

# Contents

<i>Foreword</i> .....	<i>xix</i>
<i>Dedication</i> .....	<i>xxi</i>
<i>Acknowledgments</i> .....	<i>xxi</i>
<i>About This Book</i> .....	<i>xxii</i>
<i>Introduction</i> .....	<i>xxiii</i>
<i>The C# Compiler and Other Resources</i> .....	<i>xxvii</i>
 <i>Chapter 1 Object-Oriented Basics</i> .....	 <i>1</i>
<i>What Is an Object?</i> .....	<i>1</i>
<i>Inheritance</i> .....	<i>1</i>
<i>Containment</i> .....	<i>2</i>
<i>Polymorphism and Virtual Functions</i> .....	<i>3</i>
<i>Encapsulation and Visibility</i> .....	<i>4</i>
 <i>Chapter 2 The .NET Runtime Environment</i> .....	 <i>7</i>
<i>The Execution Environment</i> .....	<i>8</i>
A Simpler Programming Model .....	<i>8</i>
Safety and Security .....	<i>10</i>
Powerful Tools Support .....	<i>11</i>
Deployment, Packaging, and Support .....	<i>11</i>
<i>Metadata</i> .....	<i>11</i>
<i>Assemblies</i> .....	<i>12</i>
<i>Language Interop</i> .....	<i>13</i>
<i>Attributes</i> .....	<i>13</i>

<b>Chapter 3 C# QuickStart and Developing in C#</b>	<b>15</b>
Hello, Universe	15
Namespaces and Using	16
Namespaces and Assemblies	17
Basic Data Types	18
Classes, Structs, and Interfaces	19
Statements	20
Enums	20
Delegates and Events	20
Properties and Indexers	20
Attributes	21
Developing in C#	22
The Command-Line Compiler	22
Visual Studio.NET	22
Other Tools of Note	24
 <b>Chapter 4 Exception Handling</b>	 <b>27</b>
What's Wrong with Return Codes?	27
Trying and Catching	28
The Exception Hierarchy	28
Passing Exceptions on to the Caller	31
Caller Beware	31
Caller Confuse	31
Caller Inform	32
User-Defined Exception Classes	34
Finally	35
Efficiency and Overhead	37
Design Guidelines	37
 <b>Chapter 5 Classes 101</b>	 <b>39</b>
A Simple Class	39
Member Functions	41
ref and out Parameters	42
Overloading	45

<b>Chapter 6</b>	<b><i>Base Classes and Inheritance</i></b>	<b>47</b>
	<i>The Engineer Class</i>	47
	<i>Simple Inheritance</i>	48
	<i>Arrays of Engineers</i>	50
	<i>Virtual Functions</i>	54
	<i>Abstract Classes</i>	56
	<i>Sealed Classes and Methods</i>	60
<b>Chapter 7</b>	<b><i>Member Accessibility and Overloading</i></b>	<b>61</b>
	<i>Class Accessibility</i>	61
	<i>Using Internal on Members</i>	61
	<i>internal protected</i>	63
	<i>The Interaction of Class and Member Accessibility</i>	63
	<i>Method Overloading</i>	64
	<i>Method Hiding</i>	64
	<i>Better Conversions</i>	66
	<i>Variable-Length Parameter Lists</i>	67
<b>Chapter 8</b>	<b><i>Other Class Details</i></b>	<b>71</b>
	<i>Nested Classes</i>	71
	<i>Other Nesting</i>	72
	<i>Creation, Initialization, Destruction</i>	72
	<i>Constructors</i>	73
	<i>Initialization</i>	75
	<i>Destructors</i>	76
	<i>Managing Non-Memory Resources</i>	77
	<i>IDisposable and the Using Statement</i>	79
	<i>IDisposable and Longer-Lived Objects</i>	80
	<i>Static Fields</i>	81
	<i>Static Member Functions</i>	82
	<i>Static Constructors</i>	83
	<i>Constants</i>	84
	<i>Read-Only Fields</i>	84

<b>Chapter 9 Structs (Value Types)</b>	<b>89</b>
A Point Struct	89
Boxing and Unboxing	90
Structs and Constructors	91
Design Guidelines	92
Immutable Classes	93
 <b>Chapter 10 Interfaces</b>	 <b>95</b>
A Simple Example	95
Working with Interfaces	96
The as Operator	98
Interfaces and Inheritance	100
Design Guidelines	101
Multiple Implementation	101
Explicit Interface Implementation	103
Implementation Hiding	106
Interfaces Based on Interfaces	107
Interfaces and Structs	107
 <b>Chapter 11 Versioning</b>	 <b>109</b>
A Versioning Example	109
Designed for Versioning	111
Designing for Versioning	112
 <b>Chapter 12 Statements and Flow of Execution</b>	 <b>113</b>
Selection Statements	113
If	113
Switch	114
Iteration Statements	116
While	116
Do	117
For	118
Foreach	119
Jump Statements	121
Break	121
Continue	121
Goto	121
Return	121

