

THE INTEGRATED .NET SERIES FROM OBJECT INNOVATIONS

# INTRODUCTION TO C# USING .NET

ROBERT J. OBERG

- Comprehensive, hands-on introduction to object-oriented programming in C#
- Running case study: build a complete system using C# and .NET
- Coverage of key .NET classes: collections, files, delegates, events, multiple threads, attributes, and Windows Forms

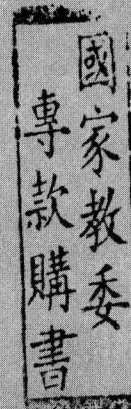




# INTRODUCTION TO C# USING

江苏工业学院图书馆  
藏书章

ROBERT J. OBERG



## Library of Congress Cataloging-in-Publication Data

*A catalog record for this book  
can be obtained from the Library of Congress*

Editorial/Production Supervision: *Nick Radhuber*  
Acquisitions Editor: *Jill Harry*  
Marketing Manager: *Dan DePasquale*  
Manufacturing Buyer: *Maura Zaldivar*  
Cover Design: *Anthony Gemmellaro*  
Cover Design Direction: *Jerry Votta*  
Interior Series Design: *Gail Cocker-Bogusz*



© 2002 by Robert J. Oberg  
Published by Prentice Hall PTR  
Prentice-Hall, Inc.  
Upper Saddle River, NJ 07458

Prentice Hall books are widely used by corporations and government agencies for training, marketing, and resale.

The publisher offers discounts on this book when ordered in bulk quantities. For more information, contact Corporate Sales Department, phone: 800-382-3419; fax: 201-236-7141; email: [corpsales@prentice-hall.com](mailto:corpsales@prentice-hall.com)

Or write Corporate Sales Department, Prentice Hall PTR, One Lake Street, Upper Saddle River, NJ 07458.

Product and company names mentioned herein are the trademarks or registered trademarks of their respective owners. The Electronic Commerce Game™ is a trademark of Object Innovations, Inc.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-041801-3

Pearson Education LTD.  
Pearson Education Australia PTY, Limited  
Pearson Education Singapore, Pte. Ltd.  
Pearson Education North Asia Ltd.  
Pearson Education Canada, Ltd.  
Pearson Educación de Mexico, S.A. de C.V.  
Pearson Education—Japan  
Pearson Education Malaysia, Pte. Ltd.

## P R E F A C E

Microsoft's .NET is a revolutionary advance in programming technology that greatly simplifies application development and is a good match for the emerging paradigm of Web-based services as opposed to proprietary applications. Part of this technology is a new language, C#. This new language combines the power of C++ and the ease of development of Visual Basic™. It bears striking resemblance to Java™ and improves upon that language. C# may well become the dominant language for building applications on Microsoft® platforms.

This book is a practical introduction to programming in C# utilizing the services provided by .NET. This book emphasizes the C# language. It is part of the Prentice Hall/Object Innovations series of books on .NET technology.

This book is intended to be fully accessible to programmers who do not already have a strong background in object-oriented programming in C-like languages such as C++ or Java. It is ideal, for example, for Visual Basic or COBOL programmers who desire to learn C#. The book may also be read by more experienced programmers who desire a simple and concise introduction to C# with many example programs. It is structured so that more experienced programmers can cleanly skip the material they already know. Although designed for working professionals, the book includes enough detail, careful explanations and sample programs so that it can be useful as a college textbook.

An important thrust of this book is to teach C# programming from an object-oriented perspective. It is often difficult for programmers trained originally in a procedural language to start "thinking in objects." This book introduces object-oriented concepts early, and C# is developed in a way that leverages its object-orientation. A banking system case study is used to illustrate creating a complete system using C# and .NET. Besides supporting traditional object-oriented features such as classes, inheritance, and polymorphism, C# introduces several additional features, such as properties, indexers, delegates, events, and interfaces that make C# a compelling language for developing object-oriented and component-based systems. This book provides thorough coverage of all these features.

C# as a language is elegant and powerful. But to fully utilize its capabilities you need to have a good understanding of how it works with the .NET Framework. The book explores several important interactions between C# and the .NET Framework, and it includes an introduction to major classes for collections, files, threads, and user interface.



## ORGANIZATION

The book is organized into four major parts and is structured in a manner to make it easy for you to navigate to what you most need to know. Part 1, which should be read by everyone, begins with an introduction to the .NET Framework, which is the underpinning for all applications and services in the .NET environment. Next comes a short introduction to hands-on programming using C#, so that you can start writing code on .NET right away. The third chapter introduces Visual Studio.NET, the latest incarnation of Microsoft's popular Visual Studio development environment. The new Visual Studio has many features that make application development easier and more pleasant. You will be equipped to use Visual Studio throughout the rest of the book.

