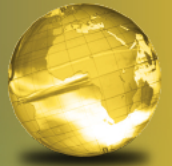


GLOBAL
EDITION



Visual C#

How to Program

SIXTH EDITION

Paul Deitel • Harvey Deitel



DIGITAL RESOURCES FOR STUDENTS

Your new textbook provides 12-month access to digital resources that may include source code, web chapters and more. Refer to the preface in the textbook for a detailed list of resources.

Follow the instructions below to register for the Companion Website for Paul Deitel and Harvey Deitel's ***Visual C# How to Program, Sixth Edition, Global Edition***.

- 1 Go to **www.pearsonglobaleditions.com**
- 2 Enter the title of your textbook or browse by author name.
- 3 Click Companion Website.
- 4 Click Register and follow the on-screen instructions to create a login name and password.

ISSDCH-NEUSS-WAXEN-PISTE-GNASH-WWRSE



Use this student access code to register for the Companion Website.

Use the login name and password you created during registration to start using the digital resources that accompany your textbook.

IMPORTANT

This access code can only be used once. This subscription is valid for 12 months upon activation and is not transferable.

For technical support go to **<https://support.pearson.com/getsupport>**

Visual C# How to Program, Global Edition

Table of Contents

Cover

Title Page

Copyright Page

Contents

Preface

Before You Begin

1 Introduction to Computers, the Internet and Visual C# 41

1.1 Introduction 42

1.2 Computers and the Internet in Industry and Research 42

1.3 Hardware and Software 45

1.3.1 Moores Law 45

1.3.2 Computer Organization 46

1.4 Data Hierarchy 47

1.5 Machine Languages, Assembly Languages and High-Level
Languages 50

1.6 Object Technology 51

1.7 Internet and World Wide Web 53

1.8 C# 55

1.8.1 Object-Oriented Programming 56

1.8.2 Event-Driven Programming 56

1.8.3 Visual Programming 56

1.8.4 Generic and Functional Programming 56

1.8.5 An International Standard 57

Table of Contents

1.8.6 C# on Non-Windows Platforms	57
1.8.7 Internet and Web Programming	57
1.8.8 Asynchronous Programming with async and await	57
1.8.9 Other Key Programming Languages	58
1.9 Microsofts .NET	60
1.9.1 .NET Framework	60
1.9.2 Common Language Runtime	60
1.9.3 Platform Independence	61
1.9.4 Language Interoperability	61
1.10 Microsofts Windows® Operating System	61
1.11 Visual Studio Integrated Development Environment	63
1.12 Painter Test-Drive in Visual Studio Community	63
2 Introduction to Visual Studio and Visual Programming	73
2.1 Introduction	74
2.2 Overview of the Visual Studio Community 2015 IDE	74
2.2.1 Introduction to Visual Studio Community 2015	74
2.2.2 Visual Studio Themes	75
2.2.3 Links on the Start Page	75
2.2.4 Creating a New Project	76
2.2.5 New Project Dialog and Project Templates	77
2.2.6 Forms and Controls	78
2.3 Menu Bar and Toolbar	79
2.4 Navigating the Visual Studio IDE	82
2.4.1 Solution Explorer	83
2.4.2 Toolbox	84
2.4.3 Properties Window	84
2.5 Help Menu and Context-Sensitive Help	86
2.6 Visual Programming: Creating a Simple App that Displays	

