurrency	572
The Role of Thread Synchronization	573
The System.Threading Namespace.....	573
The System.Threading.Thread Class	574
Obtaining Statistics About the Current Thread of Execution.....	575
The Name Property	576
The Priority Property.....	576
Manually Creating Secondary Threads.....	577
Working with the ThreadStart Delegate.....	577
Working with the ParameterizedThreadStart Delegate	579
The AutoResetEvent Class	580
Foreground Threads and Background Threads.....	581
The Issue of Concurrency	582
Synchronization Using the C# lock Keyword	584
Synchronization Using the System.Threading.Monitor Type.....	586
Synchronization Using the System.Threading.Interlocked Type	587
Programming with Timer Callbacks	588
Using a Stand-Alone Discard (New 7.0).....	589
Understanding the ThreadPool	590
Parallel Programming Using the Task Parallel Library	591
The System.Threading.Tasks Namespace	591
The Role of the Parallel Class	591
Data Parallelism with the Parallel Class	592
Accessing UI Elements on Secondary Threads.....	596
The Task Class	597
Handling Cancellation Request.....	597
Task Parallelism Using the Parallel Class	599

Parallel LINQ Queries (PLINQ).....	602
Opting in to a PLINQ Query	603
Cancelling a PLINQ Query	603
Async Calls Using the async/await Pattern.....	605
A First Look at the C# async and await Keywords (Updated 7.1, 9.0)	605
SynchronizationContext and async/await.....	607
The Role of ConfigureAwait	607
Naming Conventions for Asynchronous Methods	608
Async Methods That Don't Return Data.....	608
Async Methods with Multiple Awaits.....	611
Calling Async Methods from Synchronous Methods	613
Await in catch and finally Blocks.....	614
Generalized Async Return Types (New 7.0)	615
Local Functions with async/await (New 7.0).....	615
Cancelling async/await Operations	616
Asynchronous Streams (New 8.0)	619
The Parallel.ForEachAsync() Method (New 10.0).....	620
Update the Book Reader App with async/await.....	621
Wrapping Up async and await.....	622
Summary.....	622
■ Part V: Programming with .NET Core Assemblies	623
■ Chapter 16: Building and Configuring Class Libraries.....	625
Defining Custom Namespaces (Updated 10.0).....	625
Resolving Name Clashes with Fully Qualified Names	627
Resolving Name Clashes with Aliases	628
Creating Nested Namespaces	629
Change the Root Namespace Using Visual Studio 2022.....	630
Change the Root Namespace Using the Project File	630
The Role of .NET Assemblies.....	631
Assemblies Promote Code Reuse.....	631
Assemblies Establish a Type Boundary	632

Assemblies Are Versionable Units	632
Assemblies Are Self-Describing	632
Understanding the Format of a .NET Assembly	632
Installing the C++ Profiling Tools	633
The Operating System (Windows) File Header	633
The CLR File Header	634
CIL Code, Type Metadata, and the Assembly Manifest.....	635
Optional Assembly Resources	635
Class Libraries vs. Console Applications	636
.NET Standard vs. .NET (Core) Class Libraries.....	636
Configuring Applications with Configuration Files.....	637
Multiple Configuration Files.....	639
Working with Objects (Updated 10.0).....	639
Additional Configuration Options	642
Building and Consuming a .NET Class Library	643
Exploring the Manifest.....	645
Exploring the CIL.....	647
Exploring the Type Metadata	648
Building a C# Client Application.....	648
Building a Visual Basic Client Application.....	650
Cross-Language Inheritance in Action.....	651
Exposing internal Types to Other Assemblies	652
NuGet and .NET Core	653
Packaging Assemblies with NuGet	653
Referencing NuGet Packages	654
Publishing Console Applications (Updated .NET 5/6).....	655
Publishing Framework-Dependent Applications.....	656
Publishing Self-Contained Applications.....	656
How .NET Locates Assemblies	658
Summary	660

■ **Chapter 17: Type Reflection, Late Binding, Attribute, and Dynamic Types 661**

