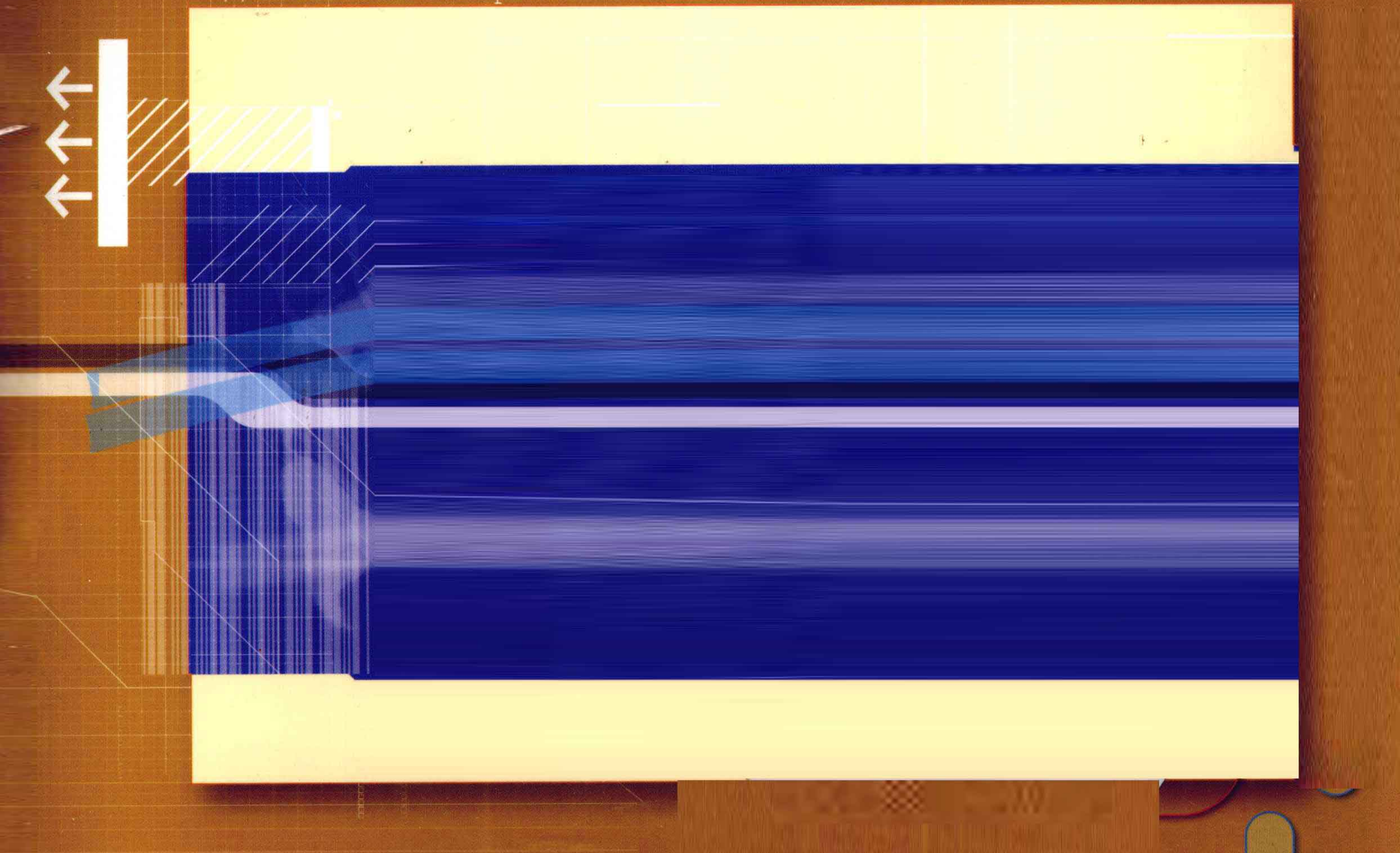
Programming in C# .NET

```
private void calculateButton_Click(object sender, System.EventArgs e)
{
    //Calculate the price and discount
```



```
        int intQuantity;
        decimal decPrice,
```

Julia Case Bradley • Anita C. Millspaugh

Programming in C# .NET

Julia Case Bradley
Mt. San Antonio College

Anita C. Millspaugh
Mt. San Antonio College

McGraw Hill

Boston Burr Ridge, IL Dubuque, IA Madison, WI New York San Francisco St. Louis
Bangkok Bogotá Caracas Kuala Lumpur Lisbon London Madrid Mexico City
Milan Montreal New Delhi Santiago Seoul Singapore Sydney Taipei Toronto

PROGRAMMING IN C# .NET

Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY 10020. Copyright © 2004 by The McGraw-Hill Companies, Inc. All rights reserved. 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 the prior written consent of The McGraw-Hill Companies, Inc., including, but not limited to, in any network or other electronic storage or transmission, or broadcast for distance learning.

