

Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™



Beginning **Visual C#® 2010**

Karli Watson, Christian Nagel, Jacob Hammer Pedersen, Jon D. Reid, Morgan Skinner

Table of Contents

[Title Page](#)

[Copyright](#)

[Dedication](#)

[About the Authors](#)

[About the Technical Editor](#)

[Credits](#)

[Acknowledgments](#)

[Introduction](#)

[Who This Book Is For](#)

[What's New in This Edition](#)

[How This Book Is Structured](#)

[What You Need to Use This Book](#)

[Conventions](#)

[Source Code](#)

[Errata](#)

p2p.wrox.com

[Part I: The C# Language](#)

[Chapter 1: Introducing C#](#)

[*What is the .NET Framework?*](#)

[*What is C#?*](#)

[*Visual Studio 2010*](#)

[*Summary*](#)

[*Chapter 2: Writing a C# Program*](#)

[*The Development Environments*](#)

[*Console Applications*](#)

[*Windows Forms Applications*](#)

[*Summary*](#)

[*Chapter 3: Variables and Expressions*](#)

[*Basic C# Syntax*](#)

[*Basic C# Console Application Structure*](#)

[*Variables*](#)

[*Expressions*](#)

[*Summary*](#)

[*Chapter 4: Flow Control*](#)

[*Boolean Logic*](#)

[*The goto Statement*](#)

[*Branching*](#)

[*Looping*](#)

[*Summary*](#)

[*Chapter 5: More About Variables*](#)

[*Type Conversion*](#)

[*Complex Variable Types*](#)

[*String Manipulation*](#)

[Summary](#)

[Chapter 6: Functions](#)

[Defining and Using Functions](#)

[Variable Scope](#)

[The Main\(\) Function](#)

[Struct Functions](#)

[Overloading Functions](#)

[Delegates](#)

[Summary](#)

[Chapter 7: Debugging and Error Handling](#)

[Debugging in VS and VCE](#)

[Error Handling](#)

[Summary](#)

[Chapter 8: Introduction to Object-Oriented Programming](#)

[What Is Object-Oriented Programming?](#)

[OOP Techniques](#)

[OOP in Windows Applications](#)

[Summary](#)

[What You Learned in this Chapter](#)

[Chapter 9: Defining Classes](#)

[Class Definitions in C#](#)

[System.Object](#)

[Constructors and Destructors](#)

[*OOP Tools in VS and VCE*](#)
[*Class Library Projects*](#)
[*Interfaces Versus Abstract Classes*](#)
[*Struct Types*](#)
[*Shallow Copying Versus Deep Copying*](#)
[*Summary*](#)

[*Chapter 10: Defining Class Members*](#)

[*Member Definitions*](#)
[*Additional Class Member Topics*](#)
[*Interface Implementation*](#)
[*Partial Class Definitions*](#)
[*Partial Method Definitions*](#)
[*Example Application*](#)
[*The Call Hierarchy Window*](#)
[*Summary*](#)

[*Chapter 11: Collections, Comparisons, and Conversions*](#)

[*Collections*](#)
[*Comparisons*](#)
[*Conversions*](#)
[*Summary*](#)

[*Chapter 12: Generics*](#)

[*What Are Generics?*](#)
[*Using Generics*](#)
[*Defining Generic Types*](#)
[*Variance*](#)

Summary

Chapter 13: Additional OOP Techniques

The :: Operator and the Global Namespace Qualifier

Custom Exceptions

Events

Expanding and Using CardLib

Summary

Chapter 14: C# Language Enhancements

Initializers

Type Inference

Anonymous Types

Dynamic Lookup

Advanced Method Parameters

Extension Methods

Lambda Expressions

Summary

Part II: Windows Programming

Chapter 15: Basic Windows Programming

Controls

The Button Control

[The Label and LinkLabel Controls](#)

[The TextBox Control](#)

[The RadioButton and CheckBox Controls](#)

[The RichTextBox Control](#)

[The ListBox and CheckedListBox Controls](#)

[The ListView Control](#)

[The TabControl Control](#)

[Summary](#)

