

Contents

Acknowledgments	xi
Introduction	xxi
Part I: Getting Started	1
Chapter 1: Getting Started	3
<hr/>	
A Short History of Object-Oriented Programming (OOP)	4
Installing C#	5
Downloading C# Express	5
Installing C# Express	6
A Test Program Using C# Express	8
Creating a Project	8
The C# Integrated Development Environment	10
Adding an Object to a Windows Form	12
Running the Program	16
Summary	17
Chapter 2: Understanding Objects	19
<hr/>	
Understanding Objects	19
Everyday Use of Objects	20
How Many Properties, How Many Methods?	22
What Do I Do After I've Defined a Class?	23
Developing a Program Plan	30
Where to Start?	30
Creating a Simple Application Using Objects	32
Using the Program Steps to Create a Program Plan	33
Using C# to Implement Our Program Plan	35
Adding Program References to Your Project	37
Adding a New Item to the Project	38
Setting the Project Properties	40
Adding Objects to the frmMain Object	41
Adding Program Code for the Process Step	44

Contents

Constructing the User Interface Object	45
The btnDisplayOutput Click Event Code	46
Critique of the btnDisplayOutput Click Event Code	48
Summary	49
Part II: Understanding C# Syntax	51
Chapter 3: Understanding Data Types	53
<hr/>	
Integer Data Types	54
Range of Integer Data Types	54
Which Integer Should You Use?	56
Sequence of Steps to Create a New Project	57
Designing the User Interface	58
Code for the Process Step	58
Variable Naming Rules and Conventions	59
Using the TryParse() Method	60
Floating-Point Data Types	63
Floating-Point Precision	65
Which Floating-Point Data Type Should You Use?	67
Monetary Values: The Decimal Data Type	67
Using Intellisense to Locate Program Errors	68
The Boolean Data Type	70
Summary	72
Exercises	73
Chapter 4: Understanding C# Statements	75
<hr/>	
Basic Building Blocks of a Programming Language	76
Operands and Operators	76
Expressions	77
Statements	77
Operator Precedence	78
Overriding the Default Precedence Order	79
Defining Variables	80
Defining a Variable from the Compiler's Point of View	80
Using a Variable in a Program	83
The Bucket Analogy	85
Types of Program Errors	86
Syntax Errors	86
Semantic Errors	87
Logic Errors	87

The Visual Studio Debugger	87
The Debugging Process	87
Making Repetitious Debugging Easier	88
Using the Debugger	88
Defensive Coding	93
Use Program Comments	93
Use Meaningful Variable Names	95
Avoid Magic Numbers	95
Summary	96
Exercises	97
Chapter 5: Understanding Reference Data Types	99
<hr/>	
String Variables	100
Defining a String Reference Variable	100
Why Reference Types are Different From Value Types	103
Reference Variable Rules	104
Reference Type Variables Versus Value Type Variables	105
Why Do Reference Variables Work the Way They Do?	105
Pass by Value versus Pass by Reference	106
A Little Efficiency Gain	106
Using String Variables	107
String Manipulation	108
Strings and Escape Sequences	121
Verbatim String Literals	122
DateTime Reference Objects	123
DateTime and ToString() Conversions	123
Summary	134
Exercises	134
Chapter 6: Making Decisions in Code	135
<hr/>	
Relational Operators	136
Using Relational Operators — The if Statement	136
The if-else Statement	140
Shorthand for Simple if-else: The Ternary Operator	141
Style Considerations for if and if-else Statements	142
Nested if Statements	144
RDC	145
Cascading if Statements	146
Logical Operators	147
The switch Statement	150
Summary	152
Exercises	152

Contents

Chapter 7: Statement Repetition Using Loops **153**

Program Loops	153
Good Loops, Bad Loops	154
The for Loop	155
Nested for Loops	162
while Loops	165
Why Have More Than One Type of Loop?	166
do-while Program Loops	167
The continue Statement	171
Summary	172
Exercises	173