Table of Contents

Text and an Image 87

2.7 Wrap-Up 96

2.8 Web Resources 97

3 Introduction to C# App Programming 105

3.1 Introduction 106

3.2 Simple App: Displaying a Line of Text 106

3.2.1 Comments 107

3.2.2 using Directive 108

3.2.3 Blank Lines and Whitespace 109

3.2.4 Class Declaration 109

3.2.5 Main Method 111

3.2.6 Displaying a Line of Text 111

3.2.7 Matching Left ({} and Right {}) Braces 112

3.3 Creating a Simple App in Visual Studio 112

3.3.1 Creating the Console App 112

3.3.2 Changing the Name of the App File 114

3.3.3 Writing Code and Using IntelliSense 114

3.3.4 Compiling and Running the App 116

3.3.5 Errors, Error Messages and the Error List Window 117

3.4 Modifying Your Simple C# App 117

3.4.1 Displaying a Single Line of Text with Multiple Statements 118

3.4.2 Displaying Multiple Lines of Text with a Single Statement 118

3.5 String Interpolation 120

3.6 Another C# App: Adding Integers 121

3.6.1 Declaring the int Variable number1 122

3.6.2 Declaring Variables number2 and sum 123

3.6.3 Prompting the User for Input 123

3.6.4 Reading a Value into Variable number1 123

3.6.5 Prompting the User for Input and Reading a Value into number2 124

Table of Contents

3.6.6 Summing number1 and number2	124
3.6.7 Displaying the sum with string Interpolation	125
3.6.8 Performing Calculations in Output Statements	125
3.7 Memory Concepts	125
3.8 Arithmetic	126
3.8.1 Arithmetic Expressions in Straight-Line Form	127
3.8.2 Parentheses for Grouping Subexpressions	127
3.8.3 Rules of Operator Precedence	127
3.8.4 Sample Algebraic and C# Expressions	128
3.8.5 Redundant Parentheses	129
3.9 Decision Making: Equality and Relational Operators	129
3.10 Wrap-Up	134
4 Introduction to Classes, Objects, Methods and strings	146
4.1 Introduction	147
4.2 Test-Driving an Account Class	148
4.2.1 Instantiating an ObjectKeyword new	148
4.2.2 Calling Class Accounts GetName Method	149
4.2.3 Inputting a Name from the User	149
4.2.4 Calling Class Accounts SetName Method	150
4.3 Account Class with an Instance Variable and Set and Get Methods	150
4.3.1 Account Class Declaration	150
4.3.2 Keyword class and the Class Body	151
4.3.3 Instance Variable name of Type string	151
4.3.4 SetName Method	152
4.3.5 GetName Method	154
4.3.7 Account UML Class Diagram	155
4.4 Creating, Compiling and Running a Visual C# Project with	