- The Necessity of Type Metadata..... 661**
 - Viewing (Partial) Metadata for the EngineStateEnum Enumeration 662
 - Viewing (Partial) Metadata for the Car Type 663
 - Examining a TypeRef 665
 - Documenting the Defining Assembly..... 665
 - Documenting Referenced Assemblies 665
 - Documenting String Literals 666
- Understanding Reflection 667**
 - The System.Type Class 667
 - Obtaining a Type Reference Using System.Object.GetType() 668
 - Obtaining a Type Reference Using typeof() 669
 - Obtaining a Type Reference Using System.Type.GetType() 669
- Building a Custom Metadata Viewer 670**
 - Reflecting on Methods..... 670
 - Reflecting on Fields and Properties..... 671
 - Reflecting on Implemented Interfaces..... 671
 - Displaying Various Odds and Ends 672
 - Adding the Top-Level Statements..... 672
 - Reflecting on Static Types 674
 - Reflecting on Generic Types 674
 - Reflecting on Method Parameters and Return Values 674
- Dynamically Loading Assemblies 676**
- Reflecting on Framework Assemblies 677**
- Understanding Late Binding 679**
 - The System.Activator Class 679
 - Invoking Methods with No Parameters..... 681
 - Invoking Methods with Parameters 681
- Understanding the Role of .NET Attributes 682**
 - Attribute Consumers..... 683
 - Applying Attributes in C# 683

C# Attribute Shorthand Notation.....	684
Specifying Constructor Parameters for Attributes.....	685
The Obsolete Attribute in Action.....	685
Building Custom Attributes.....	687
Applying Custom Attributes.....	687
Named Property Syntax.....	688
Restricting Attribute Usage.....	688
Assembly-Level Attributes.....	689
Using a Separate File for Assembly Attributes.....	690
Using the Project File for Assembly Attributes.....	690
Reflecting on Attributes Using Early Binding.....	691
Reflecting on Attributes Using Late Binding.....	692
Putting Reflection, Late Binding, and Custom Attributes in Perspective.....	694
Building an Extendable Application.....	694
Building the Multiproject ExtendableApp Solution.....	695
Building CommonSnappableTypes.dll.....	699
Building the C# Snap-In.....	699
Building the Visual Basic Snap-In.....	700
Adding the Code for the ExtendableApp.....	700
The Role of the C# dynamic Keyword.....	702
Calling Members on Dynamically Declared Data.....	704
The Scope of the dynamic Keyword.....	706
Limitations of the dynamic Keyword.....	706
Practical Uses of the dynamic Keyword.....	707
The Role of the Dynamic Language Runtime.....	707
The Role of Expression Trees.....	708
Dynamic Runtime Lookup of Expression Trees.....	708
Simplifying Late-Bound Calls Using Dynamic Types.....	709
Leveraging the dynamic Keyword to Pass Arguments.....	710
Summary.....	712