[Chapter 16: Advanced Windows](#)

[Forms Features](#)

[Menus and Toolbars](#)

[Toolbars](#)

[SDI and MDI Applications](#)

[Building MDI Applications](#)

[Creating Controls](#)

[Summary](#)

[Chapter 17: Deploying Windows](#)

[Applications](#)

[Deployment Overview](#)

[ClickOnce Deployment](#)

[Visual Studio Setup and Deployment](#)

[Project Types](#)

[Microsoft Windows Installer Architecture](#)

[Creating an Installation Package for the](#)

[MDI Editor](#)

[Building the Project](#)

[Installation](#)

Summary

Part III: Web Programming

Chapter 18: ASP.NET Web Programming

Overview of Web Applications

ASP.NET Runtime

Creating a Simple Page

Server Controls

ASP.NET Postback

ASP.NET AJAX Postback

Input Validation

State Management

Styles

Master Pages

Site Navigation

Authentication and Authorization

Reading from and Writing to a SQL Server

Database

Summary

Chapter 19: Web Services

Where to Use Web Services

Application Architecture

Web Services Architecture

Web Services and the .NET Framework

Creating a Simple ASP.NET Web Service

Testing the Web Service

[*Implementing a Windows Client*](#)
[*Calling the Service Asynchronously*](#)
[*Implementing an ASP.NET Client*](#)
[*Passing Data*](#)
[*Summary*](#)

[*Chapter 20: Deploying Web Applications*](#)

[*Internet Information Services*](#)
[*IIS Configuration*](#)
[*Copying a Website*](#)
[*Publishing a Web Application*](#)
[*Windows Installer*](#)
[*Summary*](#)

[*Part IV: Data Access*](#)

[*Chapter 21: File System Data*](#)

[*Streams*](#)
[*The Classes for Input and Output*](#)
[*Serialized Objects*](#)
[*Monitoring the File System*](#)
[*Summary*](#)

[*Chapter 22: XML*](#)

[*XML Documents*](#)
[*Using XML in Your Application*](#)
[*Summary*](#)

Chapter 23: Introduction to LINQ

First LINQ Query

Using the LINQ Method Syntax

Ordering Query Results

orderby Clause

Ordering Using Method Syntax

Querying a Large Data Set

Aggregate Operators

Querying Complex Objects

Projection: Creating New Objects in Queries

Projection: Method Syntax

Select Distinct Query

Any and All

Ordering by Multiple Levels

Multi-Level Ordering Method Syntax:

ThenBy

Group Queries

Take and Skip

First and FirstOrDefault

Set Operators

Joins

Summary

Chapter 24: Applying LINQ

LINQ Varieties

Using LINQ with Databases

Installing SQL Server and the Northwind

Sample Data

First LINQ to Database Query

[***Navigating Database Relationships***](#)
[***Using LINQ with XML***](#)
[***LINQ to XML Functional Constructors***](#)
[***Saving and Loading an XML Document***](#)
[***Working with XML Fragments***](#)
[***Generating XML from Databases***](#)
[***How to Query an XML Document***](#)
[***Using LINQ to XML Query Members***](#)
[***Summary***](#)

[**Part V: Additional Techniques**](#)

[**Chapter 25: Windows Presentation Foundation**](#)

[***What Is WPF?***](#)
[***Anatomy of a Basic WPF Application***](#)
[***WPF Fundamentals***](#)
[***Programming with WPF***](#)
[***Summary***](#)

[**Chapter 26: Windows Communication Foundation**](#)

[***What Is WCF?***](#)
[***WCF Concepts***](#)
[***WCF Programming***](#)
[***Summary***](#)

Chapter 27: Windows Workflow Foundation

[Hello World](#)

[Workflows and Activities](#)

[Arguments and Variables](#)

[Custom Activities](#)

[Summary](#)

Appendix A: Exercise Solutions

[Chapter 3 Solutions](#)

[Chapter 4 Solutions](#)

[Chapter 5 Solutions](#)

[Chapter 6 Solutions](#)

[Chapter 7 Solutions](#)

[Chapter 8 Solutions](#)