Table of Contents

Two Classes 156

4.5 Software Engineering with Set and Get Methods 157

4.6 Account Class with a Property Rather Than Set and Get
Methods 158

4.6.1 Class AccountTest Using Accounts Name Property 158

4.6.2 Account Class with an Instance Variable and a Property 160

4.6.3 Account UML Class Diagram with a Property 162

4.7 Auto-Implemented Properties 162

4.8 Account Class: Initializing Objects with Constructors 163

4.8.1 Declaring an Account Constructor for Custom Object Initialization 163

4.8.2 Class AccountTest: Initializing Account Objects When Theyre
Created 164

4.9 Account Class with a Balance; Processing Monetary Amounts
166

4.9.1 Account Class with a decimal balance Instance Variable 166

4.9.2 AccountTest Class That Uses Account Objects with Balances 169

4.10 Wrap-Up 173

5 Algorithm Development and Control Statements: Part 1
182

5.1 Introduction 183

5.2 Algorithms 184

5.3 Pseudocode 184

5.4 Control Structures 185

5.4.1 Sequence Structure 185

5.4.2 Selection Statements 186

5.4.3 Iteration Statements 187

5.4.4 Summary of Control Statements 187

5.5 if Single-Selection Statement 187

Table of Contents

5.6	ifelse Double-Selection Statement	188
5.6.1	Nested ifelse Statements	189
5.6.2	Dangling-else Problem	191
5.6.3	Blocks	191
5.6.4	Conditional Operator (?:)	192
5.7	Student Class: Nested ifelse Statements	193
5.8	while Iteration Statement	196
5.9	Formulating Algorithms: Counter-Controlled Iteration	197
5.9.1	Pseudocode Algorithm with Counter-Controlled Iteration	197
5.9.2	Implementing Counter-Controlled Iteration	198
5.9.3	Integer Division and Truncation	200
5.10	Formulating Algorithms: Sentinel-Controlled Iteration	201
5.10.1	Top-Down, Stepwise Refinement: The Top and First Refinement	201
5.10.2	Second Refinement	202
5.10.3	Implementing Sentinel-Controlled Iteration	204
5.10.4	Program Logic for Sentinel-Controlled Iteration	205
5.10.5	Braces in a while Statement	206
5.10.6	Converting Between Simple Types Explicitly and Implicitly	206
5.10.7	Formatting Floating-Point Numbers	207
5.11	Formulating Algorithms: Nested Control Statements	208
5.11.1	Problem Statement	208
5.11.2	Top-Down, Stepwise Refinement: Pseudocode Representation of the Top	209
5.11.3	Top-Down, Stepwise Refinement: First Refinement	209
5.11.4	Top-Down, Stepwise Refinement: Second Refinement	209
5.11.5	Complete Second Refinement of the Pseudocode	210
5.11.6	App That Implements the Pseudocode Algorithm	211
5.12	Compound Assignment Operators	213
5.13	Increment and Decrement Operators	213