<i>Other Statements</i> .....	122
lock .....	122
using .....	122
try/catch/finally .....	122
checked/unchecked .....	122
 <i>Chapter 13 Variable Scoping and Definite Assignment</i> .....	123
<i>Definite Assignment</i> .....	124
Definite Assignment and Arrays .....	126
 <i>Chapter 14 Operators and Expressions</i> .....	129
<i>Operator Precedence</i> .....	129
<i>Built-In Operators</i> .....	130
<i>User-Defined Operators</i> .....	131
<i>Numeric Promotions</i> .....	131
<i>Arithmetic Operators</i> .....	131
Unary Plus (+) over .....	131
Unary Minus (-) over .....	132
Bitwise Complement (~) over .....	132
Addition (+) over .....	132
Subtraction (-) over .....	133
Multiplication (*) over .....	133
Division (/) over .....	133
Remainder (%) over .....	133
Shift (<< and >>) over .....	133
Increment and Decrement (++ and --) over .....	133
<i>Relational and Logical Operators</i> .....	134
Logical Negation (!) over .....	134
Relational Operators over .....	135
Logical Operators over .....	136
Conditional Operator (?) .....	136
<i>Assignment Operators</i> .....	137
Simple Assignment .....	137
Compound Assignment .....	137
<i>Type Operators</i> .....	138
typeof .....	138
is .....	138
as .....	139
<i>Checked and Unchecked Expressions</i> .....	140

<b>Chapter 15 Conversions</b>	143
<i>Numeric Types</i>	143
Conversions and Member Lookup	144
Explicit Numeric Conversions	146
Checked Conversions	147
<i>Conversions of Classes (Reference Types)</i>	148
To the Base Class of an Object	148
To an Interface the Object Implements	150
To an Interface the Object Might Implement	150
From One Interface Type to Another	152
<i>Conversions of Structs (Value Types)</i>	153
 <b>Chapter 16 Arrays</b>	155
<i>Array Initialization</i>	155
<i>Multidimensional and Jagged Arrays</i>	155
Multidimensional Arrays	156
Jagged Arrays	157
<i>Arrays of Reference Types</i>	159
<i>Array Conversions</i>	160
<i>The System.Array Type</i>	161
Sorting and Searching	161
Reverse	161
 <b>Chapter 17 Strings</b>	163
<i>Operations</i>	163
<i>String Encodings and Conversions</i>	165
<i>Converting Objects to Strings</i>	165
<i>An Example</i>	165
<i>String Interning</i>	166
<i>StringBuilder</i>	167
<i>Regular Expressions</i>	169
‘Regular Expression Options	170
More Complex Parsing	170
 <b>Chapter 18 Properties</b>	175
<i>Accessors</i>	176
<i>Properties and Inheritance</i>	176

<i>Use of Properties</i> .....	177
<i>Side Effects When Setting Values</i> .....	178
<i>Static Properties</i> .....	179
<i>Property Efficiency</i> .....	181
<i>Property Accessibility</i> .....	182
<i>Virtual Properties</i> .....	182

## ***Chapter 19 Indexers and Enumerators*** ..... 185

<i>Indexing with an Integer Index</i> .....	185
<i>Indexing with a String Index</i> .....	187
<i>Indexing with Multiple Parameters</i> .....	190
<i>Enumerators and Foreach</i> .....	192
Improving the Enumerator .....	196
<i>Disposable Enumerators</i> .....	198
GetEnumerator() Returns IEnumerator .....	198
GetEnumerator() Returns a Class That Implements IDisposable .....	199
GetEnumerator() Returns a Class That Doesn't Implement IDisposable .....	199
<i>Design Guidelines</i> .....	199

## ***Chapter 20 Enumerations*** ..... 201

<i>A Line Style Enumeration</i> .....	201
<i>Enumeration Base Types</i> .....	202
<i>Initialization</i> .....	203
<i>Bit Flag Enums</i> .....	204
<i>Conversions</i> .....	205
<i>The System.Enum Type</i> .....	206

## ***Chapter 21 Attributes*** ..... 209

<i>Using Attributes</i> .....	210
A Few More Details .....	212
<i>An Attribute of Your Own</i> .....	214
Attribute Usage .....	214
Attribute Parameters .....	215
<i>Reflecting on Attributes</i> .....	216

