

Join the discussion @ [p2p.wrox.com](http://p2p.wrox.com)



BUILD WINDOWS 8 STYLE APPS WITH C#



# Professional C# 2012 and .NET 4.5

Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner



# PROFESSIONAL C# 2012 AND .NET 4.5

---

INTRODUCTION.....	xlix
-------------------	------

## ► PART I THE C# LANGUAGE

CHAPTER 1	.NET Architecture.....	3
CHAPTER 2	Core C# .....	23
CHAPTER 3	Objects and Types.....	65
CHAPTER 4	Inheritance.....	89
CHAPTER 5	Generics.....	107
CHAPTER 6	Arrays and Tuples .....	129
CHAPTER 7	Operators and Casts.....	151
CHAPTER 8	Delegates, Lambdas, and Events .....	183
CHAPTER 9	Strings and Regular Expressions.....	209
CHAPTER 10	Collections .....	229
CHAPTER 11	Language Integrated Query.....	279
CHAPTER 12	Dynamic Language Extensions.....	313
CHAPTER 13	Asynchronous Programming .....	325
CHAPTER 14	Memory Management and Pointers.....	347
CHAPTER 15	Reflection.....	375
CHAPTER 16	Errors and Exceptions.....	393

## ► PART II VISUAL STUDIO

CHAPTER 17	Visual Studio 2012 .....	417
CHAPTER 18	Deployment.....	467

## ► PART III FOUNDATION

CHAPTER 19	Assemblies .....	487
CHAPTER 20	Diagnostics .....	519
CHAPTER 21	Tasks, Threads, and Synchronization.....	551

*Continued*

<b>CHAPTER 22</b>	Security .....	601
<b>CHAPTER 23</b>	Interop .....	627
<b>CHAPTER 24</b>	Manipulating Files and the Registry .....	661
<b>CHAPTER 25</b>	Transactions .....	705
<b>CHAPTER 26</b>	Networking .....	737
<b>CHAPTER 27</b>	Windows Services .....	771
<b>CHAPTER 28</b>	Localization .....	803
<b>CHAPTER 29</b>	Core XAML .....	845
<b>CHAPTER 30</b>	Managed Extensibility Framework .....	863
<b>CHAPTER 31</b>	Windows Runtime .....	893
 <b>► PART IV DATA</b>		
<b>CHAPTER 32</b>	Core ADO.NET .....	917
<b>CHAPTER 33</b>	ADO.NET Entity Framework .....	963
<b>CHAPTER 34</b>	Manipulating XML .....	995
 <b>► PART V PRESENTATION</b>		
<b>CHAPTER 35</b>	Core WPF .....	1049
<b>CHAPTER 36</b>	Business Applications with WPF .....	1101
<b>CHAPTER 37</b>	Creating Documents with WPF .....	1153
<b>CHAPTER 38</b>	Windows Store Apps .....	1175
<b>CHAPTER 39</b>	Core ASP.NET .....	1211
<b>CHAPTER 40</b>	ASP.NET Web Forms .....	1239
<b>CHAPTER 41</b>	ASP.NET MVC .....	1283
<b>CHAPTER 42</b>	ASP.NET Dynamic Data .....	1321
 <b>► PART VI COMMUNICATION</b>		
<b>CHAPTER 43</b>	Windows Communication Foundation .....	1337
<b>CHAPTER 44</b>	WCF Data Services .....	1379
<b>CHAPTER 45</b>	Windows Workflow Foundation .....	1399
<b>CHAPTER 46</b>	Peer-to-Peer Networking .....	1425
<b>CHAPTER 47</b>	Message Queuing .....	1439
<b>INDEX .....</b>		<b>1473</b>

PROFESSIONAL

**C# 2012 and .NET 4.5**



PROFESSIONAL

# **C# 2012 and .NET 4.5**

Christian Nagel  
Bill Evjen  
Jay Glynn  
Karli Watson  
Morgan Skinner



WILEY

John Wiley & Sons, Inc.