Table of Contents

5.13.1 Prefix Increment vs. Postfix Increment 214

5.13.2 Simplifying Increment Statements 215

5.13.3 Operator Precedence and Associativity 216

5.14 Simple Types 216

5.15 Wrap-Up 217

6 Control Statements: Part 2 232

6.1 Introduction 233

6.2 Essentials of Counter-Controlled Iteration 233

6.3 for Iteration Statement 235

6.3.1 A Closer Look at the for Statements Header 236

6.3.2 General Format of a for Statement 236

6.3.3 Scope of a for Statements Control Variable 236

6.3.4 Expressions in a for Statements Header Are Optional 237

6.3.5 Placing Arithmetic Expressions in a for Statements Header 237

6.3.6 Using a for Statements Control Variable in the Statements Body 238

6.3.7 UML Activity Diagram for the for Statement 238

6.4 Examples Using the for Statement 238

6.5 App: Summing Even Integers 239

6.6 App: Compound-Interest Calculations 240

6.6.1 Performing the Interest Calculations with Math Method pow 241

6.6.2 Formatting with Field Widths and Alignment 242

6.6.3 Caution: Do Not Use float or double for Monetary Amounts 243

6.7 dowhile Iteration Statement 244

6.8 switch Multiple-Selection Statement 245

6.8.1 Using a switch Statement to Count A, B, C, D and F Grades. 245

6.8.2 switch Statement UML Activity Diagram 249

6.8.3 Notes on the Expression in Each case of a switch 250

6.9 Class AutoPolicy Case Study: strings in switch Statements 251

Table of Contents

6.10 break and continue Statements 253

6.10.1 break Statement 253

6.10.2 continue Statement 254

6.11 Logical Operators 255

6.11.1 Conditional AND (&&) Operator 256

6.11.2 Conditional OR (||) Operator 256

6.11.3 Short-Circuit Evaluation of Complex Conditions 257

6.11.4 Boolean Logical AND (&) and Boolean Logical OR (||) Operators 257

6.11.5 Boolean Logical Exclusive OR (^) 258

6.11.6 Logical Negation (!) Operator 258

6.11.7 Logical Operators Example 259

6.12 Structured-Programming Summary 261

6.13 Wrap-Up 266

7 Methods: A Deeper Look 277

7.1 Introduction 278

7.2 Packaging Code in C# 279

7.2.1 Modularizing Programs 279

7.2.2 Calling Methods 280

7.3 static Methods, static Variables and Class Math 280

7.3.1 Math Class Methods 281

7.3.2 Math Class Constants PI and E 282

7.3.3 Why Is Main Declared static? 282

7.3.4 Additional Comments About Main 283

7.4 Methods with Multiple Parameters 283

7.4.1 Keyword static 285

7.4.2 Method Maximum 285

7.4.3 Assembling strings with Concatenation 285

7.4.4 Breaking Apart Large string Literals 286

7.4.5 When to Declare Variables as Fields 287

Table of Contents

7.4.6 Implementing Method Maximum by Reusing Method Math.Max	287
7.5 Notes on Using Methods	287
7.6 Argument Promotion and Casting	288
7.6.1 Promotion Rules	289
7.6.2 Sometimes Explicit Casts Are Required	289
7.7 The .NET Framework Class Library	290
7.8 Case Study: Random-Number Generation	292
7.8.1 Creating an Object of Type Random	292
7.8.2 Generating a Random Integer	292
7.8.3 Scaling the Random-Number Range	293
7.8.4 Shifting Random-Number Range	293
7.8.5 Combining Shifting and Scaling	293
7.8.6 Rolling a Six-Sided Die	293
7.8.7 Scaling and Shifting Random Numbers	296
7.8.8 Repeatability for Testing and Debugging	297
7.9 Case Study: A Game of Chance; Introducing Enumerations	297
7.9.1 Method RollDice	300
7.9.2 Method Mains Local Variables	300
7.9.3 enum Type Status	301
7.9.4 The First Roll	301
7.9.5 enum Type DiceNames	301
7.9.6 Underlying Type of an enum	301
7.9.7 Comparing Integers and enum Constants	302
7.10 Scope of Declarations	302
7.11 Method-Call Stack and Activation Records	305
7.11.1 Method-Call Stack	305
7.11.2 Stack Frames	305
7.11.3 Local Variables and Stack Frames	306
7.11.4 Stack Overflow	306

