

# Contents at a Glance

---

<b><i>Introduction</i></b> .....	<b>1</b>
<b><i>Part I: Getting Started with C#</i></b> .....	<b>9</b>
Chapter 1: Creating Your First C# Console Application .....	11
<b><i>Part II: Basic C# Programming</i></b> .....	<b>25</b>
Chapter 2: Living with Variability — Declaring Value-Type Variables .....	27
Chapter 3: Smooth Operators .....	49
Chapter 4: Getting into the Program Flow.....	61
Chapter 5: Lining Up Your Ducks with Collections .....	89
Chapter 6: Pulling Strings .....	117
<b><i>Part III: Using Objects</i></b> .....	<b>147</b>
Chapter 7: Showing Some Class.....	149
Chapter 8: We Have Our Methods .....	161
Chapter 9: Let Me Say This about this .....	191
Chapter 10: Object-Oriented Programming — What’s It All About? .....	209
<b><i>Part IV: Object-Oriented Programming</i></b> .....	<b>217</b>
Chapter 11: Holding a Class Responsible.....	219
Chapter 12: Inheritance — Is That All I Get? .....	251
Chapter 13: Poly-what-ism? .....	275
Chapter 14: Interfacing with the Interface .....	303
<b><i>Part V: Now Showing in C# 3.0</i></b> .....	<b>331</b>
Chapter 15: Delegating Those Important Events .....	333
Chapter 16: Mary Had a Little Lambda Expression .....	353
Chapter 17: LINQing Up with Query Expressions .....	367
<b><i>Part VI: The Part of Tens</i></b> .....	<b>403</b>
Chapter 18: Ten Common Build Errors (And How to Fix Them) .....	405
<b><i>Index</i></b> .....	<b>417</b>



# Table of Contents

---

<b><i>Introduction</i></b> .....	<b>1</b>
What's New in C# 3.0 .....	2
About This Book.....	3
What You Need to Use the Book .....	3
How to Use This Book .....	4
How This Book Is Organized.....	4
Part I: Getting Started with C# .....	4
Part II: Basic C# Programming .....	4
Part III: Using Objects .....	5
Part IV: Object-Oriented Programming.....	5
Part V: Now Showing in C# 3.0.....	5
Part VI: The Part of Tens .....	5
About the Web site.....	6
Icons Used in This Book.....	6
Conventions Used in This Book .....	7
Where's the Code? And the Bonus Goodies?.....	7
Where to Go from Here.....	8
<b><i>Part I: Getting Started with C#</i></b> .....	<b>9</b>
<b>Chapter 1: Creating Your First C# Console Application</b> .....	<b>11</b>
Getting a Handle on Computer Languages, C#, and .NET.....	11
What's a program?.....	12
What's C#?.....	12
What's .NET? .....	13
What is Visual Studio 2008? What about Visual C#? .....	14
Creating Your First Console Application .....	15
Creating the source program .....	15
Taking it out for a test drive.....	18
Making Your Console App Do Something .....	19
Reviewing Your Console Application.....	20
The program framework.....	20
Comments .....	21
The meat of the program.....	21
Introducing the Toolbox Trick.....	22
Saving code in the Toolbox .....	23
Reusing code from the Toolbox.....	23

**Part II: Basic C# Programming.....25**

<b>Chapter 2: Living with Variability — Declaring Value-Type Variables .....</b>	<b>27</b>
Declaring a Variable .....	27
What's an int? .....	28
Rules for declaring variables .....	29
Variations on a theme — Different types of int .....	30
New feature — The BigInteger type .....	31
Representing Fractions .....	31
Handling Floating-Point Variables .....	32
Declaring a floating-point variable .....	33
Converting some more temperatures.....	34
Examining some limitations of floating-point variables .....	34
Using the Decimal Type — Is It an Integer or a Float?.....	35
Declaring a decimal.....	36
Comparing decimals, integers, and floating-point types.....	36
Examining the bool Type — Is It Logical?.....	37
Checking Out Character Types .....	37
The char variable type.....	38
Special chars .....	38
The string type .....	39
What's a Value-Type? .....	40
Comparing string and char .....	40
Is This a Leap Year? — DateTime.....	41
Declaring Numeric Constants .....	43
Changing Types — The Cast.....	44
New Feature: Letting the C# Compiler Infer Data Types .....	46
<b>Chapter 3: Smooth Operators .....</b>	<b>49</b>
Performing Arithmetic.....	49
Simple operators .....	49
Operating orders .....	50
The assignment operator .....	51
The increment operator .....	52
Performing Logical Comparisons — Is That Logical? .....	53
Comparing floating-point numbers:	
Is your float bigger than mine? .....	54
Compounding the confusion with compound logical operations .....	55
Matching Expression Types at TrackDownAMate.com.....	57
Calculating the type of an operation .....	57
Assigning types.....	59