## **Professional C# 2012 and .NET 4.5**

Published by  
John Wiley & Sons, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256  
[www.wiley.com](http://www.wiley.com)

Copyright © 2013 by John Wiley & Sons, Inc., Indianapolis, Indiana

Published simultaneously in Canada

ISBN: 978-1-118-31442-5  
ISBN: 978-1-118-38800-6 (ebk)  
ISBN: 978-1-118-33212-2 (ebk)  
ISBN: 978-1-118-33538-3 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 646-8600. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

**Limit of Liability/Disclaimer of Warranty:** The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation warranties of fitness for a particular purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies contained herein may not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting, or other professional services. If professional assistance is required, the services of a competent professional person should be sought. Neither the publisher nor the author shall be liable for damages arising herefrom. The fact that an organization or Web site is referred to in this work as a citation and/or a potential source of further information does not mean that the author or the publisher endorses the information the organization or Web site may provide or recommendations it may make. Further, readers should be aware that Internet Web sites listed in this work may have changed or disappeared between when this work was written and when it is read.

For general information on our other products and services please contact our Customer Care Department within the United States at (877) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://book-support.wiley.com>. For more information about Wiley products, visit [www.wiley.com](http://www.wiley.com).

**Library of Congress Control Number:** 2012944687

**Trademarks:** Wiley, the Wiley logo, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.



*To my family – Angela, Stephanie, and Matthias – I  
love you all!*

—CHRISTIAN NAGEL

*This work is dedicated to my wife and son. They are  
my world.*

—JAY GLYNN

*Love is as strong as death;  
Many waters cannot quench love,  
Neither can the floods drown it.*

—MORGAN SKINNER



# ABOUT THE AUTHORS

**CHRISTIAN NAGEL** is a Microsoft Regional Director and Microsoft MVP, an associate of thinktecture, and founder of CN innovation. A software architect and developer, he offers training and consulting on how to develop solutions using the Microsoft platform. He draws on more than 25 years of software development experience. Christian started his computing career with PDP 11 and VAX/VMS systems, covering a variety of languages and platforms. Since 2000, when .NET was just a technology preview, he has been working with various .NET technologies to build .NET solutions. Currently, he mainly coaches the development of Windows Store apps accessing Windows Azure services. With his profound knowledge of Microsoft technologies, he has written numerous books, and is certified as a Microsoft Certified Trainer and Professional Developer. Christian speaks at international conferences such as TechEd, Basta!, and TechDays, and he founded INETA Europe to support .NET user groups. You can contact Christian via his websites, [www.cninnovation.com](http://www.cninnovation.com) and [www.thinktecture.com](http://www.thinktecture.com), and follow his tweets at @christiannagel.

**JAY GLYNN** started writing software more than 20 years ago, writing applications for the PICK operating system using PICK basic. Since then, he has created software using Paradox PAL and Object PAL, Delphi, VBA, Visual Basic, C, Java, and of course C#. He currently works for UL PureSafety as a senior software engineer writing web-based software.

**MORGAN SKINNER** began his computing career at a young age on the Sinclair ZX80 at school, where he was underwhelmed by some code a teacher had written and so began programming in assembly language. Since then he has used a wide variety of languages and platforms, including VAX Macro Assembler, Pascal, Modula2, Smalltalk, X86 assembly language, PowerBuilder, C/C++, VB, and currently C#. He's been programming in .NET since the PDC release in 2000, and liked it so much he joined Microsoft in 2001. He's now an independent consultant.

# ABOUT THE TECHNICAL EDITORS

**DAVID FRANSON** has been a professional in the field of networking, programming, and 2D and 3D computer graphics since 1990. He is the author of *2D Artwork and 3D Modeling for Game Artists*, *The Dark Side of Game Texturing*, and *Game Character Design Complete*.

**DON REAMEY** is an architect/principal engineer for TIBCO Software working on TIBCO Spotfire business intelligence analytics software. Prior to TIBCO Don spent 12 years with Microsoft as a software development engineer working on SharePoint, SharePoint Online and InfoPath Forms Service. Don has also spent 10 years writing software in the financial service industry for capital markets.