■ **Chapter 18: Understanding CIL and the Role of Dynamic Assemblies..... 713**

Motivations for Learning the Grammar of CIL 713

Examining CIL Directives, Attributes, and Opcodes 715

 The Role of CIL Directives..... 715

 The Role of CIL Attributes 715

 The Role of CIL Opcodes..... 716

 The CIL Opcode/CIL Mnemonic Distinction..... 716

Pushing and Popping: The Stack-Based Nature of CIL..... 717

Understanding Round-Trip Engineering 718

 The Role of CIL Code Labels 721

 Interacting with CIL: Modifying an *.il File 721

 Compiling CIL Code with ILASM.EXE 722

 Compiling CIL Code with Microsoft.NET.Sdk.il Projects 722

Understanding CIL Directives and Attributes..... 724

 Specifying Externally Referenced Assemblies in CIL..... 724

 Defining the Current Assembly in CIL 725

 Defining Namespaces in CIL..... 726

 Defining Class Types in CIL..... 726

 Defining and Implementing Interfaces in CIL..... 727

 Defining Structures in CIL..... 728

 Defining Enums in CIL..... 728

 Defining Generics in CIL 729

 Compiling the CILTypes.il File 730

.NET Base Class Library, C#, and CIL Data Type Mappings..... 730

Defining Type Members in CIL..... 731

 Defining Field Data in CIL 731

 Defining Type Constructors in CIL..... 732

 Defining Properties in CIL 732

 Defining Member Parameters..... 733

Examining CIL Opcodes	734
The .maxstack Directive	736
Declaring Local Variables in CIL	736
Mapping Parameters to Local Variables in CIL	737
The Hidden this Reference	738
Representing Iteration Constructs in CIL	738
The Final Word on CIL	739
Understanding Dynamic Assemblies	739
Exploring the System.Reflection.Emit Namespace.....	740
The Role of the System.Reflection.Emit.ILGenerator	741
Emitting a Dynamic Assembly	742
Emitting the Assembly and Module Set.....	744
The Role of the ModuleBuilder TypeC.....	744
Emitting the HelloClass Type and the String Member Variable.....	745
Emitting the Constructors.....	746
Emitting the SayHello() Method	746
Using the Dynamically Generated Assembly	747
Summary	748
■ Part VI: File Handling, Object Serialization, and Data Access.....	749
■ Chapter 19: File I/O and Object Serialization.....	751
Exploring the System.IO Namespace	751
The DirectoryInfo(Info) and File(Info) Types.....	752
The Abstract FileSystemInfo Base Class	753
Working with the DirectoryInfo Type	753
Enumerating Files with the DirectoryInfo Type.....	755
Creating Subdirectories with the DirectoryInfo Type	756
Working with the Directory Type	757
Working with the DriveInfo Class Type	758
Working with the FileInfo Class.....	759
The FileInfo.Create() Method	760

The FileInfo.Open() Method.....	761
The FileInfo.OpenRead() and FileInfo.OpenWrite() Methods	762
The FileInfo.OpenText() Method	763
The FileInfo.CreateText() and FileInfo.AppendText() Methods.....	763
Working with the File Type	764
Additional File-centric Members	764
The Abstract Stream Class	765
Working with FileStreams	766
Working with StreamWriters and StreamReaders.....	768
Writing to a Text File	768
Reading from a Text File	769
Directly Creating StreamWriter/StreamReader Types.....	770
Working with StringWriters and StringReaders	771
Working with BinaryWriters and BinaryReaders	772
Watching Files Programmatically	774
Understanding Object Serialization	776
The Role of Object Graphs	776
Creating the Sample Types and Top-Level Statements.....	777
Extensible Markup Language (XML)	779
JavaScript Object Notation (JSON) Serialization	784
Summary.....	796
■ Chapter 20: Data Access with ADO.NET	797
ADO.NET vs. ADO	797
Understanding ADO.NET Data Providers.....	798
ADO.NET Data Providers	799
The Types of the System.Data Namespace	800
The Role of the IDbConnection Interface	801
The Role of the IDbTransaction Interface.....	801
The Role of the IDbCommand Interface.....	801
The Role of the IDbDataParameter and IDataParameter Interfaces	802

The Role of the IDbDataAdapter and IDataAdapter Interfaces.....	803
The Role of the IDataReader and IDataRecord Interfaces	803
Abstracting Data Providers Using Interfaces.....	804
Setting Up SQL Server and Azure Data Studio	807
Installing SQL Server	807
Installing a SQL Server IDE.....	809
Connecting to SQL Server.....	810
Restoring the AutoLot Database Backup	812
Copying the Backup File to Your Container.....	813
Restoring the Database with SSMS.....	813
Restoring the Database with Azure Data Studio.....	815
Creating the AutoLot Database.....	816
Creating the Database	817
Creating the Tables	817
Creating the Table Relationships	819
Creating the GetPetName() Stored Procedure	821
Adding Test Records	821
The ADO.NET Data Provider Factory Model	823
A Complete Data Provider Factory Example	824
A Potential Drawback with the Data Provider Factory Model.....	828
Diving Deeper into Connections, Commands, and DataReaders	828
Working with Connection Objects.....	829
Working with Command Objects	832
Working with Data Readers	833
Working with Create, Update, and Delete Queries.....	835
Create the Car and CarViewModel Classes.....	836
Adding the InventoryDal Class.....	836
Adding the Deletion Logic.....	841
Adding the Update Logic.....	841