---

<b>Chapter 4: Getting into the Program Flow</b> .....	<b>61</b>
Branching Out with if and switch.....	62
Introducing the if statement .....	63
Examining the else statement.....	65
Avoiding even the else.....	66
Nesting if statements .....	67
Running the switchboard .....	69
Here We Go Loop the Loop .....	72
Looping for a while.....	72
Doing the do . . . while loop.....	76
Breaking up is easy to do .....	77
Looping until you get it right .....	78
Focusing on scope rules .....	82
Looping a Specified Number of Times with for .....	82
An example.....	83
Why do you need another loop? .....	84
Nesting Loops.....	85
Don't goto Pieces .....	86
<b>Chapter 5: Lining Up Your Ducks with Collections</b> .....	<b>89</b>
The C# Array.....	90
The argument for the array.....	90
The fixed-value array .....	90
The variable-length array .....	92
The Length property.....	95
How do I initialize my array? .....	96
A Loop Made foreach Array.....	96
Sorting Through Arrays of Data .....	97
New Feature — Using var for Arrays .....	102
Loosening Up with C# Collections .....	103
Understanding Collection Syntax.....	104
C#'s collection classes .....	104
Figuring out <T>.....	104
Going generic .....	105
Using Lists.....	105
Using Dictionaries .....	108
New Feature — Array and Collection Initializers .....	110
Initializing arrays .....	110
Initializing collections.....	110
New Feature — Using Sets .....	112
On Not Using Old-Fashioned Collections .....	116

<b>Chapter 6: Pulling Strings</b> .....	<b>117</b>
The Union Is Indivisible, and So Are Strings.....	118
Performing Common Operations on a String .....	119
Comparing Strings.....	120
Equality for all strings: The Compare() method .....	121
Would you like your compares with or without case? .....	124
What If I Want to Switch Case? .....	125
Distinguishing between all-uppercase and all-lowercase strings .....	125
Converting a string to upper- or lowercase .....	125
Looping Through a String .....	126
Searching Strings.....	127
Can I find it? .....	127
Is my string empty? .....	128
Getting Input from the Command Line.....	128
Trimming excess white space.....	128
Parsing numeric input .....	129
Handling a series of numbers .....	131
Joining an array of strings into one string .....	133
Controlling Output Manually .....	134
Using the Trim() and Pad() methods.....	134
Using the Concatenate() method .....	136
Let's Split() that concatenate program .....	138
Formatting Your Strings Precisely .....	139
StringBuilder: Manipulating Strings More Efficiently .....	144

## ***Part III: Using Objects***..... **147**

<b>Chapter 7: Showing Some Class</b> .....	<b>149</b>
What's an Object? What's a Class?.....	150
Defining a class .....	150
What's the object? .....	151
Accessing the Members of an Object .....	152
An Example Object-Based Program .....	153
Discriminating Between Objects .....	154
Can You Give Me References?.....	155
Classes That Contain Classes Are the Happiest Classes in the World.....	157
Generating Static in Class Members .....	158
Defining const and readonly Data Members .....	160

<b>Chapter 8: We Have Our Methods</b> .....	<b>161</b>
Defining and Using a Method.....	161
An Example Method for Your Files .....	163
Having Arguments with Methods.....	169
Passing an argument to a method.....	170
Passing multiple arguments to methods.....	172
Matching argument definitions with usage.....	173
Overloading a method does not mean giving it too much to do.....	174
Implementing default arguments.....	175
Passing Value-Type and Reference-Type Arguments.....	177
Passing value-type arguments by value .....	177
Passing value-type arguments by reference .....	178
Passing reference-type arguments .....	180
Don't pass a variable to a method by reference twice simultaneously.....	180
Returning Values after Christmas .....	182
Returning a value via return postage.....	183
Returning a value using pass-by-reference .....	184
When do I return and when do I out? .....	184
Defining a method with no value.....	185
 <b>Chapter 9: Let Me Say This about this</b> .....	 <b>191</b>
Passing an Object to a Method.....	191
Defining Methods .....	193
Defining a static method.....	193
Defining an instance method .....	195
Expanding a method's full name .....	197
Accessing the Current Object.....	198
What is the this keyword? .....	199
When is this explicit? .....	200
What happens when I don't have this?.....	202
New Feature: Extension Methods.....	205
 <b>Chapter 10: Object-Oriented Programming — What's It All About?</b> .....	 <b>209</b>
Object-Oriented Concept #1 — Abstraction.....	209
Preparing procedural nachos .....	210
Preparing object-oriented nachos.....	211
Object-Oriented Concept #2 — Classification.....	211
Why Classify?.....	212