Some ancillaries, including electronic and print components, may not be available to customers outside the United States.

This book is printed on acid-free paper.

domestic 1 2 3 4 5 6 7 8 9 0 QPD/QPD 0 9 8 7 6 5 4 3
international 1 2 3 4 5 6 7 8 9 0 QPD/QPD 0 9 8 7 6 5 4 3

ISBN 0-07-285232-1

Editor-in-chief: *Bob Woodbury*
Sponsoring editor: *Steve Schuetz*
Developmental editor: *Craig S. Leonard*
Marketing manager: *Paul Murphy*
Media producer: *Greg Bates*
Senior project manager: *Jean Lou Hess*
Production supervisor: *Gina Hangos*
Coordinator freelance design: *Artemio Ortiz Jr.*
Senior supplement producer: *Rose M. Range*
Senior digital content specialist: *Brian Nacik*
Cover design: *JoAnne Schopler*
Interior design: *Artemio Ortiz Jr.*
Typeface: *11/13 Bodoni*
Compositor: *GAC Indianapolis*
Printer: *Quebecor World Dubuque Inc.*

Library of Congress Cataloging-in-Publication Data

Bradley, Julia Case.
Programming in C#.NET / Julia Case Bradley, Anita C. Millspaugh.
p. cm.
ISBN 0-07-285232-1 (alk. paper) -- ISBN 0-07-121564-6 (international : alk. paper)
1. Microsoft Visual BASIC. 2. BASIC (Computer program language) 3. Microsoft .NET. I. Millspaugh, A. C. (Anita C.) II. Title.

QA76.76.B3 B73 2003
005.2'768--dc21

2002041070

INTERNATIONAL EDITION ISBN 0-07-121564-6

www.mhhe.com

PREFACE

C# (C Sharp) is a new language introduced by Microsoft along with Visual Studio .NET. Their 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 new .NET Framework. This book incorporates the object-oriented concepts throughout, as well as the syntax and terminology of the language.

C# .NET 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, 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 to use. Students are presented with interface design guidelines throughout the text.

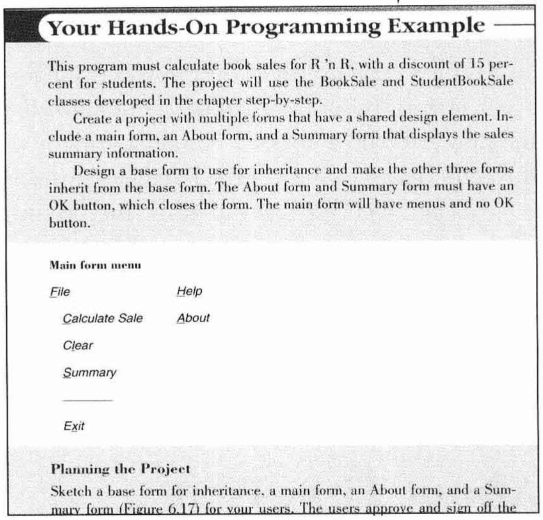
This text follows essentially the same sequence as the Bradley/Millspaugh Visual Basic .NET text. We have introduced object-oriented programming (OOP) in Chapter 1 and use its features in every chapter of the book.

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

Features of This Text

Hands-On Programming Examples

These complete programming exercises guide students through the process of planning, writing, and executing C# programs.



Your Hands-On Programming Example

This program must calculate book sales for R 'n R, with a discount of 15 percent for students. The project will use the BookSale and StudentBookSale classes developed in the chapter step-by-step.