Part 2 covers the C-like features of C#, which are shared by C, C++, Java, and various scripting languages. Thus, if you know any of these C-like languages, you will have a definite leg up in learning C#, and you can quickly skim this section, paying attention to the information in the sidebars. A sidebar will alert you the first time a concept new to C# is introduced. Then you will know to look for further elaborations where they occur later. If you are not familiar with C or a similar language, this section is for you. It will quickly bring you up to speed on the core topics of data types, operators, and control structures.

Part 3 is the core of the book, systematically covering the features of the C# programming language that go beyond C. The object-oriented features of C# are covered gradually and thoroughly, making this part of the book accessible to readers without OOP background. A case study is used, illustrating how the object-oriented features of C# work in combination. This case study is progressively built from Chapters 12 through 18. The C# data types, based on the .NET Common Type System, are explored in detail. We cover features new in C# such as properties and indexers. We cover practical issues of formatting and conversions, and we discuss the important topic of exceptions. We conclude this part with a study of interfaces, which provide a better level of abstraction in expressing system functionality.

Part 4 explores thoroughly the relationships between C# and the .NET Framework and introduces some important .NET services. We introduce collections, which generalize arrays, and we examine fundamental operations such as copying and comparing objects. In .NET, interfaces are provided for such basic operations, which makes for a very flexible architecture, as different classes can implement these interfaces in a manner appropriate for them. The .NET Framework provides a very flexible callback mechanism, known as delegates, which has many applications. Delegates are the foundation of events, and they are also used in starting threads. We look at directories and files, multiple thread programming, and attributes. Attributes are a powerful

mechanism in .NET, enabling the programmer to accomplish tasks declaratively, with writing little or no code. You can implement your own custom attributes in C#. You can read information about custom attributes, or any other metadata, by a mechanism known as “reflection.” C# permits you to code at a lower level by writing “unsafe” code, which can help you interoperate with legacy code. The book concludes with introductions to components and to Windows programming.

## SAMPLE PROGRAMS

The only way to really learn a programming language is to read and write many, many programs, including some of reasonable size. This book provides many small programs that illustrate pertinent features of C# in isolation, where they are easy to understand. The programs are clearly labeled in the text, and they can all be found in the software distribution that accompanies this book. Directions for downloading the software are given below.

There is also a major case study, that is progressively developed in Chapters 12 through 18. This case study illustrates many features of C# working together in combination, as they would in a practical application. A special point is made of demonstrating the object-oriented features of C#. If you are new to OO, studying the case study is a must!

The sample programs are provided in a self-extracting file. When expanded, a directory structure is created rooted in **c:\OI\CSharp**. The sample programs are in directories **Chap1**, **Chap2**, etc. All the samples for a given chapter are in individual folders within the chapter directories. The names of the folders are clearly identified in the text. Each chapter that contains a step of the case study has a folder **CaseStudy** containing that step.

This book is part of the Integrated .NET Series from Object Innovations and Prentice Hall PTR series. The sample programs for other books in the series are located in their own directory underneath **c:\OI**, so all the .NET examples from all books in the series will be located in a common area as you install them.

## EXERCISES

Although exercises are not provided in the book itself, a comprehensive set of exercises is available for download from our website.



## WEBSITE

The website for the Integrated .NET Series from Object Innovations and Prentice Hall PTR series is:

**[www.objectinnovations.com/dotnet.htm](http://www.objectinnovations.com/dotnet.htm)**

A link is provided at that website for downloading the sample programs.

## ACKNOWLEDGMENTS

I am indebted to Michael Stiefel and Mike Meehan for helping this project get off the ground. They met at the PDC when Microsoft announced .NET, and that conversation put into motion what has become a substantial series of books on .NET technology, of which this volume is the first book. Mike helped me to focus, and Michael has been a great collaborator on the series as a whole, especially on the second book that we are writing together. Another early collaborator was Charlie Ferebee of Microsoft. Although Charlie had to drop out of the project for business reasons, Charlie's enthusiasm for .NET was a tremendous early boost to the project. Howard Lee Harkness reviewed the entire manuscript and provided many helpful suggestions. My wife, Marianne, has provided enormous support and encouragement for all my writing efforts. This time, above and beyond the call of duty, she went through the whole book, proofreading and checking examples. Thank you all, and the other colleagues, friends and students who have helped me over the years and who are too numerous to mention individually.

