

THE ADDISON-WESLEY MICROSOFT TECHNOLOGY SERIES



ESSENTIAL C# 7.0



"This book has been a classic for years, and remains as one of the most venerable and trusted titles in the world of C# content, and probably far beyond!"

—Mads Torgersen

MARK MICHAELIS

ERIC LIPPERT, *Technical Editor*

Foreword by **MADS TORGENSEN**,
C# Program Manager, Microsoft



IntelliTect

Essential C# 7.0

Essential C# 7.0

Table of Contents

Cover

Title Page

Copyright Page

Contents

Figures

Tables

Foreword

Preface

Acknowledgments

About the Author

1 Introducing C#

- Hello, World

- C# Syntax Fundamentals

- Working with Variables

- Console Input and Output

- Managed Execution and the Common Language Infrastructure

- Multiple .NET Frameworks

2 Data Types

- Fundamental Numeric Types

- More Fundamental Types

- null and void

- Conversions between Data Types

Table of Contents

3 More with Data Types

- Categories of Types

- Nullable Modifier

- Tuples

- Arrays

4 Operators and Control Flow

- Operators

- Introducing Flow Control

- Code Blocks ({})

- Code Blocks, Scopes, and Declaration Spaces

- Boolean Expressions

- Bitwise Operators (<<, >>, |, &, ^, ~)

- Control Flow Statements, Continued

- Jump Statements

- C# Preprocessor Directives

5 Methods and Parameters

- Calling a Method

- Declaring a Method

- The using Directive

- Returns and Parameters on Main()

- Advanced Method Parameters

- Recursion

- Method Overloading

- Optional Parameters

- Basic Error Handling with Exceptions

6 Classes

- Declaring and Instantiating a Class

Table of Contents

- Instance Fields
- Instance Methods
- Using the this Keyword
- Access Modifiers
- Properties
- Constructors
- Static Members
- Extension Methods
- Encapsulating the Data
- Nested Classes
- Partial Classes

7 Inheritance

- Derivation
- Overriding the Base Class
- Abstract Classes
- All Classes Derive from System.Object
- Verifying the Underlying Type with the is Operator
- Pattern Matching with the is Operator
- Pattern Matching within a switch Statement
- Conversion Using the as Operator

8 Interfaces

- Introducing Interfaces
- Polymorphism through Interfaces
- Interface Implementation
- Converting between the Implementing Class and Its Interfaces
- Interface Inheritance
- Multiple Interface Inheritance

Table of Contents

Extension Methods on Interfaces

Implementing Multiple Inheritance via Interfaces

Versioning

Interfaces Compared with Classes

Interfaces Compared with Attributes

9 Value Types

Structs

Boxing

Enums

10 Well-Formed Types

Overriding object Members

Operator Overloading

Referencing Other Assemblies

Defining Namespaces

XML Comments

Garbage Collection

Resource Cleanup

Lazy Initialization

11 Exception Handling

Multiple Exception Types

Catching Exceptions

General Catch Block

Guidelines for Exception Handling

Defining Custom Exceptions

Rethrowing a Wrapped Exception

12 Generics

C# without Generics

Table of Contents

Introducing Generic Types

Constraints

Generic Methods

Covariance and Contravariance

Generic Internals

13 Delegates and Lambda Expressions

Introducing Delegates

Declaring Delegate Types

Lambda Expressions

Anonymous Methods

14 Events

Coding the Publish-Subscribe Pattern with Multicast Delegates

Understanding Events

15 Collection Interfaces with Standard Query Operators

Collection Initializers

What Makes a Class a Collection: `IEnumerable<T>`

Standard Query Operators

Anonymous Types with LINQ

16 LINQ with Query Expressions

Introducing Query Expressions

Query Expressions Are Just Method Invocations

17 Building Custom Collections

More Collection Interfaces

Primary Collection Classes

Providing an Indexer

Returning Null or an Empty Collection

Iterators

Table of Contents

18 Reflection, Attributes, and Dynamic Programming

Reflection

nameof Operator

Attributes

Programming with Dynamic Objects

19 Multithreading

Multithreading Basics

Working with System.Threading

Asynchronous Tasks

Canceling a Task

The Task-based Asynchronous Pattern

Executing Loop Iterations in Parallel

Running LINQ Queries in Parallel

20 Thread Synchronization

Why Synchronization?

Timers

21 Platform Interoperability and Unsafe Code

Platform Invoke

Pointers and Addresses

Executing Unsafe Code via a Delegate

22 The Common Language Infrastructure

Defining the Common Language Infrastructure

CLI Implementations

.NET Standard

Base Class Library

C# Compilation to Machine Code

Runtime

Table of Contents

Assemblies, Manifests, and Modules

Common Intermediate Language

Common Type System

Common Language Specification

Metadata

.NET Native and Ahead of Time Compilation

Index

Index of 7.0 Topics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

Table of Contents

V

Index of 6.0 Topics

A

B

C

D

E

F

I

M

N

O

P

Q

R

S

T

U

V

W

X

Index of 5.0 Topics

A

C

D

E

F

Table of Contents

I
L
M
N
O
R
S
T
U
W
X