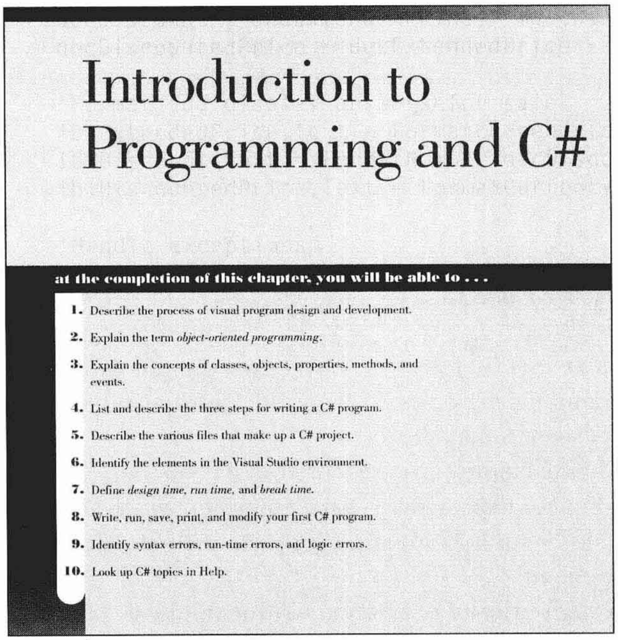
Create a project with multiple forms that have a shared design element. Include a main form, an About form, and a Summary form that displays the sales summary information.

Design a base form to use for inheritance and make the other three forms inherit from the base form. The About form and Summary form must have an OK button, which closes the form. The main form will have menus and no OK button.

Main form menu

Planning the Project

Sketch a base form for inheritance, a main form, an About form, and a Summary form (Figure 6.17) for your users. The users approve and sign off the



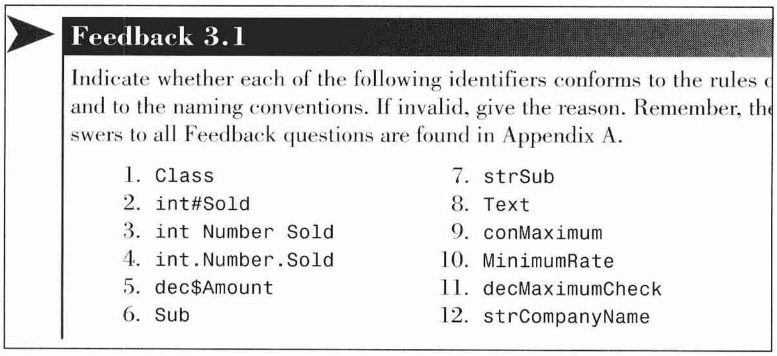
Introduction to Programming and C#

at the completion of this chapter, you will be able to . . .

1. Describe the process of visual program design and development.
2. Explain the term *object-oriented programming*.
3. Explain the concepts of classes, objects, properties, methods, and events.
4. List and describe the three steps for writing a C# program.
5. Describe the various files that make up a C# project.
6. Identify the elements in the Visual Studio environment.
7. Define *design time*, *run time*, and *break time*.
8. Write, run, save, print, and modify your first C# program.
9. Identify syntax errors, run-time errors, and logic errors.
10. Look up C# topics in Help.

Learning Objectives

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



Feedback 3.1

Indicate whether each of the following identifiers conforms to the rules c
and to the naming conventions. If invalid, give the reason. Remember, th
swers to all Feedback questions are found in Appendix A.

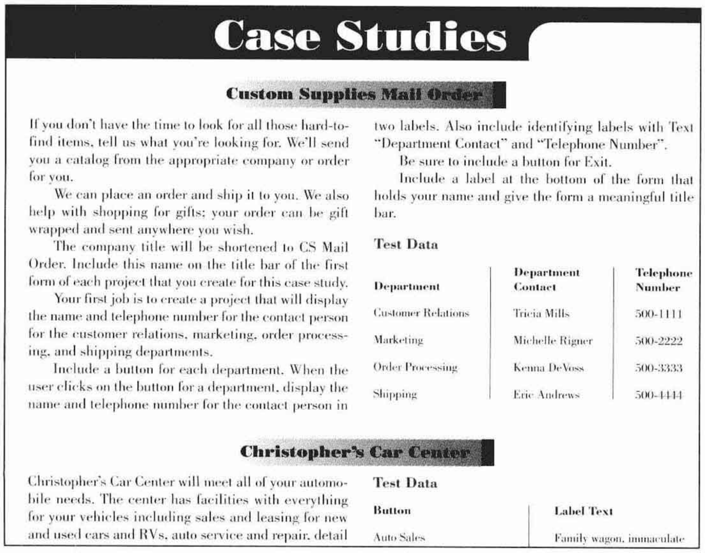
1. `Class`
2. `int#Sold`
3. `int Number Sold`
4. `int.Number.Sold`
5. `dec$Amount`
6. `Sub`
7. `strSub`
8. `Text`
9. `conMaximum`
10. `MinimumRate`
11. `decMaximumCheck`
12. `strCompanyName`

Feedback Questions

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

Case Studies

The Case Studies provide continuing-theme exercises that may be used throughout the course.