**MITCHEL SELLERS** specializes in software development using Microsoft technologies. As the CEO of IowaComputerGurus Inc., he works with small and large companies worldwide. He is a Microsoft C# MVP, a Microsoft Certified Professional, and the author of *Professional DotNetNuke Module Programming* (Wrox Press, 2009). Mitchel frequently writes technical articles for online and print publications including SQL Server magazine, and he regularly speaks to user groups and conferences. He is also a DotNetNuke Core Team member as well as an active participant in the .NET and DotNetNuke development communities. Additional information on Mitchel's professional experience, certifications, and publications can be found at <http://mitschelsellers.com/>.

# CREDITS

**ACQUISITIONS EDITOR**

Mary James

**SENIOR PROJECT EDITOR**

Adaobi Obi Tulton

**TECHNICAL EDITORS**

David Franson

Don Reamey

Mitchel Sellers

**PRODUCTION EDITOR**

Kathleen Wisor

**COPY EDITOR**

Luann Rouff

**EDITORIAL MANAGER**

Mary Beth Wakefield

**FREELANCER EDITORIAL MANAGER**

Rosemarie Graham

**ASSOCIATE DIRECTOR OF MARKETING**

David Mayhew

**MARKETING MANAGER**

Ashley Zurcher

**BUSINESS MANAGER**

Amy Knies

**PRODUCTION MANAGER**

Tim Tate

**VICE PRESIDENT AND EXECUTIVE GROUP****PUBLISHER**

Richard Swadley

**VICE PRESIDENT AND EXECUTIVE PUBLISHER**

Neil Edde

**ASSOCIATE PUBLISHER**

Jim Minatel

**PROJECT COORDINATOR, COVER**

Katie Crocker

**PROOFREADER**

Word One, New York

**INDEXER**

Robert Swanson

**COVER DESIGNER**

Ryan Sneed

**COVER IMAGE**

© Punchstock



# ACKNOWLEDGMENTS

**I WOULD LIKE TO THANK** Adaobi Obi Tulton, Maureen Spears, and Luann Rouff for making this text more readable; Mary James; and Jim Minatel; and everyone else at Wiley who helped to get another edition of this great book published. I would also like to thank my wife and children for supporting my writing. You're my inspiration.

—CHRISTIAN NAGEL

**I WANT TO THANK** my wife and son for putting up with the time and frustrations of working on a project like this. I also want to thank all the dedicated people at Wiley for getting this book out the door.

—JAY GLYNN





# CONTENTS

## INTRODUCTION

*xlix*

## PART I: THE C# LANGUAGE

<b>CHAPTER 1: .NET ARCHITECTURE</b>	<b>3</b>
<b>The Relationship of C# to .NET</b>	<b>3</b>
<b>The Common Language Runtime</b>	<b>4</b>
Platform Independence	4
Performance Improvement	4
Language Interoperability	5
<b>A Closer Look at Intermediate Language</b>	<b>7</b>
Support for Object Orientation and Interfaces	7
Distinct Value and Reference Types	8
Strong Data Typing	8
Error Handling with Exceptions	13
Use of Attributes	13
<b>Assemblies</b>	<b>14</b>
Private Assemblies	14
Shared Assemblies	15
Reflection	15
Parallel Programming	15
Asynchronous Programming	16
<b>.NET Framework Classes</b>	<b>16</b>
<b>Namespaces</b>	<b>17</b>
<b>Creating .NET Applications Using C#</b>	<b>17</b>
Creating ASP.NET Applications	17
Windows Presentation Foundation (WPF)	19
Windows 8 Apps	20
Windows Services	20
Windows Communication Foundation	20
Windows Workflow Foundation	20
<b>The Role of C# in the .NET Enterprise Architecture</b>	<b>21</b>
<b>Summary</b>	<b>21</b>

