Welcome to the Visual C#® 2012 computer programming language and the world of Microsoft® Windows® and Internet and web programming with Microsoft’s .NET platform. Please read the book’s back cover and inside back cover—these concisely capture the book’s essence. In this Preface we provide more details.

This book is appropriate for introductory course sequences based on the curriculum recommendations of two key professional organizations—the ACM and the IEEE. The examples are accessible to computer science, information technology, software engineering and business students in novice-level and intermediate-level C# courses. The book can also be used by professional programmers.

At the heart of the book is the Deitel signature live-code approach—rather than using code snippets, we present concepts in the context of complete working programs followed by sample executions. Read the Before You Begin section after this Preface for instructions on setting up your computer to run the hundreds of code examples. The source code is available at www.deitel.com/books/vcsharp2012htp and www.pearsonhighered.com/deitel. Use the source code we provide to compile and run each program as you study it—this will help you master Visual C# and related Microsoft technologies faster and at a deeper level.

We believe that this book and its supplements for students and instructors will give you an informative, engaging, challenging and entertaining introduction to Visual C#. If you have questions, we’re easy to reach at deitel@deitel.com—we’ll respond promptly. For book updates, visit www.deitel.com/books/vcsharp2012htp, join our social media communities on Facebook (www.deitel.com/DeitelFan), Twitter (@deitel), Google+ (googleplus.to/deitel) and LinkedIn (bit.ly/DeitelLinkedIn), and subscribe to the Deitel® Buzz Online newsletter (www.deitel.com/newsletter/subscribe.html).

**Visual C#® 2012, the Visual Studio® 2012 IDE, .NET 4.5, Windows® 7 and Windows® 8**

The new Visual C# 2012 and its associated technologies motivated us to write *Visual C# 2012 How to Program, 5/e*. These are some of the key features of this new edition:

- **Use with Windows 7, Windows 8 or both.** The book is designed so that you can continue to use Windows 7 now and begin to evolve to Windows 8, if you like, or you can move right to Windows 8. All of the code examples in Chapters 1–24 and 28–35 were tested on both Windows 7 and Windows 8. The code examples for the Windows-8-specific chapters—Chapter 25 (Windows 8 UI and XAML), Chapter 26 (Windows 8 Graphics and Multimedia) and Chapter 27 (Building a Windows Phone 8 App)—were tested only on Windows 8.

- **C# and Visual C#**. The C# language has been standardized internationally by ECMA and ISO (the standards document is available free of charge at bit.ly/ECMA334). Visual C# 2012 is Microsoft’s implementation of this standard.
Preface

- **Modular multi-GUI treatment with Windows Forms, Windows 8 UI and WPF.** The printed book features Windows Forms GUI; optional online chapters contain treatments of Windows 8 UI (user interface) and WPF GUI. Windows 8 UI apps are called Windows Store apps. In Chapter 25, you'll learn how to create and test Windows Store apps and upload them to Microsoft's Windows Store.

- **Modular treatment of graphics and multimedia with Windows 8 and WPF.** The book features optional online chapters on both Windows 8 Graphics and Multimedia (Chapter 26) and WPF Graphics and Multimedia (Chapter 33).

- **Database with LINQ to Entities.** In the previous edition of this book, we discussed LINQ (Language Integrated Query) to SQL (Microsoft's SQL Server database system). Microsoft stopped further development on LINQ to SQL in 2008 in favor of the newer and more robust LINQ to Entities and the ADO.NET Entity Framework, which we've switched to in this edition, keeping the discussion friendly for novices.

- **SQL Server database.** We use Microsoft's free SQL Server Express 2012 (which installs with the free Visual Studio Express 2012 for Windows Desktop) to present the fundamentals of database programming. Chapters 22–23 and 29–30 use database and LINQ capabilities to build an address-book desktop app, a web-based guestbook app, a bookstore app and an airline reservation system app.

- **ASP.NET 4.5.** Microsoft’s .NET server-side technology, ASP.NET, enables you to create robust, scalable web-based apps. In Chapter 23, you’ll build several apps, including a web-based guestbook that uses ASP.NET and the ADO .NET Entity Framework to store data in a database and display data in a web page. The chapter also discusses the IIS Express web server for testing your web apps on your local computer.

