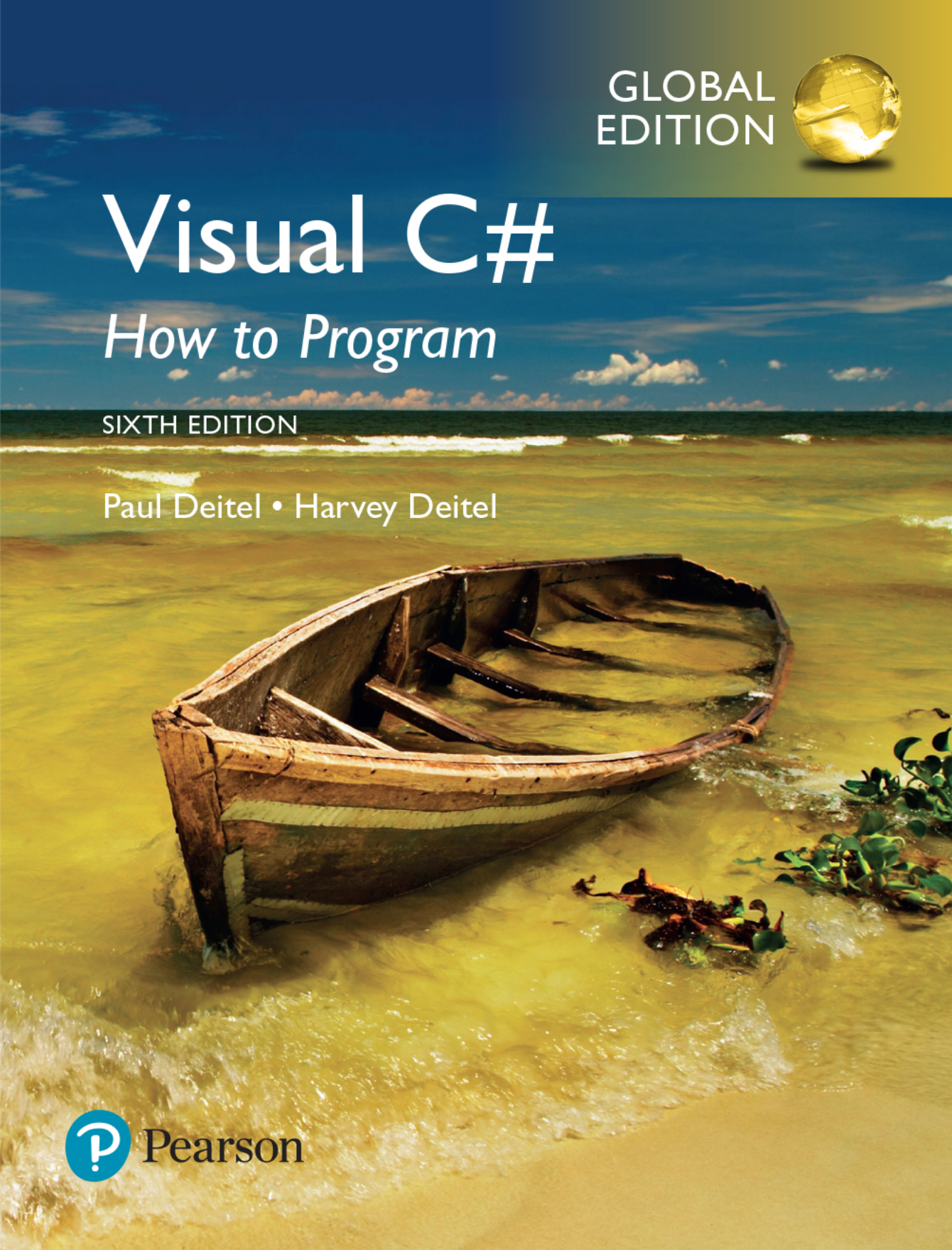
GLOBAL EDITION

Visual C#

How to Program

SIXTH EDITION

Paul Deitel • Harvey Deitel



Pearson

Visual C#®

HOW TO PROGRAM

This page intentionally left blank

Visual C#®

HOW TO PROGRAM

SIXTH EDITION
GLOBAL EDITION

Paul Deitel
Harvey Deitel
Deitel & Associates, Inc.