<b>CHAPTER 2: CORE C#</b>	<b>23</b>
<b>Fundamental C#</b>	<b>24</b>
<b>Your First C# Program</b>	<b>24</b>
The Code	24
Compiling and Running the Program	24
A Closer Look	25
<b>Variables</b>	<b>27</b>
Initialization of Variables	27
Type Inference	28
Variable Scope	29
Constants	31
<b>Predefined Data Types</b>	<b>31</b>
Value Types and Reference Types	31
CTS Types	33
Predefined Value Types	33
Predefined Reference Types	35
<b>Flow Control</b>	<b>37</b>
Conditional Statements	37
Loops	40
Jump Statements	43
<b>Enumerations</b>	<b>43</b>
<b>Namespaces</b>	<b>45</b>
The using Directive	46
Namespace Aliases	47
<b>The Main() Method</b>	<b>47</b>
Multiple Main() Methods	47
Passing Arguments to Main()	48
<b>More on Compiling C# Files</b>	<b>49</b>
<b>Console I/O</b>	<b>50</b>
<b>Using Comments</b>	<b>52</b>
Internal Comments within the Source Files	52
XML Documentation	52
<b>The C# Preprocessor Directives</b>	<b>54</b>
#define and #undef	54
#if, #elif, #else, and #endif	55
#warning and #error	56
#region and #endregion	56
#line	56
#pragma	57
<b>C# Programming Guidelines</b>	<b>57</b>
Rules for Identifiers	57

Usage Conventions	58
<b>Summary</b>	<b>63</b>
<b>CHAPTER 3: OBJECTS AND TYPES</b>	<b>65</b>
<b>Creating and Using Classes</b>	<b>65</b>
<b>Classes and Structs</b>	<b>66</b>
<b>Classes</b>	<b>66</b>
Data Members	67
Function Members	67
readonly Fields	78
<b>Anonymous Types</b>	<b>79</b>
<b>Structs</b>	<b>80</b>
Structs Are Value Types	81
Structs and Inheritance	82
Constructors for Structs	82
<b>Weak References</b>	<b>82</b>
<b>Partial Classes</b>	<b>83</b>
<b>Static Classes</b>	<b>85</b>
<b>The Object Class</b>	<b>85</b>
System.Object Methods	85
The ToString() Method	86
<b>Extension Methods</b>	<b>87</b>
<b>Summary</b>	<b>88</b>
<b>CHAPTER 4: INHERITANCE</b>	<b>89</b>
<b>Inheritance</b>	<b>89</b>
<b>Types of Inheritance</b>	<b>89</b>
Implementation Versus Interface Inheritance	90
Multiple Inheritance	90
Structs and Classes	90
<b>Implementation Inheritance</b>	<b>90</b>
Virtual Methods	91
Hiding Methods	92
Calling Base Versions of Functions	93
Abstract Classes and Functions	94
Sealed Classes and Methods	94
Constructors of Derived Classes	95
<b>Modifiers</b>	<b>99</b>
Visibility Modifiers	99
Other Modifiers	100
<b>Interfaces</b>	<b>100</b>

Defining and Implementing Interfaces	101
Derived Interfaces	104
<b>Summary</b>	<b>105</b>
<b>CHAPTER 5: GENERICS</b>	<b>107</b>
<b>Generics Overview</b>	<b>107</b>
Performance	108
Type Safety	109
Binary Code Reuse	109
Code Bloat	110
Naming Guidelines	110
<b>Creating Generic Classes</b>	<b>110</b>
<b>Generics Features</b>	<b>114</b>
Default Values	114
Constraints	115
Inheritance	117
Static Members	118
<b>Generic Interfaces</b>	<b>118</b>
Covariance and Contra-variance	119
Covariance with Generic Interfaces	120
Contra-Variance with Generic Interfaces	121
<b>Generic Structs</b>	<b>122</b>
<b>Generic Methods</b>	<b>124</b>
Generic Methods Example	125
Generic Methods with Constraints	125
Generic Methods with Delegates	126
Generic Methods Specialization	127
<b>Summary</b>	<b>128</b>
<b>CHAPTER 6: ARRAYS AND TUPLES</b>	<b>129</b>
<b>Multiple Objects of the Same     and Different Types</b>	<b>129</b>
<b>Simple Arrays</b>	<b>130</b>
Array Declaration	130
Array Initialization	130
Accessing Array Elements	131
Using Reference Types	131
<b>Multidimensional Arrays</b>	<b>132</b>
<b>Jagged Arrays</b>	<b>133</b>
<b>Array Class</b>	<b>134</b>
Creating Arrays	134