Case Studies

Custom Supplies Mail Order

If you don't have the time to look for all those hard-to-find items, tell us what you're looking for. We'll send you a catalog from the appropriate company or order for you.

We can place an order and ship it to you. We also help with shopping for gifts; your order can be gift wrapped and sent anywhere you wish.

The company title will be shortened to CS Mail Order. Include this name on the title bar of the first form of each project that you create for this case study.

Your first job is to create a project that will display the name and telephone number for the contact person for the customer relations, marketing, order processing, and shipping departments.

Include a button for each department. When the user clicks on the button for a department, display the name and telephone number for the contact person in two labels. Also include identifying labels with Text "Department Contact" and "Telephone Number".

Be sure to include a button for Exit.

Include a label at the bottom of the form that holds your name and give the form a meaningful title bar.

Test Data

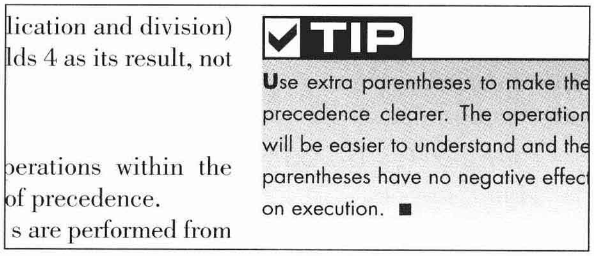
Department	Department Contact	Telephone Number
Customer Relations	Tricia Mills	500-1111
Marketing	Michelle Rigner	500-2222
Order Processing	Kenna DeVoss	500-3333
Shipping	Eric Andrews	500-4444

Christopher's Car Center

Christopher's Car Center will meet all of your automobile needs. The center has facilities with everything for your vehicles including sales and leasing for new and used cars and RVs, auto service and repair, detail

Test Data

Button	Label Text
Auto Sales	Family wagon, immaculate



lication and division)
lds 4 as its result, not

berations within the
of precedence.
s are performed from

✓ TIP

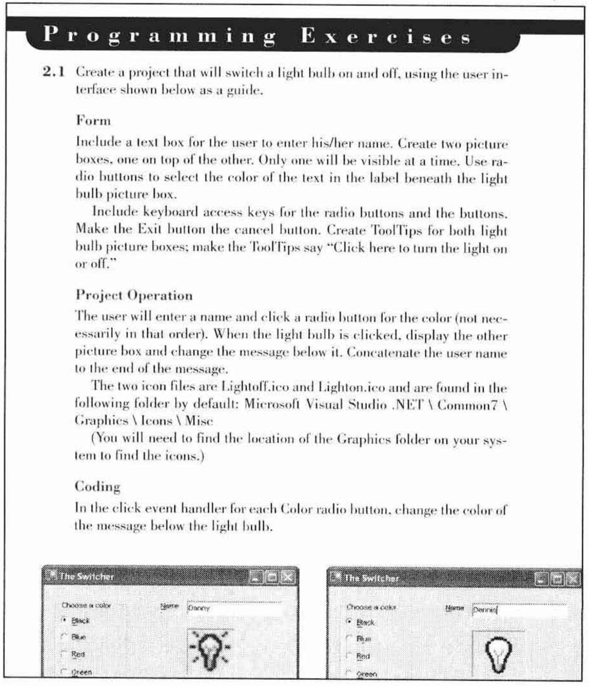
Use extra parentheses to make the precedence clearer. The operation will be easier to understand and the parentheses have no negative effect on execution. ■

TIPs

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

Programming Exercises

The Programming Exercises test students' understanding of the programming skills covered in the chapter.



Programming Exercises

2.1 Create a project that will switch a light bulb on and off, using the user interface shown below as a guide.

Form

Include a text box for the user to enter his/her name. Create two picture boxes, one on top of the other. Only one will be visible at a time. Use radio buttons to select the color of the text in the label beneath the light bulb picture box.