Boston Columbus Hoboken Indianapolis New York San Francisco
Amsterdam Cape Town Dubai London Madrid Milan Munich Paris Montreal
Toronto Delhi Mexico City São Paulo Sydney Hong Kong Seoul Singapore Taipei Tokyo

Vice President, Editorial Director: ***Marcia Horton***
Acquisitions Editor: ***Tracy Johnson***
Editorial Assistant: ***Kristy Alaura***
Acquisitions Editor, Global Editions: ***Sourabh Maheshwari***
VP of Marketing: ***Christy Lesko***
Field Marketing Manager: ***Demetrius Hall***
Marketing Assistant: ***Jon Bryant***
Director of Product Management: ***Erin Gregg***
Team Lead, Program and Project Management: ***Scott Disanno***
Program Manager: ***Carole Snyder***
Project Manager: ***Robert Engelhardt***
Project Editor, Global Editions: ***K.K. Neelakantan***
Manufacturing Buyer, Higher Ed | RR Donnelley: ***Maura Zaldivar-Garcia***
Senior Manufacturing Controller, Global Editions: ***Trudy Kimber***
Media Production Manager, Global Editions: ***Vikram Kumar***
Cover Design: ***Lumina Datamatics***
R&P Manager: ***Ben Ferrini***
Inventory Manager: ***Ann Lam***
Cover Photo Credit: © ***Pichugin Dmitry/Shutterstock.com***

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on page 6.

Pearson Education Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the world

Visit us on the World Wide Web at:
www.pearsonglobaleditions.com

Authorized adaptation from the United States edition, entitled Visual C# How to Program,6th Edition, ISBN 978-0-13-460154-0 by Paul Deitel and Harvey Deitel published by Pearson Education © 2017.

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library

10 9 8 7 6 5 4 3 2 1

ISBN-10: 1-292-15346-6

ISBN-13: 978-1-292-15346-9

Typeset by GEX Publishing Services

Printed and bound in Malaysia

In memory of William Siebert, Professor Emeritus of Electrical Engineering and Computer Science at MIT:

Your use of visualization techniques in your Signals and Systems lectures inspired the way generations of engineers, computer scientists, educators and authors present their work.

Harvey and Paul Deitel

Trademarks

DEITEL and the double-thumbs-up bug are registered trademarks of Deitel and Associates, Inc.

Microsoft® Windows®, and Microsoft Visual C#® are registered trademarks of the Microsoft Corporation in the U.S.A. And other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

UNIX is a registered trademark of The Open Group.

Throughout this book, trademarks are used. Rather than put a trademark symbol in every occurrence of a trademarked name, we state that we are using the names in an editorial fashion only and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Contents

Online Topics

PDFs presenting additional topics for advanced college courses and professionals are available through the book's Companion Website:

```
http://www.pearsonglobaleditions.com/deitel
```

New copies of this book come with a Companion Website access code that's located on the book's inside front cover. If the access code is already visible or there is no card, you purchased a used book or an edition that does not come with an access code.

Preface

Welcome to the world of desktop, mobile and web app development with Microsoft's® Visual C#® programming language. *Visual C# How to Program, 6/e* is based on C# 6[1] and related Microsoft software technologies. You'll be using the .NET platform and the Visual Studio® Integrated Development Environment on which you'll conveniently write, test and debug your applications and run them on Windows® devices. The Windows operating system runs on desktop and notebook computers, mobile phones and tablets, game systems and a great variety of devices associated with the emerging "Internet of Things."

