



# Programming in Visual C# 2008

Julia Case Bradley  
Anita C. Millspaugh

---

**Computer and Information Technology**

**Programming in Visual C#  
2008**

**Bradley-Millsbaugh**

*McGraw-Hill*

*A Division of The McGraw-Hill Companies*



**McGraw-Hill Primis**

ISBN-10: 0-39-017402-5  
ISBN-13: 978-0-39-017402-4

Text:

Programming in Visual C# 2008

**Bradley-Millsbaugh**



This book was printed on recycled paper.

**Computer and Information Technology**

**<http://www.primisonline.com>**

Copyright ©2009 by The McGraw-Hill Companies, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without prior written permission of the publisher.

This McGraw-Hill Primis text may include materials submitted to McGraw-Hill for publication by the instructor of this course. The instructor is solely responsible for the editorial content of such materials.

# Computer and Information Technology

## Contents

Bradley–Millsbaugh • *Programming in Visual C# 2008*

<b>Front Matter</b>	<b>1</b>
Preface	1
To the Student	7
<b>1. Introduction to Programming and Visual C# 2008</b>	<b>9</b>
Text	9
<b>2. User Interface Design</b>	<b>75</b>
Text	75
<b>3. Variables, Constants, and Calculations</b>	<b>115</b>
Text	115
<b>4. Decisions and Conditions</b>	<b>165</b>
Text	165
<b>5. Menus, Common Dialog Boxes, and Methods</b>	<b>225</b>
Text	225
<b>6. Multiform Projects</b>	<b>267</b>
Text	267
<b>7. Lists, Loops, and Printing</b>	<b>301</b>
Text	301
<b>8. Arrays</b>	<b>339</b>
Text	339

<b>9. Web Applications</b>	<b>377</b>
Text	377
<b>10. Database Applications</b>	<b>419</b>
Text	419
<b>11. Data Files</b>	<b>459</b>
Text	459
<b>12. OOP: Creating Object-Oriented Programs</b>	<b>489</b>
Text	489
<b>13. Graphics, Animation, Sound, and Drag-and-Drop</b>	<b>543</b>
Text	543
<b>14. Additional Topics in C#</b>	<b>579</b>
Text	579
<b>Back Matter</b>	<b>621</b>
Appendix A: Answers to Feedback Questions	621
Appendix B: Methods for Working with Dates, Mathematics, and String Operations	635
Appendix C: Tips and Shortcuts for Mastering the Environment	643
Appendix D: Security	661
Glossary	665
Index	676

# PREFACE

Visual C# (C Sharp) is a relatively new language introduced by Microsoft along with Visual Studio. Its goal was to provide the ease of working with Visual Basic with the flexibility and power of the Java and C++ languages. The syntax of C# is similar to Java and C++ but the ease of creating a graphical user interface and an event-driven application rivals Visual Basic.

C# is fully object-oriented, compatible with many other languages using the .NET Framework. This book incorporates the object-oriented concepts throughout, as well as the syntax and terminology of the language.

C# is designed to allow the programmer to develop applications that run under Windows and/or in a Web browser without the complexity generally associated with programming. With very little effort, the programmer can design a screen that holds standard elements such as buttons, check boxes, radio buttons, text boxes, and list boxes. Each of these objects operates as expected, producing a “standard” Windows or Web user interface.

## About This Text

This textbook is intended for use in an introductory programming course, which assumes no prior knowledge of computer programming. The later chapters are also appropriate for professional programmers who are learning a new language to upgrade their skills.

This text assumes that the student is familiar with the Windows operating environment and can use an Internet browser application.

## Approach

This text incorporates the basic concepts of programming, problem solving, and programming logic, as well as the design techniques of an object-oriented event-driven language.

Chapter topics are presented in a sequence that allows the programmer to learn how to deal with a visual interface while acquiring important programming skills such as creating projects with objects, decisions, loops, and data management.

A high priority is given to writing applications that are easy for the user to understand and use. Students are presented with interface design guidelines throughout the text.