- **Building a Windows Phone 8 App.** Windows Phone 8 is Microsoft’s latest operating system for smartphones. It features multi-touch support for touchpads and touchscreen devices, enhanced security features and more. In Chapter 27, you’ll build a complete working Windows Phone 8 app and test it on the Windows Phone simulator; we’ll discuss how to upload apps to the Windows Phone Store.

- **Building a Windows Azure™ Cloud Computing App.** Windows Azure is a cloud computing platform that allows you to develop, manage and distribute your apps in the cloud. Chapter 31 shows you how to build a Windows Azure app that can store data in the cloud.

- **Asynchronous programming with **`async`** and **`await`**.** Asynchronous programming is simplified in Visual C# 2012 with the new **`async`** and **`await`** capabilities. We introduce asynchronous programming with **`async`** and **`await`** in Chapter 28.

### Object-Oriented Programming

- **Early-objects approach.** The book introduces the basic concepts and terminology of object technology in Chapter 1. In Chapter 2, Dive Into Visual Studio 2012 Express for Windows Desktop, you’ll **visually** manipulate objects, such as labels and images. In Chapter 3, Introduction to C# Apps, you’ll write Visual C# **program code**
that manipulates preexisting objects. You’ll develop your first customized classes and objects in Chapter 4. Presenting objects and classes early gets you “thinking about objects” immediately and mastering these concepts more thoroughly.

- **Rich coverage of programming fundamentals.** Chapters 5 and 6 present a friendly treatment of control statements and problem solving.
- **A clear, example-driven presentation of classes, objects, inheritance, polymorphism and interfaces.**
- **Optional case study: Using the UML to develop an object-oriented design and Visual C# implementation of an Automated Teller Machine (ATM).** The UML™ (Unified Modeling Language™) is the industry-standard graphical language for modeling object-oriented systems. We introduce the UML in the early chapters. Online Chapters 34 and 35 include an optional case study on object-oriented design using the UML. We design and implement the software for a simple automated teller machine. We analyze a typical requirements document that specifies the system to be built. We determine the classes needed to implement that system, the attributes the classes need to have, the behaviors the classes need to exhibit and we specify how the classes must interact with one another to meet the system requirements. From the design we produce a complete working Visual C# implementation. Students often report a “light bulb moment”—the case study helps them “tie it all together” and truly understand object orientation.

- **Three programming paradigms.** We discuss structured programming, object-oriented programming and generic programming.

**Complete Code Examples**

We include a broad range of example programs selected from computer science, business, simulation, game playing, graphics, multimedia and many other areas (Fig. 1).

### Examples

<table>
<thead>
<tr>
<th>Account class</th>
<th>Card shuffling and dealing</th>
<th>Credit-inquiry program</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address book case study</td>
<td>CheckedListBox control</td>
<td>Data binding</td>
</tr>
<tr>
<td>Airline reservation web-service</td>
<td>ComboBox control</td>
<td>Date class</td>
</tr>
<tr>
<td>Animating the width and height of a video</td>
<td>CommissionEmployee class</td>
<td>DateTimePicker control</td>
</tr>
<tr>
<td>Applying transforms to a polygon</td>
<td>Common Windows 8 UI controls</td>
<td>Defining gradients in XAML</td>
</tr>
<tr>
<td>Array initializer</td>
<td>Common WPF controls</td>
<td>Dice rolling</td>
</tr>
<tr>
<td>ArrayList class</td>
<td>Compound interest calculations</td>
<td>Directory class</td>
</tr>
<tr>
<td>BasePlusCommissionEmployee class</td>
<td>Counter-controlled repetition</td>
<td>Document navigation using XNode</td>
</tr>
<tr>
<td>Binary search</td>
<td>Craps dice game simulation</td>
<td>Drawing basic shapes</td>
</tr>
<tr>
<td>Blackjack game web-service</td>
<td>Creating and using a text file</td>
<td>Drawing polylines and polygons</td>
</tr>
<tr>
<td>Books database</td>
<td>Creating custom windows and using timers</td>
<td>Employee class</td>
</tr>
<tr>
<td></td>
<td></td>
<td>File class</td>
</tr>
</tbody>
</table>

**Fig. 1** | A small sample of the book’s hundreds of examples. (Part 1 of 2.)
**Interesting, Entertaining and Challenging Exercises**

- Extensive self-review exercises *and* answers are included for self-study.
- Each chapter concludes with a substantial set of exercises, which generally includes simple recall of important terminology and concepts, identifying the errors in code samples, writing individual program statements, writing small portions of Visual C# classes, writing complete programs and implementing major projects. Figure 2 lists a small sampling of the book’s hundreds of exercises, including selections from our *Making a Difference* exercises set, which encourage you to use computers and the Internet to research and solve significant social problems—we hope you’ll approach these exercises with your own values, politics and beliefs.