[Chapter 9 Solutions](#)

[Chapter 10 Solutions](#)

[Chapter 11 Solutions](#)

[Chapter 12 Solutions](#)

[Chapter 13 Solutions](#)

[Chapter 14 Solutions](#)

[Chapter 15 Solutions](#)

[Chapter 16 Solutions](#)

[Chapter 17 Solutions](#)

[Chapter 18 Solutions](#)

[Chapter 19 Solutions](#)

[Chapter 20 Solutions](#)

[Chapter 21 Solutions](#)

[Chapter 22 Solutions](#)

[Chapter 23 Solutions](#)

[Chapter 24 Solutions](#)

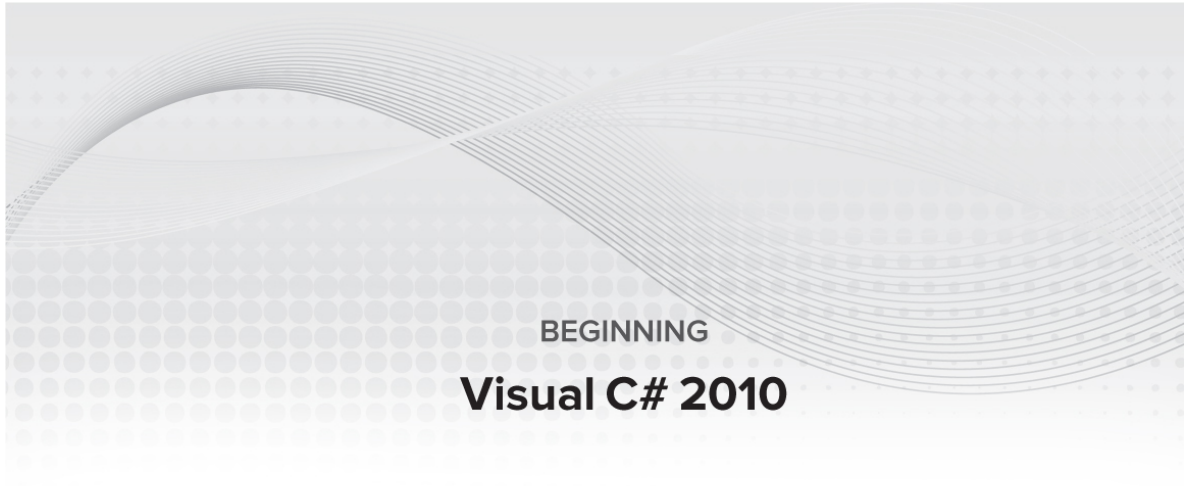
[Chapter 25 Solutions](#)

[Chapter 26 Solutions](#)

[Answers to Chapter 27 Exercises](#)

[Index](#)

[Advertisement](#)



BEGINNING
Visual C# 2010

Karli Watson
Christian Nagel
Jacob Hammer Pedersen
Jon Reid
Morgan Skinner



WILEY
Wiley Publishing, Inc.

Beginning Visual C# 2010

Published by

Wiley Publishing, Inc.

10475 Crosspoint Boulevard

Indianapolis, IN 46256

www.wiley.com

Copyright © 2010 by Wiley Publishing, Inc., Indianapolis,
Indiana

Published simultaneously in Canada

ISBN: 978-0-470-50226-6

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 also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic books.

Library of Congress Control Number: 2010920663

Trademarks: Wiley, the Wiley logo, Wrox, the Wrox logo, Wrox 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. C# is a registered trademark of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners. Wiley Publishing, Inc. is not associated with any product or vendor mentioned in this book.

for Donna
—KARLI WATSON

About the Authors

KARLI WATSON is consultant at Infusion Development (www.infusion.com), a technology architect at Boost.net (www.boost.net), and a freelance IT specialist, author, and developer. For the most part, he immerses himself in .NET (in particular C# and lately WPF) and has written numerous books in the field for several publishers. He specializes in communicating complex ideas in a way that is accessible to anyone with a passion to learn, and spends much of his time playing with new technology to find new things to teach people about.