Include keyboard access keys for the radio buttons and the buttons. Make the Exit button the cancel button. Create ToolTips for both light bulb picture boxes; make the ToolTips say "Click here to turn the light on or off."

Project Operation

The user will enter a name and click a radio button for the color (not necessarily in that order). When the light bulb is clicked, display the other picture box and change the message below it. Concatenate the user name to the end of the message.

The two icon files are Lightoff.ico and Lighton.ico and are found in the following folder by default: Microsoft Visual Studio .NET \ Common7 \ Graphics \ Icons \ Misc

(You will need to find the location of the Graphics folder on your system to find the icons.)

Coding

In the click event handler for each Color radio button, change the color of the message below the light bulb.

The Switcher

Choose a color

● Black
○ Blue
○ Red
○ Green

Name: Danny

The Switcher

Choose a color

● Black
○ Blue
○ Red
○ Green

Name: Dennis

Chapter Organization

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 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 C#," introduces Microsoft's new Visual Studio integrated development environment (IDE). The single environment is now 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 are also introduced.

Chapter 2, "More Controls," 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, group boxes, check boxes, radio buttons, and picture boxes.

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.

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 conditions formed with the relational and logical operators. In addition to the `if` statement and the case structure, this chapter shows several techniques for validating input data. The debugging features of the IDE are covered, including a step-by-step tutorial covering stepping through program statements and checking intermediate values during execution.

Chapter 5, "Menus, Common Dialog Boxes, and Methods," covers the Menu Editor. Menus and context menus are components that are added to a component tray. Students learn to include both menus and context menus in projects, as well as to write general methods.

Chapter 6, "OOP: Creating Object-Oriented Programs," explains the theory of object-oriented programming. Although we have been using OOP concepts since Chapter 1, in this chapter students learn the terminology and application of OOP. Inheritance is covered for visual objects (forms) and for extending existing classes. The samples are kept simple enough for an introductory class.

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. The printing controls also include a Print Preview to view output without actually printing it.

Chapter 8, "Arrays," introduces arrays, which follow logically from the lists covered in Chapter 7. Structures are also introduced.

Chapter 9, "Programming with Web Forms," 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.

Chapter 10, "Accessing Database Files," introduces ADO.NET, which is Microsoft's new technology for accessing data in a database. This chapter shows how to create connections, data 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.

Chapter 11, "Saving Data and Objects in 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.

Object serialization is used to persist objects. The hands-on example includes both serialization (saving) and deserialization (restoring) objects.

Chapter 12, " Graphics and Animation," covers the classes and methods of GDI+. The chapter covers Graphics objects, pens, and brushes for drawing shapes and lines. Animation is accomplished using the Timer control and the `SetBounds` method for moving controls.

Chapter 13, "Additional Topics in C#," introduces some advanced topics. This final chapter covers multiple document interfaces (MDI), toolbars and status bars, and creating reports from databases using Crystal Reports.

The Appendices offer important additional material. Appendix A holds the answers to all Feedback questions. Appendix B covers methods for dates, math, and string handling. In the OOP style, actions are accomplished with methods of the Math class and String class. Appendix C is on mastering the Visual Studio environment, and Appendix D shows the differences between C# and Visual Basic.

Resources for Instructors

Instructors Manual The Instructor's Manual is available on CD-ROM or on the book's Web site. It contains the following features:

- Objectives with built-in summaries for each chapter.
- Teaching suggestions.
- Answers to the Review Questions from the back of each chapter.
- Chapter topics covered in the Programming Exercises.

Testbank The Testbank provides questions that cover the terminology and concepts found in each chapter. The test questions appear in the form of True/False and Multiple Choice.

Diploma by Brownstone. Diploma is the most flexible, powerful, and easy-to-use computer-based testing system available for higher education.

The Diploma system allows instructors to create an exam as a printed version, as a LAN-based online version, or as an Internet version. Diploma also includes grade book features, which automate the entire testing process.

PowerPoint Presentation The PowerPoint presentation follows the outline of the Instructor's Manual and gives instructors a resource for presenting the text material to a classroom.

Figures from the Book All of the illustrations, screenshots, and tables are available electronically for use in presentations, transparencies, or handouts.