**Fig. 1** | A small sample of the book’s hundreds of examples. (Part 2 of 2.)

**Fig. 2** | A sampling of the book’s exercises. (Part 1 of 2.)
Illustrations and Figures

Abundant tables, line drawings, UML diagrams, programs and program outputs are included. A sampling of these is shown in Figs. 3 and 4.

Fig. 3 | A sampling of the book’s tables, drawings and diagrams. (Part 1 of 2.)

Fig. 2 | A sampling of the book’s exercises. (Part 2 of 2.)

Illustrations and Figures

Main text tables, drawings and diagrams

<table>
<thead>
<tr>
<th>Anchoring demonstration</th>
<th>Escape sequences</th>
<th>LINQ to XML class hierarchy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ajax-enabled web app</td>
<td>GroupBox properties</td>
<td>Master/Detail app</td>
</tr>
<tr>
<td>Binary tree graphical representation</td>
<td>HttpSessionState properties</td>
<td>Mouse class methods</td>
</tr>
<tr>
<td>Circular, doubly linked list</td>
<td>Implicit conversions between simple types</td>
<td>Mouse events and event arguments</td>
</tr>
<tr>
<td>Circular, singly linked list</td>
<td>Increment and decrement operators</td>
<td>.NET Framework Class Library namespaces</td>
</tr>
<tr>
<td>Client receiving a response from a web server</td>
<td>insertAtBack operation represented graphically</td>
<td>Number of comparisons for common Big O notations</td>
</tr>
<tr>
<td>Client requesting a response from a web server</td>
<td>insertAtFront operation represented graphically</td>
<td>Object methods inherited by all classes</td>
</tr>
<tr>
<td>Creating a web service</td>
<td>Interaction between a web-service client and a SOAP web service</td>
<td>Polymorphic interface for the Employee hierarchy classes</td>
</tr>
<tr>
<td>Collection classes of the .NET Framework</td>
<td>Interfaces of the .NET Framework Class Library</td>
<td>Precedence of arithmetic operators</td>
</tr>
<tr>
<td>Common built-in commands from the WPF command library</td>
<td>Keyboard events and event arguments</td>
<td>removeFromBack operation represented graphically</td>
</tr>
<tr>
<td>Components and controls</td>
<td>Keywords and contextual keywords</td>
<td>removeFromFront operation represented graphically</td>
</tr>
<tr>
<td>Custom-control creation</td>
<td>Linked list graphical representation</td>
<td>Rules of forming structured apps</td>
</tr>
<tr>
<td>DatePicker properties/event</td>
<td></td>
<td>SDI and MDI forms</td>
</tr>
<tr>
<td>Doubly linked list</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Preface

Other Features

- **We use LINQ to query files, databases, XML and collections.** The introductory LINQ to Objects chapter (Chapter 9), is intentionally simple and brief to encourage instructors to begin covering LINQ technology early. Later in the book, we take a deeper look, using LINQ to Entities (Chapters 22–23 and 29–30) and LINQ to XML (Chapters 24, 30 and 31).

- **Local type inference.** When you initialize a local variable in its declaration, you can omit the variable’s type—the compiler *infers* it from the initializer value.

- **Object initializers.** For new objects, you can use object initializer syntax (similar to array initializer syntax) to assign values to the new object’s public properties and public instance variables.

- **We emphasize the IDE’s IntelliSense feature** that helps you write code faster and with fewer errors.

- **Files and strings.**

- **Data structures chapter sequence, including searching and sorting, data structures, generics and collections.**
- **Integrated exception handling.** We introduce exception handling early (Chapter 8, Arrays; Introduction to Exception Handling) to ensure that we do not access an array element outside the array’s bounds. Chapter 10, Classes and Objects: A Deeper Look, shows how to indicate an exception when a member function receives an invalid argument. We cover the complete details of exception handling in Chapter 13, Exception Handling: A Deeper Look.