<b>Chapter 22 Delegates</b>	221
Using Delegates	221
Delegates to Instance Members	223
Multicasting	224
Delegates as Static Members	226
Delegates as Static Properties	228
 <b>Chapter 23 Events</b>	 231
Add and Remove Functions	233
Custom Add and Remove	236
 <b>Chapter 24 User-Defined Conversions</b>	 243
A Simple Example	243
Pre and Post Conversions	246
Conversions Between Structs	246
Classes and Pre and Post Conversions	252
Design Guidelines	258
Implicit Conversions Are Safe Conversions	259
Define the Conversion in the More Complex Type	259
One Conversion to and from a Hierarchy	259
Add Conversions Only as Needed	259
Conversions That Operate in Other Languages	259
How It Works	261
Conversion Lookup	262
 <b>Chapter 25 Operator Overloading</b>	 265
Unary Operators	265
Binary Operators	266
An Example	266
Restrictions	267
Guidelines	267
A Complex Number Class	268



<b>Chapter 26 Other Language Details</b>	273
<i>The Main Function</i>	273
Returning an Int Status	273
Command-Line Parameters	274
Multiple Mains	274
<i>Preprocessing</i>	275
Preprocessing Directives	276
Other Preprocessor Functions	277
<i>Lexical Details</i>	278
Identifiers	278
Literals	279
Comments	282
 <b>Chapter 27 Making Friends with             the .NET Frameworks</b>	 285
<i>Things All Objects Will Do</i>	285
ToString	285
Equals	287
<i>Hashes and GetHashCode()</i>	288
<i>Design Guidelines</i>	291
Value Type Guidelines	291
Reference Type Guidelines	292
 <b>Chapter 28 System.Array             and the Collection Classes</b>	 297
<i>Sorting and Searching</i>	297
Implementing IComparable	298
Using IComparer	299
IComparer as a Property	303
Overloading Relational Operators	305
Advanced Use of Hashes	307
<i>Synchronized Collections</i>	310
<i>Case-Insensitive Collections</i>	310
IClonable	311
<i>Other Collections</i>	313
<i>Design Guidelines</i>	314
Functions and Interfaces by Framework Class	314

<b>Chapter 29 Threading and Asynchronous Operations .....</b>	<b>317</b>
<i>Data Protection and Synchronization .....</i>	<i>317</i>
A Slightly Broken Example .....	317
Protection Techniques .....	322
<i>Access Reordering and Volatile .....</i>	<i>325</i>
Use of Volatile .....	328
<i>Threads .....</i>	<i>330</i>
Joining .....	330
Waiting with WaitHandle .....	331
Thread-Local Storage .....	333
<i>Asynchronous Calls .....</i>	<i>333</i>
A Simple Example .....	334
Return Values .....	336
Waiting for Completion .....	338
<i>Classes That Support Asynchronous Calls Directly .....</i>	<i>342</i>
<i>Design Guidelines .....</i>	<i>343</i>
 <b>Chapter 30 Execution-Time Code Generation .....</b>	 <b>345</b>
<i>Loading Assemblies .....</i>	<i>345</i>
Making It Dynamic .....	347
<i>Custom Code Generation .....</i>	<i>349</i>
Polynomial Evaluation .....	349
A Custom C# Class .....	355
A Fast Custom C# Class .....	358
A CodeDOM Implementation .....	359
A Reflection.Emit Implementation .....	363
<i>Summary .....</i>	<i>367</i>
 <b>Chapter 31 Interop .....</b>	 <b>369</b>
<i>Using COM Objects .....</i>	<i>369</i>
<i>Being Used by COM Objects .....</i>	<i>369</i>
<i>Calling Native DLL Functions .....</i>	<i>369</i>
Pointers and Declarative Pinning .....	370
Structure Layout .....	373
Calling a Function with a Structure Parameter .....	374
Hooking Up to a Windows Callback .....	375
Design Guidelines .....	377