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#. The book presents leading-edge computing technologies in a friendly manner appropriate for introductory college course sequences, based on the curriculum recommendations of two key professional organizations—the ACM and the IEEE.[2]

You'll study four of today's most popular programming paradigms:

- object-oriented programming,
- structured programming,
- generic programming and
- functional programming (new in this edition).

At the heart of the book is the Deitel signature *live-code approach*—rather than using code snippets, we generally present concepts in the context of complete working programs followed by sample executions. We include a broad range of example programs and exercises selected from computer science, business, education, social issues, personal utilities, sports, mathematics, puzzles, simulation, game playing, graphics, multimedia and many other areas. We also provide abundant tables, line drawings and UML diagrams for a more visual learning experience.

1. At the time of this writing, Microsoft has not yet released the official C# 6 Specification. To view an unofficial copy, visit `https://github.com/ljw1004/csharpspec/blob/gh-pages/README.md`
2. These recommendations include *Computer Science Curricula 2013 Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, December 20, 2013, The Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM), IEEE Computer Society.

Read the Before You Begin section after this Preface for instructions on setting up your computer to run the hundreds of code examples and to enable you to develop your own C# apps. The source code for all of the book's examples is available at

```
http://www.pearsonglobaleditions.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. Most of the book's examples work in Visual Studio on Windows 7, 8 or 10 (there is no 9). The code examples for the online presentation of the Universal Windows Platform (UWP) and XAML specifically require Windows 10.

Contacting the Authors

As you read the book, if you have a question, we're easy to reach at

```
deitel@deitel.com
```

We'll respond promptly.

Join the Deitel & Associates, Inc. Social Media Communities

Subscribe to the *Deitel® Buzz Online* newsletter