Copying Arrays	136
Sorting	136
<b>Arrays as Parameters</b>	<b>139</b>
Array Covariance	139
ArraySegment<T>	139
<b>Enumerations</b>	<b>140</b>
IEnumerator Interface	141
foreach Statement	141
yield Statement	141
<b>Tuples</b>	<b>146</b>
<b>Structural Comparison</b>	<b>147</b>
<b>Summary</b>	<b>149</b>
 <b>CHAPTER 7: OPERATORS AND CASTS</b>	 <b>151</b>
<b>Operators and Casts</b>	<b>151</b>
<b>Operators</b>	<b>151</b>
Operator Shortcuts	153
Operator Precedence	157
<b>Type Safety</b>	<b>157</b>
Type Conversions	158
Boxing and Unboxing	161
<b>Comparing Objects for Equality</b>	<b>162</b>
Comparing Reference Types for Equality	162
Comparing Value Types for Equality	163
<b>Operator Overloading</b>	<b>163</b>
How Operators Work	164
Operator Overloading Example: The Vector Struct	165
Which Operators Can You Overload?	171
<b>User-Defined Casts</b>	<b>172</b>
Implementing User-Defined Casts	173
Multiple Casting	178
<b>Summary</b>	<b>181</b>
 <b>CHAPTER 8: DELEGATES, LAMBDA, AND EVENTS</b>	 <b>183</b>
<b>Referencing Methods</b>	<b>183</b>
<b>Delegates</b>	<b>184</b>
Declaring Delegates	185
Using Delegates	186
Simple Delegate Example	189
Action<T> and Func<T> Delegates	190
BubbleSorter Example	191

Multicast Delegates	193
Anonymous Methods	197
<b>Lambda Expressions</b>	<b>198</b>
Parameters	199
Multiple Code Lines	199
Closures	199
Closures with Foreach Statements	200
<b>Events</b>	<b>201</b>
Event Publisher	201
Event Listener	203
Weak Events	204
<b>Summary</b>	<b>208</b>
<b>CHAPTER 9: STRINGS AND REGULAR EXPRESSIONS</b>	<b>209</b>
<hr/>	
<b>Examining System.String</b>	<b>210</b>
Building Strings	211
StringBuilder Members	214
Format Strings	215
<b>Regular Expressions</b>	<b>221</b>
Introduction to Regular Expressions	221
The RegularExpressionsPlayaround Example	222
Displaying Results	225
Matches, Groups, and Captures	226
<b>Summary</b>	<b>228</b>
<b>CHAPTER 10: COLLECTIONS</b>	<b>229</b>
<hr/>	
<b>Overview</b>	<b>229</b>
<b>Collection Interfaces and Types</b>	<b>230</b>
<b>Lists</b>	<b>231</b>
Creating Lists	232
Read-Only Collections	241
<b>Queues</b>	<b>241</b>
<b>Stacks</b>	<b>245</b>
<b>Linked Lists</b>	<b>247</b>
<b>Sorted List</b>	<b>251</b>
<b>Dictionaries</b>	<b>253</b>
Key Type	254
Dictionary Example	255
Lookups	259
Sorted Dictionaries	260
<b>Sets</b>	<b>260</b>