During those (seemingly few) times where he isn't doing the above, Karli will probably be wishing he was hurtling down a mountain on a snowboard. Or possibly trying to get his novel published. Either way, you'll know him by his brightly colored clothes. You can also find him tweeting online at www.twitter.com/karlequin, and maybe one day he'll get around to making himself a website. Karli authored chapters 1 through 14, 21, 25 and 26.

CHRISTIAN NAGEL is a Microsoft Regional Director and Microsoft MVP, an associate of Thinktecture, and owner of CN Innovation. He is a software architect and developer who offers training and consulting on how to develop Microsoft .NET solutions. He looks back 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 numerous .NET solutions. With his profound knowledge of Microsoft technologies, he has written numerous .NET books, and is certified as a Microsoft Certified Trainer and Professional Developer. Christian speaks at international conferences

such as TechEd and Tech Days, and started INETA Europe to support .NET user groups. You can contact Christian via his web sites, www.cninnovation.com and www.thinktecture.com and follow his tweets on www.twitter.com/christiannagel. Christian wrote chapters 17 through 20.

JACOB HAMMER PEDERSEN is a Senior Application Developer at Elbek & Vejrup. He just about started programming when he was able to spell the word 'BASIC', which, incidentally is the first programming language he ever used. He started programming the PC in the early '90s, using Pascal but soon changed his focus to C++, which still holds his interest. In the mid '90s his focus changed again, this time to Visual Basic. In the summer of 2000 he discovered C# and has been happily exploring it ever since. Primarily working on the Microsoft platforms, his other expertise includes MS Office development, SQL Server, COM and Visual Basic.Net.

A Danish citizen, Jacob works and lives in Aarhus, Denmark. He authored chapters 15, 16, and 22.

JON D. REID is a software engineering manager at Metrix LLC, an ISV of field service management software for the Microsoft environment. He has co-authored a variety .NET books, including *Beginning Visual C# 2008*, *Beginning C# Databases: From Novice to Professional*, *Pro Visual Studio .NET*, and many others. Jon wrote chapters 23 and 24.

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's used all sorts of languages and platforms, including VAX Macro Assembler, Pascal, Modula2, Smalltalk, X86 assembly language, PowerBuilder, C/C++, VB, and currently C# (of course). He's been programming in .NET since the PDC release in 2000, and liked it so much he joined Microsoft in

2001. He now works in premier support for developers and spends most of his time assisting customers with C#. Morgan wrapped up the book by authoring chapter 27. You can reach Morgan at www.morganskinner.com.

About the Technical Editor

A “blue-badge” .NET architect and developer at Intel Corporation since March 2007, **Doug Holland** is part of the Visual Computing Group and is presently working within an advanced tools and development team with an emphasis on chipset and driver testing. Doug Holland holds a Master's Degree in Software Engineering from Oxford University and has been awarded both the Microsoft MVP and Intel Black Belt Developer awards. Outside of work, Doug enjoys spending time with his wife and four children; and is also an officer in the Civil Air Patrol/U.S. Air Force Auxiliary. Beyond architecting and developing software you can often find Doug at the local airport flying Cessnas over the California landscape.

Credits

Acquisitions Editor

Paul Reese

Development Editor

Maryann Steinhart

Project Editor

Ami Frank Sullivan

Technical Editor

Doug Holland

Production Editor

Rebecca Anderson

Copy Editor

Luann Rouff

Editorial Director

Robyn B. Siesky

Editorial Manager

Mary Beth Wakefield

Associate Director of Marketing

David Mayhew

Production Manager

Tim Tate

Vice President and Executive Group Publisher

Richard Swadley

Vice President and Executive Publisher

Barry Pruett

Associate Publisher

Jim Minatel

Project Coordinator, Cover

Lynsey Stanford

Proofreader

Josh Chase, Word One

Indexer

J & J Indexing

Cover Designer

Michael E. Trent

Cover Image

© Lisa Loyd/istockphoto

Acknowledgments

FROM KARLI WATSON: Thanks to all at Wiley for their support and assistance on this project, as well as their understanding and flexibility in dealing with an author who never seems to have enough time to write. Special thanks to my editor for this book, Ami Sullivan, for adding sparkle and making this book shine. Also, thanks to friends, family, and work colleagues for understanding why I haven't have time for much socializing lately, and to Donna, as always, for all her support and for putting up with all the late nights.