Table of Contents

7.11.5 Method-Call Stack in Action 306

7.12 Method Overloading 309

7.12.1 Declaring Overloaded Methods 309

7.12.2 Distinguishing Between Overloaded Methods 310

7.12.3 Return Types of Overloaded Methods 311

7.13 Optional Parameters 311

7.14 Named Parameters 312

7.15 C# 6 Expression-Bodied Methods and Properties 313

7.16 Recursion 314

7.16.1 Base Cases and Recursive Calls 314

7.16.2 Recursive Factorial Calculations 314

7.16.3 Implementing Factorial Recursively 315

7.17 Value Types vs. Reference Types 317

7.18 Passing Arguments By Value and By Reference 318

7.18.1 ref and out Parameters 319

7.18.2 Demonstrating ref, out and Value Parameters 320

7.19 Wrap-Up 322

8 Arrays; Introduction to Exception Handling 339

8.1 Introduction 340

8.2 Arrays 341

8.3 Declaring and Creating Arrays 342

8.4 Examples Using Arrays 343

8.4.1 Creating and Initializing an Array 343

8.4.2 Using an Array Initializer 344

8.4.3 Calculating a Value to Store in Each Array Element 345

8.4.4 Summing the Elements of an Array 347

8.4.5 Iterating Through Arrays with foreach 347

8.4.6 Using Bar Charts to Display Array Data Graphically; Introducing Type Inference

Table of Contents

with var 349

8.4.7 Using the Elements of an Array as Counters 352

8.5 Using Arrays to Analyze Survey Results; Intro to Exception Handling
353

8.5.1 Summarizing the Results 354

8.5.2 Exception Handling: Processing the Incorrect Response 355

8.5.3 The try Statement 355

8.5.4 Executing the catch Block 355

8.5.5 Message Property of the Exception Parameter 356

8.6 Case Study: Card Shuffling and Dealing Simulation 356

8.6.1 Class Card and Getter-Only Auto-Implemented Properties 356

8.6.2 Class DeckOfCards 357

8.6.3 Shuffling and Dealing Cards 360

8.7 Passing Arrays and Array Elements to Methods 361

8.8 Case Study: GradeBook Using an Array to Store Grades 363

8.9 Multidimensional Arrays 369

8.9.1 Rectangular Arrays 369

8.9.2 Jagged Arrays 370

8.9.3 Two-Dimensional Array Example: Displaying Element Values 371

8.10 Case Study: GradeBook Using a Rectangular Array 374

8.11 Variable-Length Argument Lists 380

8.12 Using Command-Line Arguments 382

8.13 (Optional) Passing Arrays by Value and by Reference 384

8.14 Wrap-Up 388

9 Introduction to LINQ and the List Collection 410

9.1 Introduction 411

9.2 Querying an Array of int Values Using LINQ 412

9.2.1 The from Clause 414

Table of Contents

9.2.2 The where Clause	415
9.2.3 The select Clause	415
9.2.4 Iterating Through the Results of the LINQ Query	415
9.2.5 The orderby Clause	415
9.2.6 Interface IEnumerable<T>	416
9.3 Querying an Array of Employee Objects Using LINQ	416
9.3.1 Accessing the Properties of a LINQ Query's Range Variable	420
9.3.2 Sorting a LINQ Query's Results by Multiple Properties	420
9.3.3 Any, First and Count Extension Methods	420
9.3.4 Selecting a Property of an Object	420
9.3.5 Creating New Types in the select Clause of a LINQ Query	420
9.4 Introduction to Collections	421
9.4.1 List<T> Collection	421
9.4.2 Dynamically Resizing a List<T> Collection	422
9.5 Querying the Generic List Collection Using LINQ	426
9.5.1 The let Clause	428
9.5.2 Deferred Execution	428
9.5.3 Extension Methods ToArray and ToList	428
9.5.4 Collection Initializers	428
9.6 Wrap-Up	429
9.7 Deitel LINQ Resource Center	429
10 Classes and Objects: A Deeper Look	434
10.1 Introduction	435
10.2 Time Class Case Study; Throwing Exceptions	435
10.2.1 Time1 Class Declaration	436
10.2.2 Using Class Time1	437
10.3 Controlling Access to Members	439
10.4 Referring to the Current Object's Members with the this Reference	

Table of Contents

440

10.5 Time Class Case Study: Overloaded Constructors 442

10.5.1 Class Time2 with Overloaded Constructors 442

10.5.2 Using Class Time2s Overloaded Constructors 446

10.6 Default and Parameterless Constructors 448

10.7 Composition 449

10.7.1 Class Date 449

10.7.2 Class Employee 451

10.7.3 Class EmployeeTest 452

10.8 Garbage Collection and Destructors 453

10.9 static Class Members 453

10.10 readonly Instance Variables 457

10.11 Class View and Object Browser 458

10.11.1 Using the Class View Window 458

10.11.2 Using the Object Browser 459

10.12 Object Initializers 460

10.13 Operator Overloading; Introducing struct 460

10.13.1 Creating Value Types with struct 461

10.13.2 Value Type ComplexNumber 461

10.13.3 Class ComplexTest 463

10.14 Time Class Case Study: Extension Methods 464

10.15 Wrap-Up 467

11 Object-Oriented Programming: Inheritance 475

11.1 Introduction 476

11.2 Base Classes and Derived Classes 477

11.3 protected Members 479

11.4 Relationship between Base Classes and Derived Classes 480

11.4.1 Creating and Using a CommissionEmployee Class 481

Table of Contents

11.4.2	Creating a BasePlusCommissionEmployee Class without Using Inheritance	485
11.4.3	Creating a CommissionEmployeeBasePlusCommissionEmployee Inheritance Hierarchy	490
11.4.4	CommissionEmployeeBasePlusCommissionEmployee Inheritance Hierarchy Using protected Instance Variables	493
11.4.5	CommissionEmployeeBasePlusCommissionEmployee Inheritance Hierarchy Using private Instance Variables	496
11.5	Constructors in Derived Classes	500
11.6	Software Engineering with Inheritance	500
11.7	Class object	501
11.8	Wrap-Up	502
12	OOP: Polymorphism and Interfaces	508
12.1	Introduction	509
12.2	Polymorphism Examples	511
12.3	Demonstrating Polymorphic Behavior	512
12.4	Abstract Classes and Methods	515
12.5	Case Study: Payroll System Using Polymorphism	517
12.5.1	Creating Abstract Base Class Employee	518
12.5.2	Creating Concrete Derived Class SalariedEmployee	520
12.5.3	Creating Concrete Derived Class HourlyEmployee	522
12.5.4	Creating Concrete Derived Class CommissionEmployee	523
12.5.5	Creating Indirect Concrete Derived Class BasePlusCommissionEmployee	525
12.5.6	Polymorphic Processing, Operator is and Downcasting	526
12.5.7	Summary of the Allowed Assignments Between Base-Class and Derived-Class Variables	531
12.6	sealed Methods and Classes	532
12.7	Case Study: Creating and Using Interfaces	533