- **Visual C# XML capabilities.** Extensible Markup Language (XML) is pervasive in the software-development industry, e-business and throughout the .NET platform. In optional online Chapter 24, we introduce XML syntax and programmatically manipulate the elements of an XML document using LINQ to XML. XAML is an XML vocabulary that’s used to describe graphical user interfaces, graphics and multimedia. We discuss XAML in optional online Chapters 25–26 and 32–33.

- **Web app development with ASP.NET 4.5 and ASP.NET AJAX.** Optional online Chapter 29 extends Chapter 23’s ASP.NET discussion with a case study on building a password-protected, web-based bookstore app. Also, we introduce in Chapter 29 ASP.NET AJAX controls and use them to add AJAX functionality to web apps to give them a look and feel similar to that of desktop apps.

- **WCF (Windows Communication Foundation) web services.** Web services enable you to package app functionality in a manner that turns the web into a library of reusable services. Optional online Chapter 30 includes case studies on building an airline reservation web service, a blackjack web service and a math question generator web service that’s called by a math tutor app.

- **WPF (Windows Presentation Foundation) GUI, graphics and multimedia.** We extend the core book’s GUI coverage in optional online Chapters 32–33 with an introduction to Windows Presentation Foundation (WPF)—a XAML-based Microsoft framework that preceded Windows 8 UI and integrates GUI, graphics and multimedia capabilities. We implement a painting app, a text editor, a color chooser, a book-cover viewer, a television video player, various animations, and speech synthesis and recognition apps.

**Companion Website**

The printed book contains the core content (Chapters 1–23) for introductory course sequences. Several optional online chapters are available for advanced courses and professionals. Figure 5 lists the chapters and appendices that are available in searchable PDF format on the book’s password-protected Companion Website at:

www.pearsonhighered.com/deitel

See the inside front cover of the book for an access code.

**Online chapters**

Chapter 24, XML and LINQ to XML  
Chapter 25, Windows 8 UI and XAML
Preface

Online chapters

Chapter 26, Windows 8 Graphics and Multimedia
Chapter 27, Building a Windows Phone 8 App
Chapter 28, Introduction to Concurrency: async and await
Chapter 29, Web App Development with ASP.NET: A Deeper Look
Chapter 30, Web Services
Chapter 31, Building a Windows Azure™ Cloud Computing App
Chapter 32, GUI with Windows Presentation Foundation
Chapter 33, WPF Graphics and Multimedia
Chapter 34, ATM Case Study, Part 1: Object-Oriented Design with the UML
Chapter 35, ATM Case Study, Part 2: Implementing an Object-Oriented Design
Appendix D, Number Systems
Appendix E, UML 2: Additional Diagram Types
Appendix F, Unicode®
Appendix G, Using the Visual Studio 2012 Debugger
Index (The online index includes the content from the printed book and the online content. The printed book index covers only the printed material.)

VideoNotes

The Companion Website also includes extensive VideoNotes—watch and listen as co-author Paul Deitel discusses key code examples in the core chapters of the book. VideoNotes allow for self-paced instruction with easy navigation, including the ability to select, play, rewind, fast-forward and stop within each video.

We’ve created a jump table that maps each VideoNote to the corresponding figures in the book (www.deitel.com/books/vcsharphtp5/jump_table.pdf). VideoNotes are free with the purchase of a new textbook. If you have a used book you can purchase access to the VideoNotes for this book as follows:

1. Go to www.pearsonhighered.com/deitel/.
2. Scroll to Visual C# 2012 How to Program, 5/e and click Companion Website.
3. Click the Register button.
4. On the registration page, enter your student access code found beneath the scratch-off panel on the inside front cover of this book. Do not type the dashes. You can use lower- or uppercase. The access code can be used only once. This subscription is valid for twelve months upon activation and is not transferable. If this access code on your book has already been revealed, it may no longer be valid. If this is the case, click the Website Purchase link and follow the instructions.
5. Once your personal Login Name and Password are confirmed, you can begin using the Visual C# 2012 How to Program, 5/e Companion Website.
Book Overview and Chapter Dependencies

This section discusses the book’s modular organization to help instructors plan their syllabi.