---

<b>Observable Collections</b>	<b>262</b>
<b>Bit Arrays</b>	<b>263</b>
BitArray	263
BitVector32	266
<b>Concurrent Collections</b>	<b>268</b>
Creating Pipelines	269
Using BlockingCollection	272
Using ConcurrentDictionary	273
Completing the Pipeline	275
<b>Performance</b>	<b>276</b>
<b>Summary</b>	<b>278</b>
 <b>CHAPTER 11: LANGUAGE INTEGRATED QUERY</b>	 <b>279</b>
<hr/>	
<b>LINQ Overview</b>	<b>279</b>
Lists and Entities	280
LINQ Query	283
Extension Methods	284
Deferred Query Execution	285
<b>Standard Query Operators</b>	<b>287</b>
Filtering	289
Filtering with Index	289
Type Filtering	290
Compound from	290
Sorting	291
Grouping	292
Grouping with Nested Objects	293
Inner Join	294
Left Outer Join	295
Group Join	296
Set Operations	299
Zip	300
Partitioning	301
Aggregate Operators	302
Conversion Operators	303
Generation Operators	304
<b>Parallel LINQ</b>	<b>305</b>
Parallel Queries	305
Partitioners	306
Cancellation	306
<b>Expression Trees</b>	<b>307</b>
<b>LINQ Providers</b>	<b>310</b>
<b>Summary</b>	<b>310</b>

<b>CHAPTER 12: DYNAMIC LANGUAGE EXTENSIONS</b>	<b>313</b>
<b>Dynamic Language Runtime</b>	<b>313</b>
<b>The Dynamic Type</b>	<b>314</b>
Dynamic Behind the Scenes	315
<b>Hosting the DLR ScriptRuntime</b>	<b>318</b>
<b>DynamicObject and ExpandoObject</b>	<b>321</b>
DynamicObject	321
ExpandoObject	322
<b>Summary</b>	<b>324</b>
<b>CHAPTER 13: ASYNCHRONOUS PROGRAMMING</b>	<b>325</b>
<b>Why Asynchronous Programming Is Important</b>	<b>325</b>
<b>Asynchronous Patterns</b>	<b>326</b>
Synchronous Call	333
Asynchronous Pattern	334
Event-Based Asynchronous Pattern	335
Task-Based Asynchronous Pattern	336
<b>Foundation of Asynchronous Programming</b>	<b>338</b>
Creating Tasks	338
Calling an Asynchronous Method	338
Continuation with Tasks	339
Synchronization Context	339
Using Multiple Asynchronous Methods	340
Converting the Asynchronous Pattern	341
<b>Error Handling</b>	<b>341</b>
Handling Exceptions with Asynchronous Methods	342
Exceptions with Multiple Asynchronous Methods	343
Using AggregateException Information	343
<b>Cancellation</b>	<b>344</b>
Starting a Cancellation	344
Cancellation with Framework Features	345
Cancellation with Custom Tasks	345
<b>Summary</b>	<b>346</b>
<b>CHAPTER 14: MEMORY MANAGEMENT AND POINTERS</b>	<b>347</b>
<b>Memory Management</b>	<b>347</b>
<b>Memory Management Under the Hood</b>	<b>348</b>
Value Data Types	348
Reference Data Types	349
Garbage Collection	351