Table of Contents

12.7.1	Developing an IPayable Hierarchy	534
12.7.2	Declaring Interface IPayable	536
12.7.3	Creating Class Invoice	536
12.7.4	Modifying Class Employee to Implement Interface IPayable	538
12.7.5	Using Interface IPayable to Process Invoices and Employees Polymorphically	539
12.7.6	Common Interfaces of the .NET Framework Class Library	541
12.8	Wrap-Up	542
13	Exception Handling: A Deeper Look	547
13.1	Introduction	548
13.2	Example: Divide by Zero without Exception Handling	549
13.2.1	Dividing By Zero	550
13.2.2	Enter a Non-Numeric Denominator	551
13.2.3	Unhandled Exceptions Terminate the App	552
13.3	Example: Handling DivideByZeroExceptions and FormatExceptions	552
13.3.1	Enclosing Code in a try Block	554
13.3.2	Catching Exceptions	554
13.3.3	Uncaught Exceptions	555
13.3.4	Termination Model of Exception Handling	556
13.3.5	Flow of Control When Exceptions Occur	556
13.4	.NET Exception Hierarchy	557
13.4.1	Class SystemException	557
13.4.2	Which Exceptions Might a Method Throw?	558
13.5	finally Block	559
13.5.1	Moving Resource-Release Code to a finally Block	559
13.5.2	Demonstrating the finally Block	560
13.5.3	Throwing Exceptions Using the throw Statement	564
13.5.4	Rethrowing Exceptions	564

Table of Contents

13.5.5	Returning After a finally Block	565
13.6	The using Statement	566
13.7	Exception Properties	567
13.7.1	Property InnerException	567
13.7.2	Other Exception Properties	568
13.7.3	Demonstrating Exception Properties and Stack Unwinding	568
13.7.4	Throwing an Exception with an InnerException	570
13.7.5	Displaying Information About the Exception	571
13.8	User-Defined Exception Classes	571
13.9	Checking for null References; Introducing C# 6's ?. Operator	575
13.9.1	Null-Conditional Operator (?.)	575
13.9.2	Revisiting Operators is and as	576
13.9.3	Nullable Types	576
13.9.4	Null Coalescing Operator (??)	577
13.10	Exception Filters and the C# 6 when Clause	577
13.11	Wrap-Up	578
14	Graphical User Interfaces with Windows Forms: Part 1	584
14.1	Introduction	585
14.2	Windows Forms	586
14.3	Event Handling	588
14.3.1	A Simple Event-Driven GUI	589
14.3.2	Auto-Generated GUI Code	591
14.3.3	Delegates and the Event-Handling Mechanism	593
14.3.4	Another Way to Create Event Handlers	594
14.3.5	Locating Event Information	595
14.4	Control Properties and Layout	596
14.4.1	Anchoring and Docking	597

Table of Contents

14.4.2 Using Visual Studio To Edit a GUIs Layout	599
14.5 Labels, TextBoxes and Buttons	600
14.6 GroupBoxes and Panels	603
14.7 CheckBoxes and RadioButtons	606
14.7.1 CheckBoxes	606
14.7.2 Combining Font Styles with Bitwise Operators	608
14.7.3 RadioButtons	609
14.8 PictureBoxes	614
14.9 ToolTips	617
14.10 NumericUpDown Control	618
14.11 Mouse-Event Handling	621
14.12 Keyboard-Event Handling	623
14.13 Wrap-Up	627
15 Graphical User Interfaces with Windows Forms: Part 2	637
15.1 Introduction	638
15.2 Menus	638
15.3 MonthCalendar Control	648
15.4 DateTimePicker Control	649
15.5 LinkLabel Control	652
15.6 ListBox Control	655
15.7 CheckedListBox Control	660
15.8 ComboBox Control	663
15.9 TreeView Control	667
15.10 ListView Control	673
15.11 TabControl Control	679
15.12 Multiple Document Interface (MDI) Windows	683