Object-Oriented Concept #3 — Usable Interfaces .....	213
Object-Oriented Concept #4 — Access Control .....	214
How Does C# Support Object-Oriented Concepts? .....	215

## ***Part IV: Object-Oriented Programming*.....217**

### **Chapter 11: Holding a Class Responsible .....219**

Restricting Access to Class Members .....	219
A public example of public BankAccount .....	220
Jumping ahead — other levels of security .....	222
Why Worry about Access Control? .....	223
Accessor methods .....	224
Access control to the rescue — an example .....	225
So what? .....	228
Defining Class Properties .....	228
Static properties .....	230
Properties with side effects .....	230
New Feature: Let the compiler write properties for you .....	231
Accessors with access levels .....	231
Getting Your Objects Off to a Good Start — Constructors .....	232
The C#-Provided Constructor .....	232
Replacing the Default Constructor .....	234
Constructing something .....	235
Executing the constructor from the debugger .....	237
Initializing an object directly with an initializer .....	239
Seeing that construction stuff with initializers .....	241
New Feature — Initializing an object without a constructor .....	242
Overloading the Constructor (Is That Like Overtaxing a Carpenter?) .....	243
Avoiding Duplication among Constructors .....	246
Being Object-Stingy .....	249

### **Chapter 12: Inheritance — Is That All I Get? .....251**

Inheriting a Class .....	252
Why Do You Need Inheritance? .....	254
A More Involved Example — Inheriting from a BankAccount Class .....	254
IS_A versus HAS_A — I'm So Confused .....	257
The IS_A relationship .....	258
Gaining access to BankAccount through containment .....	259
The HAS_A relationship .....	260
When to IS_A and When to HAS_A? .....	261

Other Features That Support Inheritance.....	261
Substitutable classes .....	262
Invalid casts at run time .....	262
Avoiding invalid conversions with the is operator .....	263
Avoiding invalid conversions with the as operator .....	264
The object Class.....	265
Inheritance and the Constructor.....	266
Invoking the default base class constructor.....	266
Passing arguments to the base class constructor — mama sing base.....	268
Getting specific with base .....	269
The Updated BankAccount Class.....	271
<b>Chapter 13: Poly-what-ism? .....</b>	<b>275</b>
Overloading an Inherited Method.....	276
It's a simple case of method overloading.....	276
Different class, different method.....	277
Peek-a-boo — hiding a base class method.....	277
Calling back to base .....	282
Polymorphism .....	284
What's wrong with using the declared type every time? .....	285
Using "is" to access a hidden method polymorphically .....	287
Declaring a method virtual and overriding it .....	288
Getting the most out of polymorphism — the do-to-each trick .....	291
The Class Business Card: ToString( ).....	291
C# During Its Abstract Period.....	291
Class factoring .....	292
I'm left with nothing but a concept — the abstract class .....	297
How do you use an abstract class? .....	298
Creating an abstract object — not! .....	300
Sealing a Class .....	300
<b>Chapter 14: Interfacing with the Interface .....</b>	<b>303</b>
What Is CAN_BE_USED_AS?.....	303
What Is an Interface? .....	305
How to implement an interface .....	305
How to name your interface.....	306
So why does C# include interfaces?.....	306
Mixing inheritance and interface implementation .....	307
And here's the payoff.....	307

Using an Interface .....	308
As a method return type.....	308
As the base type of an array or collection .....	309
As a more general type of object reference .....	309
Using C#'s Predefined Interface Types .....	310
Can I See a Program That CAN_BE_USED_AS an Example? .....	310
Creating your own interface at home in your spare time .....	311
Implementing the incomparable IComparable<T> interface .....	312
Putting it all together .....	313
Getting back to the Main() event.....	316
Unifying Class Hierarchies .....	317
Hiding Behind an Interface .....	319
Inheriting an Interface .....	322
Using Interfaces to Manage Change in Object-Oriented Programs .....	323
Flexible dependencies through interfaces .....	324
Abstract or concrete? When to use an abstract class, when to use an interface.....	324
Doing HAS_A with interfaces .....	325
The Strategy Design Pattern .....	326
What makes this pattern so useful? .....	327