---

<b>Freeing Unmanaged Resources</b>	<b>353</b>
Destructors	353
The IDisposable Interface	354
Implementing IDisposable and a Destructor	356
<b>Unsafe Code</b>	<b>357</b>
Accessing Memory Directly with Pointers	357
Pointer Example: PointerPlayground	366
Using Pointers to Optimize Performance	370
<b>Summary</b>	<b>374</b>
 <b>CHAPTER 15: REFLECTION</b>	 <b>375</b>
<hr/>	
<b>Manipulating and Inspecting Code at Runtime</b>	<b>375</b>
<b>Custom Attributes</b>	<b>376</b>
Writing Custom Attributes	376
Custom Attribute Example: WhatsNewAttributes	380
<b>Using Reflection</b>	<b>382</b>
The System.Type Class	382
The TypeView Example	385
The Assembly Class	386
Completing the WhatsNewAttributes Example	388
<b>Summary</b>	<b>391</b>
 <b>CHAPTER 16: ERRORS AND EXCEPTIONS</b>	 <b>393</b>
<hr/>	
<b>Introduction</b>	<b>393</b>
<b>Exception Classes</b>	<b>394</b>
<b>Catching Exceptions</b>	<b>395</b>
Implementing Multiple Catch Blocks	398
Catching Exceptions from Other Code	401
System.Exception Properties	401
What Happens If an Exception Isn't Handled?	402
Nested try Blocks	402
<b>User-Defined Exception Classes</b>	<b>404</b>
Catching the User-Defined Exceptions	405
Throwing the User-Defined Exceptions	407
Defining the User-Defined Exception Classes	410
<b>Caller Information</b>	<b>411</b>
<b>Summary</b>	<b>413</b>

**PART II: VISUAL STUDIO**

<b>CHAPTER 17: VISUAL STUDIO 2012</b>	<b>417</b>
<b>Working with Visual Studio 2012</b>	<b>417</b>
Project File Changes	420
Visual Studio Editions	420
Visual Studio Settings	421
<b>Creating a Project</b>	<b>421</b>
Multi-Targeting the .NET Framework	422
Selecting a Project Type	423
<b>Exploring and Coding a Project</b>	<b>426</b>
Solution Explorer	426
Working with the Code Editor	432
Learning and Understanding Other Windows	433
Arranging Windows	437
<b>Building a Project</b>	<b>437</b>
Building, Compiling, and Making	437
Debugging and Release Builds	438
Selecting a Configuration	440
Editing Configurations	440
<b>Debugging Your Code</b>	<b>441</b>
Setting Breakpoints	441
Using Data Tips and Debugger Visualizers	442
Monitoring and Changing Variables	444
Exceptions	444
Multithreading	445
IntelliTrace	446
<b>Refactoring Tools</b>	<b>446</b>
<b>Architecture Tools</b>	<b>448</b>
Dependency Graph	448
Layer Diagram	449
<b>Analyzing Applications</b>	<b>450</b>
Sequence Diagram	451
Profiler	451
Concurrency Visualizer	453
Code Analysis	454
Code Metrics	455
<b>Unit Tests</b>	<b>455</b>

Creating Unit Tests	456
Running Unit Tests	456
Expecting Exceptions	458
Testing All Code Paths	458
External Dependencies	459
Fakes Framework	461
<b>Windows 8, WCF, WF, and More</b>	<b>463</b>
Building WCF Applications with Visual Studio 2012	463
Building WF Applications with Visual Studio 2012	464
Building Windows 8 Apps with Visual Studio 2012	464
<b>Summary</b>	<b>466</b>
<b>CHAPTER 18: DEPLOYMENT</b>	<b>467</b>
<b>Deployment as Part of the Application Life Cycle</b>	<b>467</b>
<b>Planning for Deployment</b>	<b>468</b>
Overview of Deployment Options	468
Deployment Requirements	469
Deploying the .NET Runtime	469
<b>Traditional Deployment</b>	<b>469</b>
xcopy Deployment	470
xcopy and Web Applications	471
Windows Installer	471
<b>ClickOnce</b>	<b>471</b>
ClickOnce Operation	472
Publishing a ClickOnce Application	472
ClickOnce Settings	474
Application Cache for ClickOnce Files	475
Application Installation	475
ClickOnce Deployment API	476
<b>Web Deployment</b>	<b>477</b>
Web Application	477
Configuration Files	477
Creating a Web Deploy Package	478
<b>Windows 8 Apps</b>	<b>479</b>
Creating an App Package	480
Windows App Certification Kit	481
Sideloaded	482
Windows Deployment API	482
<b>Summary</b>	<b>484</b>