Table of Contents

15.13 Visual Inheritance	691
15.14 User-Defined Controls	696
15.15 Wrap-Up	699
16 Strings and Characters: A Deeper Look	707
16.1 Introduction	708
16.2 Fundamentals of Characters and Strings	709
16.3 string Constructors	710
16.4 string Indexer, Length Property and CopyTo Method	711
16.5 Comparing strings	712
16.6 Locating Characters and Substrings in strings	716
16.7 Extracting Substrings from strings	719
16.8 Concatenating strings	720
16.9 Miscellaneous string Methods	720
16.10 Class StringBuilder	722
16.11 Length and Capacity Properties, EnsureCapacity Method and Indexer of Class StringBuilder	723
16.12 Append and AppendFormat Methods of Class StringBuilder	725
16.13 Insert, Remove and Replace Methods of Class StringBuilder	727
16.14 Char Methods	730
16.15 Introduction to Regular Expressions (Online)	732
16.16 Wrap-Up	732
17 Files and Streams	739
17.1 Introduction	740
17.2 Files and Streams	740
17.3 Creating a Sequential-Access Text File	741

Table of Contents

17.4	Reading Data from a Sequential-Access Text File	750
17.5	Case Study: Credit-Inquiry Program	754
17.6	Serialization	759
17.7	Creating a Sequential-Access File Using Object Serialization	760
17.8	Reading and Deserializing Data from a Binary File	764
17.9	Classes File and Directory	767
17.9.1	Demonstrating Classes File and Directory	768
17.9.2	Searching Directories with LINQ	771
17.10	Wrap-Up	775
18	Searching and Sorting	782
18.1	Introduction	783
18.2	Searching Algorithms	784
18.2.1	Linear Search	784
18.2.2	Binary Search	788
18.3	Sorting Algorithms	792
18.3.1	Selection Sort	793
18.3.2	Insertion Sort	796
18.3.3	Merge Sort	800
18.4	Summary of the Efficiency of Searching and Sorting Algorithms	806
18.5	Wrap-Up	807
19	Custom Linked Data Structures	812
19.1	Introduction	813
19.2	Simple-Type structs, Boxing and Unboxing	813
19.3	Self-Referential Classes	814
19.4	Linked Lists	815
19.5	Stacks	828
19.6	Queues	832