## The Integrated .NET Series from Object Innovations and Prentice Hall PTR

### About this Series

Robert J. Oberg, Series Editor

### Introduction

The Integrated .NET Book Series from Object Innovations and Prentice Hall PTR is a unique series of introductory and intermediate books on Microsoft's important .NET technology. These books are based on proven industrial-strength course development experience. The authors are expert practitioners, teachers, and writers who combine subject-matter expertise with years of experience in presenting complex programming technologies such as C++, Java™, MFC, OLE, and COM/COM+. These books *teach* in a systematic, step-by-step manner and are not merely summaries of the documentation. All the books come with a rich set of programming examples, and a thematic case study is woven through several of the books.

From the beginning, these books have been conceived as an *integrated whole*, and not as independent efforts by a diverse group of authors. The initial set of books consists of three introductory books on .NET languages and four intermediate books on the .NET Framework. Each book in the series is targeted at a specific part of the important .NET technology, as illustrated by the diagram below.

		C# Learning Pathway	VB.NET Learning Pathway		
.NET Language Introductions	Programming PERL in the .NET Environment	Introduction to C# Using .NET	Introduction to Programming Visual Basic Using .NET		
Intermediate .NET Framework Titles		Application Development Using C# and .NET	Application Development Using Visual Basic and .NET	.NET Architecture and Programming Using Visual C++	Fundamentals of Web Applications Using .NET and XML

### Introductory .NET Language Books

The first set of books teaches several of the important .NET languages. These books cover each language from the ground up and have no prerequisite other than programming experience in some language. Unlike many NET language books, which are a mixture of the language and topics in the .NET Framework, these books are focused on the languages, with attention to important interactions between the language and the framework. By concentrating on the languages, these books have much more detail and many more practical examples than similar books.

The languages selected are the new language C#, the greatly changed VB.NET, and Perl.NET, the open source language ported to the .NET environment. Visual C++ .NET is covered in a targeted, intermediate book, and JScript.NET is covered in the intermediate level .NET Web-programming book.

#### *Introduction to C# Using .NET*

This book provides thorough coverage of the C# language from the ground up. It is organized with a specific section covering the parts of C# common to other C-like languages. This section can be cleanly skipped by programmers with C experience or the equivalent, making for a good reading path for a diverse group of readers. The book gives thorough attention to the object-oriented aspects of C# and thus serves as an excellent book for programmers migrating to C# from Visual Basic or COBOL. Its gradual pace and many examples make the book an excellent candidate as a college textbook for adventurous professors looking to teach C# early in the language's life-cycle.

#### *Introduction to Programming Visual Basic Using .NET*

Learn the VB.NET language from the ground up. Like the companion book on C#, this book gives thorough attention to the object-oriented aspects of VB.NET. Thus the book is excellent for VB programmers migrating to the more sophisticated VB.NET, as well as for programmers experienced in languages such as COBOL. This book would also be suitable as a college textbook.



## ***Programming Perl in the .NET Environment***

A very important part of the vision behind Microsoft® .NET is that the platform is designed from the ground up to support multiple programming languages from many sources, and not just Microsoft languages. This book, like other books in the series, is rooted in long experience in industrial teaching. It covers the Perl language from the ground up. Although oriented toward the ActiveState Perl.NET compiler, the book also provides excellent coverage of the Perl language suitable for other versions as well.

## ***Intermediate .NET Framework Books***

The second set of books is focused on topics in the .NET Framework, rather than on programming languages. Three parallel books cover the .NET Framework using the important languages C#, VB.NET, and Visual C++. The C# and VB.NET books include self-contained introductions to the languages suitable for experienced programmers, allowing them to rapidly come up to speed on these languages without having to plow through the introductory books. The fourth book covers the important topic of web programming in .NET, with substantial coverage of XML, which is so important in the .NET Framework.

The design of the series makes these intermediate books much more suitable to a wider audience than many similar books. The introductory books' focus on languages frees up the intermediate books to cover the important topics of the .NET Framework in greater depth. The series design also makes for flexible reading paths. Less experienced readers can read the introductory language books followed by the intermediate framework books, while more experienced readers can go directly to the intermediate framework books.

## ***Application Development Using C# and .NET***

This book does not require prior experience in C#. However, the reader should have experience in some object-oriented language such as C++ or Java™. The book could also be read by seasoned Visual Basic programmers who have experience working with objects and components in VB. Seasoned programmers and also a less experienced reader coming from the introductory C# book can skip the first few chapters on C# and proceed directly to a study of the Framework. The book is practical, with many examples and a major case study. The goal is to equip the reader with the knowledge necessary to begin building significant applications using the .NET Framework.