Introduction to Visual C# and Visual Studio 2012 Express

Chapter 1, Introduction to Computers, the Internet and Visual C#, introduces computing fundamentals and Microsoft’s .NET platform. If you do not need to cover these fundamentals, you should still cover the Painter app test-drive. The vast majority of the book’s examples will run on Windows 7 and Windows 8 using Visual Studio Express 2012 for Windows Desktop, which we test-drive in Section 1.14. Chapters 25–26 can be run only on Windows 8 using Visual Studio Express 2012 for Windows 8, which we test-drive in Section 1.15. There are other versions of Visual Studio Express 2012 for web development and Windows Phone development—we cover these in the corresponding chapters.

Chapter 2, Dive Into® Visual Studio Express 2012 for Windows Desktop, shows how to develop a simple GUI app that displays text and an image. We’ll look at Visual Studio Express 2012 for Windows 8 in more depth in Chapter 25.

Introduction to Visual C# Fundamentals and Object-Oriented Programming

The chapters in this module of the book:

- Chapter 3, Introduction to C# Apps
- Chapter 4, Introduction to Classes, Objects, Methods and strings
- Chapter 5, Control Statements: Part 1
- Chapter 6, Control Statements: Part 2
- Chapter 7, Methods: A Deeper Look
- Chapter 8, Arrays; Introduction to Exception Handling

present C# programming fundamentals (data types, operators, control statements, methods and arrays) and introduce object-oriented programming. These chapters should be covered in order. Chapter 8 introduces exception handling with an example that demonstrates accessing an element outside an array’s bounds.

Object-Oriented Programming: A Deeper Look

The chapters in this module of the book:

- Chapter 9, Introduction to LINQ and the List Collection
- Chapter 10, Classes and Objects: A Deeper Look
- Chapter 11, Object-Oriented Programming: Inheritance
- Chapter 12, OOP: Polymorphism, Interfaces and Operator Overloading
- Chapter 13, Exception Handling: A Deeper Look
- Chapter 34, ATM Case Study, Part 1: Object-Oriented Design with the UML
- Chapter 35, ATM Case Study, Part 2: Implementing an Object-Oriented Design

provide a deeper look at object-oriented programming, including classes, objects, inheritance, polymorphism, interfaces and exception handling. Chapter 9, Introduction to LINQ and the List Collection, introduces Microsoft’s Language Integrated Query (LINQ) technology, which provides a uniform syntax for manipulating data from various data sources,
Preface

such as arrays, collections and, as you’ll see in later chapters, XML and databases. This chapter can be deferred, but it’s required for one example in Chapter 17 (Fig. 17.6) and many of the later chapters starting with Chapter 22, Databases and LINQ. Online Chapters 34–35 present an optional object-oriented design and implementation case study that requires the C# and object-oriented programming concepts presented in Chapters 3–8 and 10–13.

Windows Forms Graphical User Interfaces (GUIs)

There are now three GUI technologies in Windows—Windows Forms (which is a legacy technology), Windows 8 UI (available only on Windows 8) and Windows Presentation Foundation (WPF). We surveyed instructors teaching Visual C# and they still prefer Windows Forms for their classes, so we provide a two-chapter introduction to Windows Forms:

- Chapter 14, Graphical User Interfaces with Windows Forms: Part 1
- Chapter 15, Graphical User Interfaces with Windows Forms: Part 2

in the print book, then use Windows Forms GUIs in several other print and online chapters. Most examples in Chapters 14–15 can be presented after Chapter 4. For those who wish to present or study Microsoft’s more recent GUI, graphics and multimedia technologies, we provide two-chapter online introductions to Windows 8 UI, graphics and multimedia (Chapters 25–26) and WPF GUI, graphics and multimedia (Chapters 32–33).

Strings and Files

We introduce strings beginning in Chapter 4 and use them throughout the book. Chapter 16, Strings and Characters: A Deeper Look, investigates strings in more depth. Chapter 17, Files and Streams, introduces text-file processing and object-serialization for input/output of entire objects. Chapter 16 can be presented at any point after Chapter 4. Chapter 17 requires C#, object-oriented programming and Windows Forms concepts presented in Chapters 3–14.

Searching, Sorting and Data Structures

The chapters in this module of the book:

- Chapter 18, Searching and Sorting
- Chapter 19, Data Structures
- Chapter 20, Generics
- Chapter 21, Collections

introduce searching, sorting and data structures. Most C# programmers should use .NET’s built-in searching, sorting and collections (prepackaged data structures) capabilities, which are discussed in Chapter 21. For instructors who wish to present how to implement customized searching, sorting and data structures capabilities, we provide Chapters 18–20, which require the concepts presented in Chapters 3–8 and 10–13.