FROM CHRISTIAN NAGEL: To my two girls Angela and Stephanie. It's great to have you. Thanks for your great support and the big love you gave me during the hardest time of my life in 2009. Without you I couldn't have made it through. Stephanie, while not born yet, you were my biggest motivation during that time. I love you both!

Also, a big thank you to my co-authors and the team at Wrox/Wiley for getting a great book out.

Introduction

C# is a relatively new language that was unveiled to the world when Microsoft announced the first version of its .NET Framework in July 2000. Since then its popularity has rocketed, and it has arguably become the language of choice for both Windows and Web developers who use the .NET Framework. Part of the appeal of C# comes from its clear syntax, which derives from C/C++ but simplifies some things that have previously discouraged some programmers. Despite this simplification, C# has retained the power of C++, and there is now no reason not to move into C#. The language is not difficult and it's a great one to learn elementary programming techniques with. This ease of learning, combined with the capabilities of the .NET Framework, make C# an excellent way to start your programming career.

The latest release of C#, C# 4, which is included with version 4 of the .NET Framework, builds on the existing successes and adds even more attractive features. The latest release of Visual Studio (Visual Studio 2010), and the Express line of development tools (including Visual C# 2010 Express) also bring many tweaks and improvements to make your life easier and dramatically increase your productivity.

This book is intended to teach you about all aspects of C# programming, from the language itself, through Windows and Web programming, to making use of data sources, and finally to some new and advanced techniques. You'll also learn about the capabilities of Visual C# 2010 Express, Visual Web Developer 2010 Express, and Visual Studio 2010, and all the ways that these products can aid your application development.

The book is written in a friendly, mentor-style fashion, with each chapter building on previous ones, and every effort is made to ease you into advanced techniques painlessly. At no point will technical terms appear from nowhere to discourage you from continuing; every concept is introduced and discussed as required. Technical jargon is kept to a minimum; but where it is necessary, it too is properly defined and laid out in context.

The authors of this book are all experts in their field, and are all enthusiastic in their passion for both the C# language and the .NET Framework. Nowhere will you find a group of people better qualified to take you under their collective wing and nurture your understanding of C# from first principles to advanced techniques. Along with the fundamental knowledge it provides, this book is packed full of helpful hints, tips, exercises, and full-fledged example code (available for download at p2p.wrox.com) that you will find yourself returning to repeatedly as your career progresses.

We pass this knowledge on without begrudging it, and hope that you will be able to use it to become the best programmer you can be. Good luck, and all the best!

Who This Book Is For

This book is for everyone who wants to learn how to program in C# using the .NET Framework. The early chapters cover the language itself, assuming no prior programming experience. If you have programmed in other languages before, then much of the material in these chapters will be familiar. Many aspects of C# syntax are shared with other languages, and many structures are common to practically all programming languages (such as looping and branching structures). However, even if you are an experienced programmer you will benefit from looking

through these chapters to learn the specifics of how these techniques apply to C#.

If you are new to programming, you should start from the beginning. If you are new to the .NET Framework but know how to program, you should read Chapter 1 and then skim through the next few chapters before continuing with the application of the C# language. If you know how to program but haven't encountered an object-oriented programming language before, you should read the chapters from Chapter 8 onward.

Alternatively, if you already know the C# language you may wish to concentrate on the chapters dealing with the most recent .NET Framework and C# language developments, specifically the chapters on collections, generics, and C# 4 language enhancements (Chapters 11 to 14), or skip the first section of the book completely and start with Chapter 15.

The chapters in this book have been written with a dual purpose in mind: They can be read sequentially to provide a complete tutorial in the C# language, and they can be dipped into as required as reference material.

In addition to the core material, starting with Chapter 3 each chapter also includes a selection of exercises at the end, which you can work through to ensure that you have understood the material. The exercises range from simple multiple choice or true/false questions to more complex exercises that require you to modify or build applications. The answers to all the exercises are provided as a download from the book's Web page at www.wrox.com.