<b>Chapter 32 .NET Frameworks Overview .....</b>	<b>379</b>
<i>Numeric Formatting .....</i>	<i>379</i>
Standard Format Strings .....	379
Custom Format Strings .....	385
<i>Date and Time Formatting .....</i>	<i>390</i>
<i>Custom Object Formatting .....</i>	<i>392</i>
<i>Numeric Parsing .....</i>	<i>393</i>
<i>Using XML in C# .....</i>	<i>393</i>
<i>InputOutput .....</i>	<i>394</i>
Binary .....	394
Text .....	395
XML .....	395
Reading and Writing Files .....	395
Traversing Directories .....	396
Starting Processes .....	398
<i>Serialization .....</i>	<i>399</i>
<i>Custom Serialization .....</i>	<i>402</i>
<i>Reading Web Pages .....</i>	<i>404</i>
<i>Accessing Environment Settings .....</i>	<i>406</i>
 <b>Chapter 33 Windows Forms .....</b>	 <b>409</b>
<i>Our Application .....</i>	<i>409</i>
<i>Getting Started .....</i>	<i>409</i>
<i>The Form Designer .....</i>	<i>412</i>
<i>Finding Directory Sizes .....</i>	<i>412</i>
Calculating Sizes .....	414
<i>Displaying the Directory Tree and Sizes .....</i>	<i>416</i>
<i>Setting the Directory .....</i>	<i>418</i>
<i>Tracking Our Progress .....</i>	<i>420</i>
 <b>Chapter 34 DiskDiff: More Sophistication .....</b>	 <b>425</b>
<i>Populating on a Thread .....</i>	<i>425</i>
<i>Interrupting a Thread .....</i>	<i>427</i>
A Cancel Button .....	428
<i>Decorating the TreeView .....</i>	<i>428</i>
Expand-o-Matic .....	430
Populate on Demand .....	431
<i>Sorting the Files .....</i>	<i>433</i>

<i>Saving and Restoring</i> .....	434
Controlling Serialization .....	436
Serialization Performance .....	438
Finer Control of Serialization .....	438
 <i>Chapter 35 DiskDiff: Making It Useful</i> .....	441
<i>Comparing Directories</i> .....	441
<i>File Manipulation</i> .....	442
<i>File and Directory Operations</i> .....	446
Delete .....	447
Delete Contents .....	447
View in Notepad and Launch .....	447
<i>Updating the User Interface</i> .....	447
A Bit of Refactoring .....	448
<i>Cleaning Up for the Parents</i> .....	448
Keyboard Accelerators .....	449
Most Recently Used List .....	449
Tooltips .....	451
<i>Increased Accuracy</i> .....	451
Switching to Use Cluster Size .....	453
 <i>Chapter 36 Deeper into C#</i> .....	455
<i>C# Style</i> .....	455
Naming .....	455
Encapsulation .....	456
<i>Guidelines for the Library Author</i> .....	456
CLS Compliance .....	456
Class Naming .....	457
<i>Unsafe Context</i> .....	457
<i>XML Documentation</i> .....	462
Compiler Support Tags .....	463
XML Documentation Tags .....	466
XML Include Files .....	467
<i>Garbage Collection in the .NET Runtime</i> .....	468
Allocation .....	468
Mark and Compact .....	468
Generations .....	469
Finalization .....	471
Controlling GC Behavior .....	471

<i>Deeper Reflection</i> .....	472
Listing All the Types in an Assembly .....	472
Finding Members .....	474
Invoking Functions .....	475
<i>Optimizations</i> .....	479
 <i>Chapter 37 Defensive Programming</i> .....	481
<i>Conditional Methods</i> .....	481
<i>Debug and Trace Classes</i> .....	482
<i>Asserts</i> .....	482
<i>Debug and Trace Output</i> .....	483
<i>Using Switches to Control Debug and Trace</i> .....	485
BooleanSwitch .....	485
TraceSwitch .....	487
User-Defined Switch .....	489
 <i>Chapter 38 The Command Line Compiler</i> .....	493
<i>Simple Usage</i> .....	493
<i>Response Files</i> .....	493
<i>Default Response File</i> .....	493
<i>Command-Line Options</i> .....	494
 <i>Chapter 39 C# Compared to Other Languages</i> .....	497
<i>Differences Between C# and C/C++</i> .....	497
A Managed Environment .....	497
.NET Objects .....	498
C# Statements .....	498
Attributes .....	499
Versioning .....	499
Code Organization .....	499
Missing C# Features .....	500
<i>Differences Between C# and Java</i> .....	500
Data Types .....	500
Extending the Type System .....	502
Classes .....	503
Interfaces .....	506
Properties and Indexers .....	506
Delegates and Events .....	506

Attributes .....	507
Statements .....	507
<i>Differences Between C# and Visual Basic 6</i> .....	509
Code Appearance .....	509
Data Types and Variables .....	510
Operators and Expressions .....	511
Classes, Types, Functions, and Interfaces .....	512
Control and Program Flow .....	512
Select Case .....	515
On Error .....	515
Missing Statements .....	515
<i>Other .NET Languages</i> .....	516
 <i>Chapter 40 C# Resources and Futures</i> .....	517
 <i>C# Resources</i> .....	517
MSDN .....	517
GotDotNet .....	517
Csharpindex .....	517
C-Sharp Corner .....	518
DotNet Books .....	518
<i>C# Futures</i> .....	518
 <i>Index</i> .....	521