Databases and an Introduction to Web App Development

Chapter 22, Databases and LINQ, introduces database app development using the ADO.NET Entity Framework and LINQ to Entities. The chapter’s examples require C#, object-oriented programming and Windows Forms concepts presented in Chapters 3–14. Chapter 23, Web App Development with ASP.NET, introduces web app development.
The last example in this chapter requires the LINQ and database techniques presented in Chapter 22.

**Extensible Markup Language (XML)**
Chapter 24, XML and LINQ to XML, introduces XML, which is used in several later chapters. The first few sections of this chapter are required to understand the XAML markup that's used to build Windows 8 GUI, graphics and multimedia apps (Chapters 25–26), Windows Phone 8 apps (Chapter 27) and WPF GUI, graphics and multimedia apps (Chapters 32–33). The remainder of the chapter discusses LINQ to XML, which allows you to manipulate XML using LINQ syntax. These capabilities are used in Chapters 30 and 31.

**Windows 8 UI, Graphics and Multimedia; Windows Phone**
The chapters in this module of the book:
- Chapter 25, Windows 8 UI and XAML
- Chapter 26, Windows 8 Graphics and Multimedia
- Chapter 27, Building a Windows Phone 8 App

present Windows 8 UI, graphics and multimedia, and Windows Phone 8 app development. These chapters can be used only on computers running Windows 8 and depend on event-handling concepts that are presented in Chapter 14, and the introduction to XML at the beginning of Chapter 24 (see Section 24.1 for details). Developing a Windows Phone 8 app is similar to developing a Windows 8 UI app.

**Asynchronous Programming**
Chapter 28, Asynchronous Programming with async and await, demonstrates .NET's and Visual C#'s new simplified asynchronous programming capabilities. These are commonly used in Web app and Web service development (among many other uses).

**Web App Development and Web Services**
The chapters in this module of the book:
- Chapter 29, Web App Development with ASP.NET: A Deeper Look
- Chapter 30, Web Services
- Chapter 31, Building a Windows Azure™ Cloud Computing App

continue our discussion of Web app development from Chapter 23 and introduce web services, including a case study on cloud computing with Windows Azure. Chapters 30 and 31 depend on the LINQ to XML discussion in Chapter 24.

**Windows Presentation Foundation (WPF) GUI, Graphics and Multimedia**
The chapters in this module of the book
- Chapter 32, GUI with Windows Presentation Foundation
- Chapter 33, WPF Graphics and Multimedia

discuss Windows Presentation Foundation GUI, graphics and multimedia. These chapters can be used on computers running Windows 7 or Windows 8 and depend on event-handling concepts that are presented in Chapter 14 and the introduction to XML at the beginning of Chapter 24.
Preface

Teaching Approach

*Visual C# 2012 How to Program, 5/e* contains a rich collection of examples. We concentrate on building good software and stress program clarity.

**Live-Code Approach.** The book is loaded with “live-code” examples. Most new concepts are presented in the context of complete working Visual C# apps, followed by one or more executions showing program inputs and outputs. In the few cases where we show a code snippet, to ensure correctness we first tested it in a complete working program then copied the code from the program and pasted it into the book.

**Syntax Shading.** For readability, we syntax shade the code, similar to the way most integrated-development environments and code editors syntax color the code. Our syntax-shading conventions are:

- **comments** appear like this
- **keywords** appear like this
- **constants and literal values** appear like this
- **all other code** appears in black

**Code Highlighting.** We place gray rectangles around each program’s key code segments.

**Using Fonts for Emphasis.** We place the key terms and the index’s page reference for each defining occurrence in **bold** text for easy reference. We show on-screen components in the **bold Helvetica** font (for example, the **File** menu) and Visual C# program text in the Lucida font (for example, `int count = 5`). We use **italics** for emphasis.

**Objectives.** The chapter objectives preview the topics covered in the chapter.

**Programming Tips.** We include programming tips to help you focus on important aspects of program development. These tips and practices represent the best we’ve gleaned from a combined seven decades of programming and teaching experience.