What's New in This Edition

This book has been given plenty of love and attention to coincide with the release of C# 4 and .NET 4. Every chapter has been given an overhaul, with less relevant material

removed, and new material added. All of the code has been tested against the latest version of the development tools used, and all of the screenshots have been retaken in Windows 7 to provide the most current windows and dialogs.

Although we hate to admit our own fallibility, any errors from previous editions have been fixed, and many other reader comments have been addressed. Hopefully, we haven't introduced many new errors, but any that may have slipped through our web of experts will be corrected online as soon as we find them.

New highlights of this edition include the following:

- Additional and improved code examples for you to try out
- Coverage of everything that's new in C# 4, from simple language improvements such as named and optional method parameters, to advanced techniques such as variance in generic types
- Streamlined coverage of advanced techniques to focus on those most appropriate to beginners without getting too obscure

How This Book Is Structured

This book is divided into six sections:

- **Introduction:** Purpose and general outline of the book's contents
- **The C# Language:** Covers all aspects of the C# language, from the fundamentals to object-oriented techniques
- **Windows Programming:** How to write Windows applications in C# and how to deploy them
- **Web Programming:** Web application development, Web services, and Web application deployment

- **Data Access:** How to use data in your applications, including data stored in files on your hard disk, data stored in XML format, and data in databases
- **Additional Techniques:** An examination of some extra ways to use C# and the .NET framework, including WPF, WCF, and WF—technologies introduced with .NET 3.0 and enhanced for .NET 4.

The following sections describe the chapters in the five major parts of this book.

The C# Language (Chapters 1-14)

Chapter 1 introduces you to C# and how it fits into the .NET landscape. You'll learn the fundamentals of programming in this environment, and how Visual C# 2010 Express (VCE) and Visual Studio 2010 (VS) fit in.

Chapter 2 starts you off with writing C# applications. You'll look at the syntax of C# and put the language to use with sample command-line and Windows applications. These examples will demonstrate just how quick and easy it can be to get up and running, and along the way you'll be introduced to the VCE and VS development environments and the basic windows and tools that you'll be using throughout the book.

Next you'll learn more about the basics of the C# language. You'll learn what variables are and how to manipulate them in **Chapter 3**. You'll enhance the structure of your applications with flow control (looping and branching) in **Chapter 4**, and see some more advanced variable types such as arrays in **Chapter 5**. In **Chapter 6** you'll start to encapsulate your code in the form of functions, which make it much easier to perform repetitive operations and make your code much more readable.

By the beginning of **Chapter 7** you'll have a handle on the fundamentals of the C# language, and will focus on debugging your applications. This involves looking at

outputting trace information as your applications are executed, and at how VS can be used to trap errors and lead you to solutions for them with its powerful debugging environment.

From **Chapter 8** onward you'll learn about object-oriented programming (OOP), starting with a look at what this term means, and an answer to the eternal question "What is an object?" OOP can seem quite difficult at first. The whole of Chapter 8 is devoted to demystifying it and explaining what makes it so great, and you won't actually deal with much C# code until the very end of the chapter.

Everything changes in **Chapter 9**, when you put theory into practice and start using OOP in your C# applications. This is where the true power of C# lies. You'll start by looking at how to define classes and interfaces, and then move on to class members (including fields, properties, and methods) in **Chapter 10**. At the end of that chapter you'll start to assemble a card game application, which is developed over several chapters, and will help to illustrate OOP.

Once you've learned how OOP works in C#, **Chapter 11** moves on to look at common OOP scenarios, including dealing with collections of objects, and comparing and converting objects. **Chapter 12** takes a look at a very useful feature of C# that was introduced in .NET 2.0: generics, which enables you to create very flexible classes. Next, **Chapter 13** continues the discussion of the C# language and OOP with some additional techniques, notably events, which become very important in, for example, Windows programming. Finally, **Chapter 14** focuses on C# language features that were introduced with versions 3.0 and 4 of the language.

Windows Programming (Chapters 15-17)