```
http://www.deitel.com/newsletter/subscribe.html
```

and join the Deitel social media communities on

- Facebook®—`http://facebook.com/DeitelFan`
- LinkedIn®—`http://linkedin.com/company/deitel-&-associates`
- YouTube®—`http://youtube.com/DeitelTV`
- Twitter®—`http://twitter.com/Deitel`
- Instagram®—`http://instagram.com/DeitelFan`
- Google+™—`http://google.com/+DeitelFan`

Object-Oriented Programming with an Early Objects Approach

The book introduces the basic concepts and terminology of object-oriented programming in Chapter 1. In Chapter 2, you'll *visually* manipulate objects, such as labels and images. In Chapter 3, Introduction to C# App Programming, 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.

Our early objects presentation continues in Chapters 5–9 with a variety of straightforward case studies. In Chapters 10–12, we take a deeper look at classes and objects,

present inheritance, interfaces and polymorphism, then use those concepts throughout the remainder of the book.

New C# 6 Features

6 We introduce key new C# 6 language features throughout the book (Fig. 1)—each defining occurrence is marked with a "6" margin icon as shown next to this paragraph.

C# 6 new language feature	First introduced in
`string` interpolation	Section 3.5
expression bodied methods and `get` accessors	Section 7.15
auto-implemented property initializers	Section 8.6.1
getter-only auto-implemented properties	Section 8.6.1
`nameof` operator	Section 10.5.1
null-conditional operator (`?.`)	Section 13.9.1
`when` clause for exception filtering	Section 13.10
`using static` directive	Section 21.3.1
null conditional operator (`?[]`)	Section 21.6
collection initializers for any collection with an `Add` extension method	Section 21.7
index initializers	Section 21.7

Fig. 1 | C# 6 new language features.

Interesting, Entertaining and Challenging Exercises

The book contains hundreds of exercises to practice the skills you learn. Extensive self-review exercises and answers are included for self-study. Also, each chapter concludes with a substantial set of exercises, which generally include

- simple recall of important terminology and concepts,
- identifying the errors in code samples,
- writing individual program statements,
- writing methods to perform specific tasks,
- writing C# classes,
- writing complete programs and
- implementing major projects.

Figure 2 lists only a sample 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 work on significant social problems. We hope you'll approach these exercises in the context of your own values, politics and beliefs. The solutions to most of the book's exercises are available only to college instructors who have adopted the book for their courses. See "Instructor Supplements" later in this Preface.

A sampling of the book's exercises

Carbon Footprint Calculator
Body-Mass-Index Calculator
Attributes of Hybrid Vehicles
Gender Neutrality
Stopwatch GUI
Login Form GUI
Calculator GUI
Alarm Clock GUI
Radio GUI
Displaying Shapes
Odd or Even?
Is a Number a Multiple of Another?
Separating an Integer's Digits
Multiplication Table of a Number
Account Class
Student Record Class
Asset Class
Coaching Class
Removing Duplicated Code
Target-Heart-Rate Calculator
Computerizing Health Records
Inventory Level Calculator
Sales-Commission Calculator
Discount Calculator
Find the Two Largest Numbers
Dangling-else Problem
Expanded Form
Decimal Equivalent of a Binary Number
Type of Parallelogram
Permutations
Infinite Series: Mathematical Constant *e*
World Population Growth
Enforcing Privacy with Cryptography
Bar Chart Display
Prime Numbers
Calculating Sales
Car-Pool Savings Calculator
Gas Mileage Calculator
Calculating the Value of log (*x*)
Pythagorean Triples
Global Warming Facts Quiz
Tax Plan Alternative: The "FairTax"
Rounding to a Specific Decimal Place
Hypotenuse of a Right Triangle
Displaying a Hollow Right-Isosceles Triangle of Any Character
Separating Digits
Temperature Conversions
Amicable Numbers
Prime Numbers
Reversing Digits
Letter Grades to a Four-Point Scale
Coin Tossing
Guess-the-Number Game
Distance Between Two Points
Craps Game with Betting
Towers of Hanoi
Computer-Assisted Instruction
Wages Rate
Identifying Multiples of Eight
Dice Game of Craps
Airline Reservations System
Knight's Tour Chess Puzzle
Eight Queens Chess Puzzle
Sieve of Eratosthenes
Tortoise and the Hare
Merging Arrays
Building Your Own Computer (Virtual Machine)
Polling
Querying an Array of Invoice Objects
Name Connector
Hemisphere Class
Depreciating-Value Class
Set of Integers
RationalNumber Class
HugeInteger Class
Tic-Tac-Toe
ComplexNumber Class
Vehicle Inheritance Hierarchy
Payroll System
Accounts Payable System
Polymorphic Banking Program
CarbonFootprint Interface: Polymorphism
Length Conversions
Painter
Guess the Number Game
Ecofont
Typing Tutor
Restaurant Bill Calculator
Story Writer
Pig Latin
Cooking with Healthier Ingredients
Spam Elimination
SMS Language
File of Product Details
Telephone-Number Words
Student Poll
Phishing Scanner
Bucket Sort
Palindromes
Evaluating Expressions with a Stack
Building Your Own Compiler
Generic Linear Search
SortedDictionary of Colors
Prime Factorization
Bucket Sort with LinkedList<int>
Sieve of Eratosthenes with BitArray
Credit-Inquiry Program
Rolling a Die 60,000,000 Times
Baseball Database App
Parsing with LINQ to XML
I/O-Bound vs. Compute-Bound Apps
Recursive Fibonacci

Fig. 2 | A sampling of the book's exercises.

A Tour of the Book

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

Introduction to Computing, Visual C# and Visual Studio 2015 Community Edition
The chapters in this module of the book

- Chapter 1, Introduction to Computers, the Internet and Visual C#
- Chapter 2, Introduction to Visual Studio and Visual Programming

introduce hardware and software fundamentals, Microsoft's .NET platform and Visual Programming. The vast majority of the book's examples will run on Windows 7, 8 and 10 using the *Visual Studio 2015 Community* edition with which we test-drive a fun **Painter** app in Section 1.12. Chapter 1's introduction to object-oriented programming defines key terminology and discusses important concepts on which the rest of the book depends.

Introduction to C# Fundamentals
The chapters in this module of the book