Online Learning Center (www.mhhe.com/cit/program/bradley/csharp) Designed to provide a wide variety of learning opportunities for students, the Web site includes additional Programming Exercises, Self-Quizzes for students, downloadable data files, and other great resources for both instructors and students.

Digital Solutions to Help You Manage Your Course

PageOut PageOut is our Course Web Site Development Center that offers a syllabus page, URL, McGraw-Hill Online Learning Center content, online exercises and quizzes, grade book, discussion board, and an area for student Web pages.

Available free with any McGraw-Hill/Irwin product, PageOut requires no prior knowledge of HTML, no long hours of coding, and a way for course coordinators and professors to provide a full-course Web site. PageOut offers a series of templates—simply fill them with your course information and click on one of 16 designs. The process takes under an hour and leaves you with a professionally designed Web site. We'll even get you started with sample Web sites, or enter your syllabus for you! PageOut is so straightforward and intuitive, it's little wonder why more than 12,000 college professors are using it. For more information, visit the PageOut Web site at www.pageout.net.

The Online Learning Center can be delivered through any of these platforms:

McGraw-Hill Learning Architecture (TopClass)

Blackboard.com

Ecollege.com (formerly Real Education)

WebCT (a product of Universal Learning Technology

McGraw-Hill has partnerships with WebCT and Blackboard to make it even easier to take your course online. Now you can have McGraw-Hill content delivered through the leading Internet-based learning tool for higher education. At McGraw-Hill, we have the following service agreements with WebCT and Blackboard:

Instructor Advantage. Instructor Advantage is a special level of service McGraw-Hill offers in conjunction with WebCT designed to help you get up and running with your new course. A team of specialists will be immediately available to ensure everything runs smoothly through the life of your adoption.

Instructor Advantage Plus. Qualified McGraw-Hill adopters will be eligible for an even higher level of service. A certified WebCT or Blackboard specialist will provide a full day of on-site training for you and your staff. You will then have unlimited email and phone support through the life of your adoption. Contact your local McGraw-Hill representative for more details.

Powerweb Powerweb for Information Technology is an exciting online product available for use with Using Information Technology. A nominally priced token grants students access through our Web site to a wealth of resources—all corresponding to the text. Features include an interactive glossary; current events with quizzing, assessment, and measurement options; Web survey; links to related text content; and WWW searching capability via Northern Lights, an academic search engine.

Thank You

Many people have worked very hard to design and produce this text. We would like to thank our editors, Steve Schuetz and Craig Leonard, and the publisher George Werthman. Our thanks also to the many people who produced this text including Jean Lou Hess, Artemio Ortiz, and Betsy Blumenthal.

We greatly appreciate Robert Price, Deanna Tague, and Haroon Ahmed of Microsoft for their thorough technical reviews, constructive criticism, and many valuable suggestions. Many thanks to Laura Claytor for her helpful assistance. And most importantly, we are grateful to Dennis, Richard, Tricia, Eric, and Kenna for their support and understanding through the long days and busy phone lines.

The Authors

We have had fun writing about C#. We hope that this feeling is evident as you read this book and that you will enjoy learning or teaching this outstanding programming language.

Julia Case Bradley

Anita C. Millspaugh

To the Student

The best way to learn to program in C# is to do it. If you enter and run the sample projects, you will be on your way to writing applications. Reading the examples without trying to run them is like trying to learn a foreign language or mathematics by just reading about it. Enter the projects, look up your questions in the extensive MSDN Help files, and make those projects *run*.

Installing C#

For the programs in this text, you need to install the .NET Framework, IIS (Internet Information Services), C#, and the MSDN (Microsoft Developers Network) library, which contains all of Help and many instructive articles. You do not need to install C++ or VB.

You need IIS if you want to write any Web applications that run in a browser. All of the programs in Chapter 9 require IIS, as well as some programs in Chapter 10.

The order of installation is important. You must install IIS and the .NET Framework before installing C#.

Format Used for C# Statements

C# statements, methods, and functions are shown in `this font`. Any values that you must supply are in *`italics`*. *Example:*

```
const int identifier = Value;
```

As you work your way through this textbook, note that you may see a subset of the available options for a C# statement or method. Generally, the options that are included reflect those covered in the chapter. If you want to see the complete format for any statement or all versions of a method, refer to Help.

J.C.B.

A.C.M.

CONTENTS