---

**Good Programming Practice**

_The Good Programming Practices call attention to techniques that will help you produce programs that are clearer, more understandable and more maintainable._

**Common Programming Error**

_Pointing out these Common Programming Errors reduces the likelihood that you’ll make them._

**Error-Prevention Tip**

_These tips contain suggestions for exposing and removing bugs from your programs; many of the tips describe aspects of Visual C# that prevent bugs from getting into programs._

**Performance Tip**

_These tips highlight opportunities for making your programs run faster or minimizing the amount of memory that they occupy._

**Portability Tip**

_The Portability Tips help you write code that will run on a variety of platforms._
Obtaining the Software Used in Visual C# How to Program, S/e

Software Engineering Observation

The Software Engineering Observations highlight architectural and design issues that affect the construction of software systems, especially large-scale systems.

Look-and-Feel Observation

These observations help you design attractive, user-friendly graphical user interfaces that conform to industry norms.

Summary Bullets. We present a detailed bullet-list summary of each chapter.

Terminology. We include a list of the important terms defined in each chapter.

Index. We’ve included an extensive index for reference. Defining occurrences of key terms in the index are highlighted with a bold page number.

Obtaining the Software Used in Visual C# How to Program, S/e

We wrote the code examples in Visual C# 2012 How to Program, S/e using Microsoft’s free Visual Studio Express 2012 products, including:

- Visual Studio Express 2012 for Windows Desktop (Chapters 1–24, 28 and 32–35), which includes Visual C# and other Microsoft development tools. This runs on Windows 7 and 8.
- Visual Studio Express 2012 for Web (Chapters 23 and 29–31)
- Visual Studio Express 2012 for Windows 8 (Chapters 25–26)
- Visual Studio Express 2012 for Windows Phone (Chapter 27)

Each of these is available for download at

www.microsoft.com/visualstudio/eng/products/
visual-studio-express-products

Instructor Supplements

The following supplements are available to qualified instructors only through Pearson Education’s Instructor Resource Center (www.pearsonhighered.com/irc):

- Solutions Manual contains solutions to most of the end-of-chapter exercises. We’ve added many Making a Difference exercises, most with solutions. Please do not write to us requesting access to the Pearson Instructor’s Resource Center. Access is restricted to college instructors teaching from the book. Instructors may obtain access only through their Pearson representatives. If you’re not a registered faculty member, contact your Pearson representative or visit www.pearsonhighered.com/educator/replocator/. Exercise Solutions are not provided for “project” exercises. Check out our Programming Projects Resource Center for lots of additional exercise and project possibilities:

www.deitel.com/ProgrammingProjects

- Test Item File of multiple-choice questions (approximately two per book section)
- Customizable PowerPoint® slides containing all the code and figures in the text, plus bulleted items that summarize the key points in the text.
Preface

Microsoft DreamSpark™

Professional Developer and Designer Tools for Students
Microsoft provides many of its professional developer tools to students for free via a program called DreamSpark (www.dreamspark.com). See the website for details on verifying your student status so you take advantage of this program.

Acknowledgments

We’d like to thank Abbey Deitel and Barbara Deitel of Deitel & Associates, Inc. for long hours devoted to this project. Abbey co-authored this Preface and Chapter 1 and she and Barbara painstakingly researched the new capabilities of Visual C# 2012, .NET 4.5, Windows 8, Windows Phone 8, Windows Azure and other key topics.

We’re fortunate to have worked with the dedicated team of publishing professionals at Pearson Higher Education. We appreciate the guidance, wisdom and energy of Tracy Johnson, Executive Editor, Computer Science. Carole Snyder did an extraordinary job recruiting the book’s reviewers and managing the review process. Bob Engelhardt did a wonderful job bringing the book to publication.

Reviewers

We wish to acknowledge the efforts of our reviewers. The book was scrutinized by academics teaching C# courses and industry experts. They provided countless suggestions for improving the presentation. Any remaining flaws in the book are our own.

Fifth Edition Reviewers: Shay Friedman (Microsoft Visual C# MVP), Octavio Hernandez (Microsoft Certified Solutions Developer), Stephen Hustedde (South Mountain College), José Antonio González Seco (Parliament of Andalusia, Spain) and Shawn Weisfeld (Microsoft MVP and President and Founder of UserGroup.tv).