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

present rich coverage of C# programming fundamentals (data types, operators, control statements, methods and arrays) and introduce object-oriented programming through a series of case studies. These chapters should be covered in order. Chapters 5 and 6 present a friendly treatment of control statements and problem solving. Chapters 7 and 8 present rich treatments of methods and arrays, respectively. Chapter 8 briefly introduces exception handling with an example that demonstrates attempting to access 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 and Interfaces
- Chapter 13, Exception Handling: A Deeper Look

provide a deeper look at object-oriented programming, including classes, objects, inheritance, polymorphism, interfaces and exception handling. An optional online two-chapter case study on designing and implementing the object-oriented software for a simple ATM is described later in this preface.

Chapter 9 introduces Microsoft's Language Integrated Query (LINQ) technology, which provides a uniform syntax for manipulating data from various data sources, such as

arrays, collections and, as you'll see in later chapters, databases and XML. Chapter 9 is intentionally simple and brief to encourage instructors to begin covering LINQ technology early. Section 9.4 introduces the `List` collection, which we use in Chapter 12. Later in the book, we take a deeper look at LINQ, using LINQ to Entities (for querying databases) and LINQ to XML. Chapter 9's LINQ coverage can be deferred if you're in a course which either skips LINQ or defers coverage until later in the book—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.

Windows `Forms` Graphical User Interfaces (GUIs)
The chapters in this module of the book

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

present a detailed introduction to building GUIs using Windows `Forms`—instructors teaching Visual C# still largely prefer Windows `Forms` for their classes. Many of the examples in GUI Chapters 14–15 can be presented after Chapter 4. We also use Windows `Forms` GUIs in several other print and online chapters.

There are two other GUI technologies in Windows—Windows Presentation Foundation (WPF) and Universal Windows Platform (UWP). We provide optional online treatments of both.[3]

Strings and Files
The chapters in this module of the book

- Chapter 16, Strings and Characters: A Deeper Look
- Chapter 17, Files and Streams

present `string` processing and file processing, respectively. We introduce `string`s beginning in Chapter 4 and use them throughout the book. Chapter 16 investigates `string`s in more detail. Most of Chapter 16's examples can be presented at any point after Chapter 4. Chapter 17 introduces text-file processing and object-serialization for inputting and outputting entire objects. Chapter 17 requires Windows `Forms` concepts presented in Chapter 14.

Searching, Sorting and Generic Data Structures
The chapters in this module of the book:

- Chapter 18, Searching and Sorting
- Chapter 19, Custom Linked Data Structures
- Chapter 20, Generics
- Chapter 21, Generic Collections; Functional Programming with LINQ/PLINQ

introduce searching, sorting and data structures. Most C# programmers should use .NET's *built-in* searching, sorting and generic 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. Chapter 18 presents sev-

3. As of Summer 2016, Windows Forms, WPF and UWP apps all can be posted for distribution via the Windows Store. See `http://bit.ly/DesktopToUWP` for more information.

eral searching and sorting algorithms and uses Big O notation to help you compare how hard each algorithm works to do its job—the code examples use especially visual outputs to show how the algorithms work. In Chapter 19, we show how to implement your own custom data structures, including lists, stacks, queues and binary trees. The data structures in Chapter 19 store references to `objects`. Chapter 20 introduces C# generics and demonstrates how to create type-safe generic methods and a type-safe generic stack data structure.

Functional Programming with LINQ, PLINQ, Lambdas, Delegates and Immutability
In addition to generic collections, Chapter 21 now introduces functional programming, showing how to use it with LINQ to Objects to write code more concisely and with fewer bugs than programs written using previous techniques. In Section 21.12, with one additional method call, we'll demonstrate how PLINQ (Parallel LINQ) can improve LINQ to Objects performance substantially on multicore systems. The chapter's exercises also ask you to reimplement earlier examples using functional-programming techniques.

Database with LINQ to Entities and SQL Server
The chapter in this module

- Chapter 22, Databases and LINQ

presents a novice-friendly introduction to database programming with the ADO.NET Entity Framework, LINQ to Entities and Microsoft's free version of SQL Server that's installed with the Visual Studio 2015 Community edition. The chapter's examples require C#, object-oriented programming and Windows `Forms` concepts presented in Chapters 3–14. Several online chapters require the techniques presented in this chapter.

Asynchronous Programming
The chapter in this module

- Chapter 23, Asynchronous Programming with `async` and `await`

shows how to take advantage of multicore architectures by writing applications that can process tasks asynchronously, which can improve app performance and GUI responsiveness in apps with long-running or compute-intensive tasks. The `async` modifier and `await` operator greatly simplify asynchronous programming, reduce errors and enable your apps to take advantage of the processing power in today's multicore computers, smartphones and tablets. In this edition, we added a case study that uses the Task Parallel Library (TPL), `async` and `await` in a GUI app—we keep a progress bar moving along in the GUI thread in parallel with a lengthy, compute-intensive calculation in another thread.

A Tour of the Online Content

The printed book contains the core content (Chapters 1–23) for introductory and intermediate course sequences. Several optional online topics for advanced courses and professionals are available on the book's password-protected Companion Website