## ***Application Development Using Visual Basic .NET***

This book is for the experienced VB programmer who wishes to learn the new VB.NET version of VB quickly and then move on to learning the .NET Framework. It is also suitable for experienced enterprise programmers in other languages who wish to learn the powerful RAD-oriented Visual Basic language in its .NET incarnation and go on to building applications. Like the companion C# book, this book is very practical, with many examples, and includes the same case study implemented in VB.NET.

## ***.NET Architecture and Programming Using Visual C++***

This parallel book is for the experienced Visual C++ programmer who wishes to learn the .NET Framework to build high-performing applications. Unlike the C# and VB.NET book, there is no coverage of the C++ language itself, because C++ is too complex to cover in a brief space. This book is specifically for experienced C++ programmers. Like the companion C# and VB.NET books, this book is very practical, with many examples, and includes the same case study implemented in Visual C++.

## ***Fundamentals of Web Applications Using .NET and XML***

The final book in the series provides thorough coverage of building Web applications using .NET. Unlike other books about ASP.NET, this book gives attention to the whole process of Web application development. The book incorporates a review tutorial on classical Web programming, making the book accessible to the experienced programmer new to the Web world. The book contains significant coverage on ASP.NET, Web Forms, Web Services, SOAP, and XML.

# CONTENTS

<i>Preface</i>	<i>xix</i>
Organization	<i>xx</i>
Sample Programs	<i>xxi</i>
Exercises	<i>xxi</i>
Website	<i>xxii</i>
<i>Acknowledgments</i>	<i>xxii</i>
<i>About This Series</i>	<i>xxiii</i>

## PART ONE INTRODUCTION TO .NET AND C# 1

▼ CHAPTER 1 .NET Framework	5
.NET: What You Need to Know	5
A .NET Testbed for C# Programming	6
Installing the Book Software	7
Your First C# Program	8
Visual Studio.NET	8
Understanding .NET	8
What Is Microsoft .NET?	9
Microsoft and the Web	10
Windows on the Desktop	10
A New Programming Platform	11
.NET Framework Overview	11
Common Language Runtime	12
.NET Framework Class Library	12
Common Language Specification	13
Languages in .NET	13
Common Language Runtime	14
Design Goals of the CLR	14
Why Use a CLR?	16



Design of Common Language Runtime	16
Assemblies	18
Types	19
Managed Data and Garbage Collection	20
Assemblies and ILDASM	20
Intermediate Language	21
Summary	22
 ▼ CHAPTER 2 First C# Programs	 23
<i>Hello, World</i>	23
Compiling and Running (Command Line)	24
<i>Program Structure</i>	24
<i>Namespaces</i>	26
<i>Variables and Expressions</i>	26
Expressions	27
Assignment	27
<i>Using C# As A Calculator</i>	28
Sample Program	28
<i>Input/Output in C#</i>	28
Output in C#	29
Input in C#	29
<i>The .NET Framework Class Library</i>	33
Summary	33
 ▼ CHAPTER 3 Visual Studio.NET	 35
<i>Overview of Visual Studio.NET</i>	35
Toolbars	39
<i>Creating a Console Application</i>	41
Creating a C# Project	41
Adding a C# File	42
Using the Visual Studio Text Editor	42
Building the Project	43
Running the Program	43
Running the Program in the Debugger	44

<i>Project Configurations</i>	44
Creating a New Configuration	44
Setting Build Settings for a Configuration	46
<i>Debugging</i>	47
Just-in Time Debugging	47
Standard Debugging	48
<i>Summary</i>	52

## PART TWO C# AS A LANGUAGE IN THE C FAMILY 53

### ▼ CHAPTER 4 Simple Data Types 57

<i>Data Types</i>	57
Strong Typing	58
<i>Integer Types</i>	60
Ranges of Integer Types	61
Integer Literals	62
<i>Floating-Point Types</i>	63
Ranges of Floating Types	64
Floating-Point Literals	64
IEEE Floating-Point	65
<i>Decimal Type</i>	65
Range of Decimal Type	66
Decimal Literals	66
<i>Character Type</i>	67
Character Literals	67
<i>Boolean Type</i>	69
<i>Conversions</i>	69
<i>Summary</i>	71