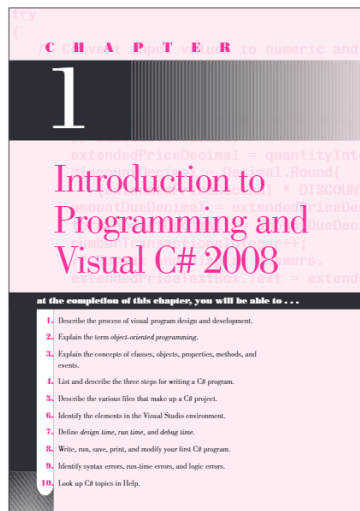
This text follows essentially the same sequence as the Bradley/Millsbaugh *Visual Basic* text. Object-oriented programming (OOP) is introduced in Chapter 1 and is used consistently in every chapter of the book.

The code for all in-chapter projects is available to instructors.

# TEXT FEATURES

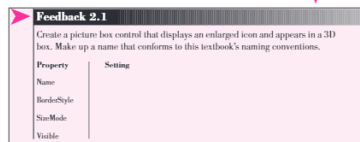
## Object-Oriented Concepts

are presented throughout the text to offer students an introduction to object-oriented design before learning to create their own classes.



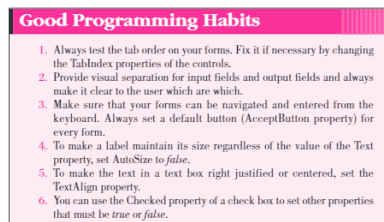
## Feedback Questions

give the students time to reflect on the current topic and to evaluate their understanding of the details.



## Interface Design Guidelines

are presented to offer students a better understanding of meeting user needs and employing industry standards.



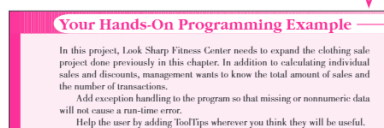
## Tips

in the margins help students avoid potential trouble spots in their programs and encourage them to develop good programming habits.



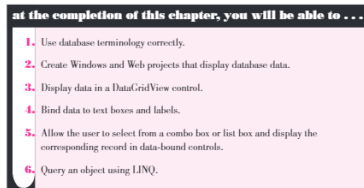
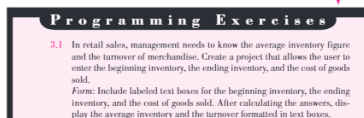
## Hands-On Programming Examples

guide students through the process of planning, writing, and executing C# programs.



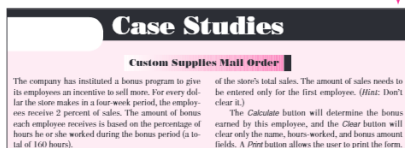
## Programming Exercises

test students' understanding of the programming skills covered in that chapter.



## Case Studies

provide continuing-theme exercises that may be used throughout the course, providing many opportunities to expand on previous projects.



## Learning Objectives

tell students what will be covered in the chapter and what they will be able to do after completing the chapter.

## Online Learning Center

Visit the Visual C# 2008 Web site at [www.mhhe.com/C#2008/](http://www.mhhe.com/C#2008/) for instructor and student resources.





## Changes in This Edition

This revision of the text is based on the Professional Edition of Visual Studio (VS) 2008. The 2008 version of VS includes Language-Integrated Queries (LINQ) for a more consistent means of querying a variety of data sources. The Windows Presentation Foundation (WPF) brings new dimensions to interface design. AJAX also has been added to provide faster postbacks for Web pages.

Microsoft has made many enhancements to the integrated development environment (IDE). The Editor now has a much richer IntelliSense. New tool windows for creating, managing, and applying styles in Web applications are now included in Chapter 9.

A new PrintForm component makes printing a form extremely easy and convenient for a classroom environment. The PrintForm component is part of a Microsoft download called the *Visual Basic Power Packs*, which can be added to the IDE for use with a C# program. PrintForm is covered in Chapter 2 for output to either the printer or a Print Preview window. Standard printing is still covered in Chapter 7.

LINQ is covered in Chapter 10 (“Database Applications”) and again in the XML section of Chapter 14.