Table of Contents

19.7 Trees 835

19.7.1 Binary Search Tree of Integer Values 836

19.7.2 Binary Search Tree of IComparable Objects 843

19.8 Wrap-Up 849

20 Generics 855

20.1 Introduction 856

20.2 Motivation for Generic Methods 857

20.3 Generic-Method Implementation 859

20.4 Type Constraints 862

20.4.1 IComparable<T> Interface 862

20.4.2 Specifying Type Constraints 862

20.5 Overloading Generic Methods 865

20.6 Generic Classes 865

20.7 Wrap-Up 875

21 Generic Collections; Functional Programming with LINQ/PLINQ 881

21.1 Introduction 882

21.2 Collections Overview 884

21.3 Class Array and Enumerators 886

21.3.1 C# 6 using static Directive 888

21.3.2 Class UsingArrays static Fields 889

21.3.3 Array Method Sort 889

21.3.4 Array Method Copy 889

21.3.5 Array Method BinarySearch 889

21.3.6 Array Method GetEnumerator and Interface IEnumerator 889

21.3.7 Iterating Over a Collection with foreach 890

21.3.8 Array Methods Clear, IndexOf, LastIndexOf and Reverse 890

21.4 Dictionary Collections 890

Table of Contents

21.4.1 Dictionary Fundamentals	891
21.4.2 Using the SortedDictionary Collection	892
21.5 Generic LinkedList Collection	896
21.6 C# 6 Null Conditional Operator ?[]	900
21.7 C# 6 Dictionary Initializers and Collection Initializers	901
21.8 Delegates	901
21.8.1 Declaring a Delegate Type	903
21.8.2 Declaring a Delegate Variable	903
21.8.3 Delegate Parameters	904
21.8.4 Passing a Method Name Directly to a Delegate Parameter	904
21.9 Lambda Expressions	904
21.9.1 Expression Lambdas	906
21.9.2 Assigning Lambdas to Delegate Variables	907
21.9.3 Explicitly Typed Lambda Parameters	907
21.9.4 Statement Lambdas	907
21.10 Introduction to Functional Programming	907
21.11 Functional Programming with LINQ Method-Call Syntax and Lambdas	909
21.11.1 LINQ Extension Methods Min, Max, Sum and Average	912
21.11.2 Aggregate Extension Method for Reduction Operations	912
21.11.3 The Where Extension Method for Filtering Operations	914
21.11.4 Select Extension Method for Mapping Operations	915
21.12 PLINQ: Improving LINQ to Objects Performance with Multicore	915
21.13 (Optional) Covariance and Contravariance for Generic Types	919
21.14 Wrap-Up	921
22 Databases and LINQ	933

Table of Contents

22.1	Introduction	934
22.2	Relational Databases	935
22.3	A Books Database	936
22.4	LINQ to Entities and the ADO.NET Entity Framework	940
22.5	Querying a Database with LINQ	941
22.5.1	Creating the ADO.NET Entity Data Model Class Library	943
22.5.2	Creating a Windows Forms Project and Configuring It to Use the Entity Data Model	947
22.5.3	Data Bindings Between Controls and the Entity Data Model	949
22.6	Dynamically Binding Query Results	955
22.6.1	Creating the Display Query Results GUI	956
22.6.2	Coding the Display Query Results App	957
22.7	Retrieving Data from Multiple Tables with LINQ	959
22.8	Creating a Master/Detail View App	965
22.8.1	Creating the Master/Detail GUI	965
22.8.2	Coding the Master/Detail App	967
22.9	Address Book Case Study	968
22.9.1	Creating the Address Book Apps GUI	970
22.9.2	Coding the Address Book App	971
22.10	Tools and Web Resources	975
22.11	Wrap-Up	975
23	Asynchronous Programming with async and await	982
23.1	Introduction	983
23.2	Basics of async and await	985
23.2.1	async Modifier	985
23.2.2	await Expression	985
23.2.3	async, await and Threads	985
23.3	Executing an Asynchronous Task from a GUI App	986

Table of Contents

23.3.1 Performing a Task Asynchronously 986

23.3.2 Method calculateButton_Click 988

23.3.3 Task Method Run: Executing Asynchronously in a SeparateThread 989

23.3.4 awaiting the Result 989

23.3.5 Calculating the Next Fibonacci Value Synchronously 989

23.4 Sequential Execution of Two Compute-Intensive Tasks 990

23.5 Asynchronous Execution of Two Compute-Intensive Tasks 992

23.5.1 awaiting Multiple Tasks with Task Method WhenAll 995

23.5.2 Method StartFibonacci 996

23.5.3 Modifying a GUI from a Separate Thread 996

23.5.4 awaiting One of Several Tasks with Task Method WhenAny 996

23.6 Invoking a Flickr Web Service Asynchronously with HttpClient 997

23.6.1 Using Class HttpClient to Invoke a Web Service 1001

23.6.2 Invoking the Flickr Web Services flickr.photos.search Method 1001

23.6.3 Processing the XML Response 1002

23.6.4 Binding the Photo Titles to the ListBox 1003

23.6.5 Asynchronously Downloading an Images Bytes 1004

23.7 Displaying an Asynchronous Tasks Progress 1004

23.8 Wrap-Up 1008

Chapters on the Web

A Operator Precedence Chart 1016

B Simple Types 1018

C ASCII Character Set 1020

Appendices on the Web 1021

Index 1023

Symbols

Numerics

Table of Contents

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z