Other recent edition reviewers: Huanhui Hu (Microsoft Corporation), Narges Kasiri (Oklahoma State University), Charles Liu (University of Texas at San Antonio), Dr. Hamid R. Nemati (The University of North Carolina at Greensboro), Jeffrey P. Scott (Blackhawk Technical College), Douglas B. Bock (MCSD.NET, Southern Illinois University Edwardsville), Dan Crevier (Microsoft), Amit K. Ghosh (University of Texas at El Paso), Marcelo Guerra Hahn (Microsoft), Kim Hamilton (Software Design Engineer at Microsoft and co-author of Learning UML 2.0), James Edward Keysor (Florida Institute of Technology), Helena Kotas (Microsoft), Chris Lovett (Software Architect at Microsoft), Bashar Lulu (INETA Country Leader, Arabian Gulf), John McIlhinney (Spatial Intelligence; Microsoft MVP 2008 Visual Developer, Visual Basic), Ged Mead (Microsoft Visual Basic MVP, DevCity.net), Anand Mukundan (Architect, Polaris Software Lab Ltd.), Timothy Ng (Microsoft), Akira Onishi (Microsoft), Joe Stagner (Senior Program Manager, Developer Tools & Platforms), Erick Thompson (Microsoft), Jesús Ubaldo Quevedo-Torres (University of Wisconsin–Parkside, Department of Computer Science) and Zijiang Yang (Western Michigan University).

As you read the book, we’d sincerely appreciate your comments, criticisms and suggestions for improving the text. Please address all correspondence to:

deitel@deitel.com
We’ll respond promptly. We really enjoyed writing this book—we hope you enjoy reading it!

Paul Deitel
Harvey Deitel

About the Authors

Paul Deitel, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of MIT, where he studied Information Technology. Through Deitel & Associates, Inc., he has delivered hundreds of programming courses to industry clients, including Cisco, IBM, Siemens, Sun Microsystems, Dell, Fidelity, NASA at the Kennedy Space Center, the National Severe Storm Laboratory, White Sands Missile Range, Rogue Wave Software, Boeing, SunGard Higher Education, Nortel Networks, Puma, iRobot, Invensys and many more. He and his co-author, Dr. Harvey M. Deitel, are the world’s best-selling programming-language textbook/professional book/video authors.

Paul was named as a Microsoft® Most Valuable Professional (MVP) for C# in 2012. According to Microsoft, “the Microsoft MVP Award is an annual award that recognizes exceptional technology community leaders worldwide who actively share their high quality, real world expertise with users and Microsoft.”

Dr. Harvey Deitel, Chairman and Chief Strategy Officer of Deitel & Associates, Inc., has 50 years of experience in the computer field. Dr. Deitel earned B.S. and M.S. degrees in Electrical Engineering from MIT and a Ph.D. in Mathematics from Boston University. He has extensive college teaching experience, including earning tenure and serving as the Chairman of the Computer Science Department at Boston College before founding Deitel & Associates, Inc., in 1991 with his son, Paul Deitel. The Deitels’ publications have earned international recognition, with translations published in Chinese, Korean, Japanese, German, Russian, Spanish, French, Polish, Italian, Portuguese, Greek, Urdu and Turkish. Dr. Deitel has delivered hundreds of programming courses to corporate, academic, government and military clients.

Deitel® Dive-Into® Series Programming Languages Training

Deitel & Associates, Inc., founded by Paul Deitel and Harvey Deitel, is an internationally recognized authoring and corporate training organization, specializing in computer programming languages, object technology, mobile app development and Internet and web software technology. The company’s training clients include many of the world’s largest companies, government agencies, branches of the military, and academic institutions. The company offers instructor-led training courses delivered at client sites worldwide on major programming languages and platforms, including Visual C#®, Visual Basic®, Visual C++®, C++, C, Java™, XML®, Python®, object technology, Internet and web programming, Android app development, Objective-C and iPhone app development and a growing list of additional programming and software development courses.
Preface

Through its 37-year publishing partnership with Prentice Hall/Pearson, Deitel & Associates, Inc., publishes leading-edge programming college textbooks, professional books and LiveLessons video courses. Deitel & Associates, Inc. and the authors can be reached at:

deitel@deitel.com

To learn more about Deitel's Dive-Into® Series Corporate Training curriculum, visit:

www.deitel.com/training

To request a proposal for worldwide on-site, instructor-led training at your organization, e-mail deitel@deitel.com.

Individuals wishing to purchase Deitel books and LiveLessons video training can do so through www.deitel.com. Bulk orders by corporations, the government, the military and academic institutions should be placed directly with Pearson. For more information, visit

www.informit.com/store/sales.aspx