```
http://www.pearsonglobaleditions.com/deitel
```

New copies of this book come with a Companion Website access code that's located on the book's inside front cover. If the access code is already visible or there isn't an access code, you purchased a used book or an edition that does not come with an access code. Figure 3 lists the online topics, and Figure 4 lists a sample of the associated exercises.

Online topics
Web App Development with ASP.NET
XML and LINQ to XML
Universal Windows Platform (UWP) GUI, Graphics, Multimedia and XAML
REST Web Services
Cloud Computing with Microsoft Azure™
Using the Visual Studio Debugger
(Optional) Windows Presentation Foundation (WPF) GUI, Graphics, Multimedia and XAML
(Optional) ATM Case Study, Part 1: Object-Oriented Design with the UML
(Optional) ATM Case Study, Part 2: Implementing an OO Design in C#

Fig. 3 | Online topics on the *Visual C# How to Program, 6/e* Companion Website.

A sampling of the online chapters' exercises		
Guestbook App	College Loan Payoff Calculator App	Favorite Flickr Searches App
Web-Based Address Book	Car Payment Calculator App	Flag Quiz App
Enhanced Painter App	Mileage Calculator App	Phone Book App with Data Binding
PhotoViewer App	Body Mass Index Calculator App	Enhanced UsingGradients App
Data Bindings to LINQ queries	Target-Heart-Rate Calculator App	Enhanced DrawStars app
Snake PolyLine App	Phone-Book Web Service	Image Reflector App
Drawing App		Accessibility: Speech-Controlled Drawing App
Enhanced Tip Calculator App		
Mortgage Calculator App		

Fig. 4 | A sampling of the online chapters' exercises.

Web App Development with ASP.NET
Microsoft's .NET server-side technology, ASP.NET, enables you to create robust, scalable web-based apps. 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.

Extensible Markup Language (XML)
The Extensible Markup Language (XML) is pervasive in the software-development industry, e-business and throughout the .NET platform. It's used in most of this book's online topics. XML is required to understand XAML—a Microsoft XML vocabulary that's used to describe graphical user interfaces, graphics and multimedia for Universal Windows Platform (UWP) GUI, graphics and multimedia apps, Windows 10 Mobile apps and Windows Presentation Foundation (WPF) apps. We present XML fundamentals, then discuss LINQ to XML, which allows you to query XML content using LINQ syntax.

Universal Windows Platform (UWP) for Desktop and Mobile Apps
The Universal Windows Platform (UWP) is designed to provide a common platform and user experience across all Windows devices, including personal computers, smartphones,