### ▼ CHAPTER 5 Operators and Expressions 73

<i>Operator Cardinality</i>	73
-----------------------------	----

<i>Arithmetic Operators</i>	74
Multiplicative Operators	74
Additive Operators	77
A Practical Example	78
<i>Relational Operators</i>	79
<i>Conditional Logical Operators</i>	81
Short Circuit Evaluation	82
Ternary Conditional Operator	82
<i>Bitwise Operators</i>	83
Bitwise Logical Operators	84
Shift Operators	84
<i>Assignment Operators</i>	87
Compound Assignment	87
<i>Expressions</i>	88
Associativity	88
Precedence Table	88
<i>Checked and Unchecked</i>	89
<i>Summary</i>	91

## ▼ CHAPTER 6 Control Structures

93

<i>If Tests</i>	93
Single Branch If	95
Blocks	95
<i>Loops</i>	96
While Loop	97
For Loops	100
Do Loops	102
<i>Preview of Arrays and Foreach</i>	102
Foreach	104
<i>More About Control Flow</i>	105
break	105
continue	106
goto	106
Structured Programming	107

*Switch Statement* 108

*Summary* 110

## PART THREE C# PROGRAMMING FUNDAMENTALS 111

### ▼ CHAPTER 7 Object-Oriented Programming 115

*Objects* 115

Objects in the Real World 116

Object Models 116

Reusable Software Components 116

Objects in Software 117

State and Behavior 117

Abstraction 117

Encapsulation 118

*Classes* 118

Inheritance 119

Relationships Among Classes 119

*Polymorphism* 120

*Object-Oriented Languages* 120

Simula and C++ 121

Smalltalk 121

Java 122

Visual Basic 122

C# 122

*Components* 123

C# and Components 123

*Summary* 124

### ▼ CHAPTER 8 Classes 125

*Classes as Structured Data* 125

Classes and Objects 126

References 126

Instantiating and Using an Object 126



Assigning Object References	127
Garbage Collection	128
Sample Program	129
<i>Methods</i>	130
Public and Private	130
Abstraction	131
Encapsulation	131
<i>Constructors and Initialization</i>	132
Constructors	132
Default Constructor	133
this	134
Sample Program	135
<i>Static Fields and Methods</i>	136
Static Methods	136
Sample Program	137
Static Constructor	139
<i>Constant and Readonly Fields</i>	140
<i>Summary</i>	141
 ▼ CHAPTER 9 The C# Type System	 143
<i>Overview of Types in C#</i>	144
Value Types	144
Reference Types	144
Pointer Types	144
<i>Value Types</i>	144
Simple Types	144
Types in System Namespace	145
Structures	146
Enumeration Types	150
<i>Reference Types</i>	151
Class Types	151
Arrays	152
Interfaces	152
Delegates	152

<i>Default Values</i>	152
<i>Boxing and Unboxing</i>	153
<i>Summary</i>	154

## ▼ CHAPTER 10    Methods, Properties, and Operators    155

<i>Methods</i>	155
Classes with All Static Methods	156
<i>Parameter Passing</i>	157
Value Parameters	158
Reference Parameters	158
Output Parameters	159
Structure Parameters	160
Class Parameters	161
<i>Method Overloading</i>	162
<i>Variable Length Parameter Lists</i>	164
<i>Properties</i>	166
<i>Operator Overloading</i>	168
Sample Program	169
Operator Overloading in the Class Library	171
<i>Summary</i>	171

## ▼ CHAPTER 11    Characters and Strings    173

<i>Characters</i>	173
Sample Program	174
Character Codes	175
<i>Strings</i>	177
String Class	177
Compiler Support	178
String Equality	180
String Comparison	181
String Input	182
<i>String Methods</i>	183
int Length {get;}	183

`stringToUpper();` 183  
`stringToLower();` 183  
`public string Substring( int startIndex, int length);` 184  
`int IndexOf( string value);` 184

*String Builder Class* 184

*Programming with Strings* 185

Command Line Arguments 185

Command Loops 186

Splitting a String 187

*Summary* 187

## ▼ CHAPTER 12 Arrays and Indexers 189

*Arrays* 189

One-Dimensional Arrays 190

*System.Array* 190

Sample Program 191

Interfaces for System.Array 191

*Random Number Generation* 192

Constructors 192

Next Methods 192

NextDouble Method 192

*Jagged Arrays* 193

*Rectangular Arrays* 194

*Arrays as Collections* 195

*Bank Case Study: Step 1* 196

Account 197

Bank 198

TestBank 200

Atm 201

Running the Case Study 203

*Indexers* 203

*Summary* 205