Chapter 11 (“Data Files”) includes a simplified method for writing small amounts of data without performing an open operation.

Drag-and-drop for images is now covered in the graphics chapter (Chapter 13). Students learn the development techniques for this familiar operation.

Chapter 14 includes XML data files and an introduction to programming with WPF.

The text narrative, step-by-step exercises, screen captures, and appendices have all been updated to Visual Studio 2008. The screen captures are all based on Windows Vista.

## Features of This Text

Each chapter begins with identifiable objectives and a brief overview. Numerous coding examples as well as hands-on projects with guidance for the planning and coding appear throughout. Thought-provoking feedback questions give students time to reflect on the current topic and to evaluate their understanding of the details. The end-of-chapter items include a chapter summary, review questions, programming exercises, and four case studies. The case studies provide a continuing-theme exercise that may be used throughout the course.

**Chapter 1, “Introduction to Programming and Visual C# 2008,”** introduces Microsoft’s Visual Studio integrated development environment (IDE). The single environment is used for multiple languages. A step-by-step program gets students into programming very quickly (quicker than most books). The chapter introduces the OOP concepts of objects, properties, methods, and events. The elements of debugging and using the Help system also are introduced.

**Chapter 2, “User Interface Design,”** demonstrates techniques for good program design, including making the interface easy for users as well as guidelines for designing maintainable programs. Several controls are introduced, including text boxes, masked text boxes, rich text boxes, group boxes, check boxes, radio buttons, and picture boxes. A new section covers the controls in

## P R E F A C E

ix

**Chapter 3, “Variables, Constants, and Calculations,”** presents the concepts of using data and declaring the data type. Students learn to follow standards to indicate the data type of variables and constants and to use the proper scope.

Error handling uses the `try/catch/finally` structure, which is introduced in this chapter along with calculations. The student learns to display error messages using the `MessageBox` class and also learns about the OOP concept of overloaded constructors.

**Chapter 4, “Decisions and Conditions,”** introduces taking alternate actions based on expressions formed with the relational and logical operators. This chapter uses `if` statements to validate user input. Multiple decisions are handled with both nested `if` statements and the case structure (the `switch` statement).

The debugging features of the IDE are covered, including a step-by-step exercise that covers stepping through program statements and checking intermediate values during execution.

**Chapter 5, “Menus, Common Dialog Boxes, and Methods,”** covers the concepts of writing and calling general methods. Students learn to include both menus and context menus in projects, display common dialog boxes, and use the input provided by the user.

**Chapter 6, “Multiform Projects,”** adds splash forms and About forms to a project. Summary data are presented on a separate form.

**Chapter 7, “Lists, Loops, and Printing,”** incorporates list boxes and combo boxes into projects, providing the opportunity to discuss looping procedures and printing lists of information. Printing is accomplished in .NET using a graphics object and a callback event. The printing controls also include a Print Preview, which allows students and instructors to view output without actually printing it.

**Chapter 8, “Arrays,”** introduces arrays, which follow logically from the lists covered in Chapter 7. Students learn to use single- and multidimension arrays, table lookups, structures, and arrays of structures.

**Chapter 9, “Web Applications,”** introduces programming using Web Forms, which are used to create Web pages that execute in a browser application. Students learn to design and develop simple Web applications. CSS styles and AJAX provide the ability to create improved, more efficient Web sites.

**Chapter 10, “Database Applications,”** introduces ADO.NET, which is Microsoft’s technology for accessing data in a database. This chapter shows how to create binding sources, table adapters, and datasets. Programs include accessing data from both Windows Forms and Web Forms. Students learn to bind data tables to a data grid and bind individual data fields to controls such as labels and text boxes. LINQ is used to query an array and a database.

**Chapter 11, “Data Files,”** presents the techniques for data file handling. Students learn to save and read small amounts of data using streams. The `StreamWriter` and `StreamReader` objects are used to store and reload the contents of a combo box.

**Chapter 12, “OOP: Creating Object-Oriented Programs,”** explains more of the theory of object-oriented programming. Although we

have been using OOP concepts since Chapter 1, in this chapter students