Chapter 8: Arrays **175**

What is an Array?	175
Some Array Details	177
The Listview Object	183
Arrays are Objects	187
Multidimensional Arrays	189
Initializing Arrays	193
Variations for Initializing an Array	193
Initializing Multidimensional Arrays	195
Initializer Lists for Objects	195
Collections	198
ArrayList Objects	200
Summary	203
Exercises	203

Part III: Writing Your Own Classes **205**

Chapter 9: Designing Classes **207**

Class Design	208
Adding a Class to a Project	208
Scope	209
Visualizing Scope	213
Why Does C# Support Scope?	214
Designing a Program	215
The Five Program Steps	215
Look at the Forest, Not Just the Trees	218

UML Light	219
Access Specifiers	219
UML Methods	222
Think Like a User	224
The clsDates Design	225
namespace Modifier	227
Class Organization	227
User Interfaces Versus User Interfaces	231
Summary	232
Exercises	232
Chapter 10: Designing and Writing Custom Classes	233
<hr/>	
Constructors	233
Default Constructors	234
Non-Default Constructors	234
Constructor Overloading	235
Property Methods	237
Getters and Setters	238
What to Do if an Error Occurs in a Property Method	242
Class Design for Deck-of-Cards Program	244
UML Class Diagram	245
Designing a Card Game Using clsCardDeck	256
Summary	269
Exercises	270
Chapter 11: Exception Handling and Debugging	271
<hr/>	
Overview	271
Bugs	272
Program Errors	273
Data Validation	273
Limit User Input	274
Exception Handling	282
try-catch Statement Blocks	283
Program Debugging	288
The Nature of the Beast	289
The Visual Studio Debugger	291
Single-Stepping Through the Program	294
Scaffold Code	296
Defensive Coding	297
Summary	297
Exercises	298

Chapter 12: Generics	299
What are Generics?	299
Recursion	305
Data Problems	305
What are Generics?	306
Generics Versus ArrayLists	306
Using a Generic Class	314
Generic Quicksort	315
Using Generics with Constraints and Interfaces	319
Interfaces	319
Why Use an Interface?	320
Using an Interface	322
Summary	326
Exercises	326
Part IV: Storing Data	327
Chapter 13: Using Disk Data Files	329
Directories	329
The DriveInfo Class	330
Directory Class	330
DirectoryInfo Class	331
File Namespace	337
FileInfo Class	338
Types of Files	339
Textual Versus Binary Data Files	339
Reading the Data	349
Sequential Versus Random Access Files	350
Sequential Files	350
Random Access Files	351
frmMain	356
Navigating the Records	363
Delete a Record	364
clsRandomAccess	364
Serialization and Deserialization	382
To Serialize or Not to Serialize	388
MDI, Menus, and File Dialogs	388
Adding a Menu	389
Adding a File Open Dialog	390
Calling Another Form	392

Summary	394
Exercises	394
Chapter 14: Using Databases	395
What is a Database?	395
The Structure of a Database	396
Designing Fields for a Database Table	397
Using Your Own Databases	400
ADO.NET	401
Using SQL	402
The SELECT Statement	402
The WHERE Predicate	403
The ORDER BY Clause	403
Aggregates	404
Creating a New Database	406
Creating a New Table (CREATE TABLE)	408
Adding Records to a Table (INSERT INTO)	413
Edit a Record (UPDATE)	416
Delete a Record (DELETE)	421
Database Queries	421
Binding a DataGridView Object to a Database	423
Using a DataGridView Object Without Data Binding	426
Using LINQ	432
LINQ Query Keywords	432
Namespaces and References for LINQ	435
Summary	441
Exercises	442
Chapter 15: Inheritance and Polymorphism	443
What is Inheritance?	443
An Inheritance Example	444
Base Classes are Not Derived Classes	459
Abstract Classes	459
Polymorphism	460
Sending the Snow-Removal Message	462
Summary	463
Exercises	463
Appendix A: Exercise Solutions	465
Index	495