## ***Part V: Now Showing in C# 3.0 .....*** **331**

### **Chapter 15: Delegating Those Important Events .....** **333**

E.T. Phone Home — The Callback Problem .....	333
What's a Delegate? .....	334
Pass Me the Code, Please — Examples .....	336
I delegated the example to Igor .....	337
First, a very simple example .....	337
How About a More Real-World Example?.....	339
Getting an overview of the bigger example .....	339
Putting the app together .....	339
Looking at the code.....	341
Tracking the delegate life cycle .....	342
Shh! Keep It Quiet — Anonymous Methods .....	345
Stuff Happens — C# Events .....	346
The Observer design pattern.....	346
What's an event? Publish and Subscribe.....	347
How a publisher advertises its events .....	347
How do subscribers subscribe to an event?.....	348
How do I publish an event? .....	349

How do I pass extra information to an event handler? .....	350
A recommended way to raise your events.....	350
How do observers “handle” an event? .....	351
<b>Chapter 16: Mary Had a Little Lambda Expression .....</b>	<b>353</b>
Collections and the Do-to-Each Problem .....	353
New Feature: Using Lambda Expressions .....	354
The C# way to do-to-each.....	354
Grist for the mill: Data for the examples .....	355
Looking for answers: Some Find( ) examples.....	355
Taking action: Some ForEach( ) examples.....	358
A lambda based on a custom delegate .....	361
Summary of the examples.....	362
Details of lambda syntax .....	362
Using Lambda Expressions for Any Delegate.....	365
<b>Chapter 17: LINQing Up with Query Expressions .....</b>	<b>367</b>
Reviewing the C# 3.0 Features.....	368
What’s a Query, and Why Should I Care?.....	368
Why is “language-integrated” query important?.....	370
The general form of a C# query.....	370
LINQ from low Earth orbit.....	371
Querying in C# 3.0.....	372
The three most important query operators .....	373
What is deferred query execution?.....	375
Evaluating a query immediately.....	376
What Can I Do with a Query?.....	376
Filtering Out the Espresso Grounds .....	377
Example 1: Which customers live in Colorado? .....	377
Selecting and Grouping What You Want.....	379
Example 2: Give me a sequence of integers	
two higher than the inputs.....	379
Example 3: I want just the name and phone number	
for each customer.....	380
Using anonymous types .....	381
Example 4: Transforming numbers into their names .....	382
Example 5: Grouping names by their first character .....	383
Example 6: Screening out some groups.....	386
Example 7: Grouping products by category .....	387
Sorting the Stuff You Select.....	389
Example 8: Ordering the groups from Examples 5 and 6 .....	389

Example 9: Ordering within the groups .....	390
Example 10: Another two-level sort .....	390
Example 11: Ordering in descending order .....	391
Is There Any Method to Querying? .....	391
Example 12: Writing queries with methods rather than with keywords .....	393
About train cars: A detour for newbies .....	394
End of detour — All aboard! .....	395
Generic methods .....	395
What's inside Where( )? .....	396
Example 13: Give me all customers, but just their names and numbers .....	397
Counting, Summing, Averaging, and Stuff Like That .....	397
Example 14: Counting product types in our inventory .....	398
Example 15: Adding up the numbers array .....	398
Example 16: Summing the cost of order O .....	399
Example 17: Computing nine factorial (9!) .....	399
Treating an Object Like a Collection .....	401

## ***Part VI: The Part of Tens*.....403**

### **Chapter 18: Ten Common Build Errors (And How to Fix Them) . . . .405**

The name 'memberName' does not exist in the class or namespace 'className' .....	406
Cannot implicitly convert type 'x' into 'y' .....	407
'className.memberName' is inaccessible due to its protection level .....	409
Use of unassigned local variable 'n' .....	410
Unable to copy the file 'programName.exe' to 'programName.exe'. The process cannot . . . ..	411
'subclassName.methodName' hides inherited member 'baseclassName.methodName'. Use the new keyword if hiding was intended .....	412
'subclassName' : cannot inherit from sealed class 'baseclassName' .....	413
'className' does not implement interface member 'methodName' .....	413
'methodName' : not all code paths return a value .....	414
} expected .....	415

## ***Index*.....417**