**PART III: FOUNDATION**

<b>CHAPTER 19: ASSEMBLIES</b>	<b>487</b>
<b>What are Assemblies?</b>	<b>487</b>
Assembly Features	488
Assembly Structure	489
Assembly Manifests	489
Namespaces, Assemblies, and Components	490
Private and Shared Assemblies	490
Satellite Assemblies	490
Viewing Assemblies	491
Creating Assemblies	491
Creating Modules and Assemblies	491
Assembly Attributes	492
Creating and Loading Assemblies Dynamically	494
<b>Application Domains</b>	<b>497</b>
<b>Shared Assemblies</b>	<b>501</b>
Strong Names	501
Integrity Using Strong Names	502
Global Assembly Cache	502
Creating a Shared Assembly	503
Creating a Strong Name	503
Installing the Shared Assembly	504
Using the Shared Assembly	504
Delayed Signing of Assemblies	505
References	506
Native Image Generator	507
<b>Configuring .NET Applications</b>	<b>508</b>
Configuration Categories	509
Binding to Assemblies	510
<b>Versioning</b>	<b>511</b>
Version Numbers	511
Getting the Version Programmatically	512
Binding to Assembly Versions	512
Publisher Policy Files	513
Runtime Version	514
<b>Sharing Assemblies Between Different Technologies</b>	<b>515</b>
Sharing Source Code	515
Portable Class Library	516
<b>Summary</b>	<b>517</b>

---

<b>CHAPTER 20: DIAGNOSTICS</b>	<b>519</b>
<b>Diagnostics Overview</b>	<b>519</b>
<b>Code Contracts</b>	<b>520</b>
Preconditions	521
Postconditions	522
Invariants	523
Purity	524
Contracts for Interfaces	524
Abbreviations	525
Contracts and Legacy Code	526
<b>Tracing</b>	<b>526</b>
Trace Sources	527
Trace Switches	528
Trace Listeners	529
Filters	531
Correlation	532
Tracing with ETW	535
<b>Event Logging</b>	<b>536</b>
Event-Logging Architecture	537
Event-Logging Classes	538
Creating an Event Source	539
Writing Event Logs	540
Resource Files	540
<b>Performance Monitoring</b>	<b>544</b>
Performance-Monitoring Classes	544
Performance Counter Builder	544
Adding PerformanceCounter Components	547
perfmon.exe	549
<b>Summary</b>	<b>550</b>
<b>CHAPTER 21: TASKS, THREADS, AND SYNCHRONIZATION</b>	<b>551</b>
<b>Overview</b>	<b>552</b>
<b>Parallel Class</b>	<b>553</b>
Looping with the Parallel.For Method	553
Looping with the Parallel.ForEach Method	556
Invoking Multiple Methods with the Parallel.Invoke Method	557
<b>Tasks</b>	<b>557</b>
Starting Tasks	557
Futures—Results from Tasks	560

Continuation Tasks	561
Task Hierarchies	561
<b>Cancellation Framework</b>	<b>562</b>
Cancellation of Parallel.For	562
Cancellation of Tasks	564
<b>Thread Pools</b>	<b>565</b>
<b>The Thread Class</b>	<b>566</b>
Passing Data to Threads	567
Background Threads	568
Thread Priority	569
Controlling Threads	570
<b>Threading Issues</b>	<b>570</b>
Race Conditions	570
Deadlocks	573
<b>Synchronization</b>	<b>575</b>
The lock Statement and Thread Safety	575
Interlocked	580
Monitor	581
SpinLock	582
WaitHandle	582
Mutex	583
Semaphore	584
Events	586
Barrier	589
ReaderWriterLockSlim	590
<b>Timers</b>	<b>593</b>
<b>Data Flow</b>	<b>594</b>
Using an Action Block	594
Source and Target Blocks	595
Connecting Blocks	596
<b>Summary</b>	<b>598</b>
<b>CHAPTER 22: SECURITY</b>	<b>601</b>
<b>Introduction</b>	<b>601</b>
<b>Authentication and Authorization</b>	<b>602</b>
Identity and Principal	602
Roles	603
Declarative Role-Based Security	604
Claims	605
Client Application Services	606