

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.

The screenshot shows the beginning of a chapter. At the top, it says 'CHAPTER' in large letters. Below that is a large number '1'. The main title is 'Introduction to Programming and Visual C# 2008'. Below the title, there is a list of learning objectives under the heading 'at the completion of this chapter, you will be able to . . .'. The list contains 10 items, including describing the process of visual program design, explaining object-oriented programming, and writing a C# program.

CHAPTER

1

Introduction to Programming and Visual C# 2008

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 *debug 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.

Interface Design Guidelines

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

Good Programming Habits

1. Always test the tab order on your forms. Fix it if necessary by changing the `TabIndex` properties of the controls.
2. Provide visual separation for input fields and output fields and always make it clear to the user which are which.
3. Make sure that your forms can be navigated and entered from the keyboard. Always set a default button (`AcceptButton` property) for every form.
4. To make a label maintain its size regardless of the value of the `Text` property, set `AutoSize` to *false*.
5. To make the text in a text box right justified or centered, set the `TextAlign` property.
6. You can use the `Checked` property of a check box to set other properties that must be *true* or *false*.

Tips

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

TIP Use two ampersands when you want to make an ampersand appear in the `Text` property: `&Health &&Welfare` for "Health & Welfare". ■

Feedback Questions

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

Feedback 2.1
Create a picture box control that displays an enlarged icon and appears in a 3D box. Make up a name that conforms to this textbook's naming conventions.

Property	Setting
Name	
BorderStyle	
SizeMode	
Visible	

Hands-On Programming Examples

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

Your Hands-On Programming Example

In this project, Look Sharp Fitness Center needs to expand the clothing sale project done previously in this chapter. In addition to calculating individual sales and discounts, management wants to know the total amount of sales and the number of transactions.

Add exception handling to the program so that missing or nonnumeric data will not cause a run-time error.

Help the user by adding `ToolTips` wherever you think they will be useful.

Programming Exercises

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

Programming Exercises

- 3.1 In retail sales, management needs to know the average inventory figure and the turnover of merchandise. Create a project that allows the user to enter the beginning inventory, the ending inventory, and the cost of goods sold.
- Form:* Include labeled text boxes for the beginning inventory, the ending inventory, and the cost of goods sold. After calculating the answers, display the average inventory and the turnover formatted in text boxes.

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

1. Use database terminology correctly.
2. Create Windows and Web projects that display database data.
3. Display data in a DataGridView control.
4. Bind data to text boxes and labels.
5. Allow the user to select from a combo box or list box and display the corresponding record in data-bound controls.
6. Query an object using LINQ.

Case Studies

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

Case Studies

Custom Supplies Mail Order

The company has instituted a bonus program to give its employees an incentive to sell more. For every dollar the store makes in a four-week period, the employees receive 2 percent of sales. The amount of bonus each employee receives is based on the percentage of hours he or she worked during the bonus period (a total of 160 hours).

of the store's total sales. The amount of sales needs to be entered only for the first employee. (*Hint:* Don't clear it.)

The *Calculate* button will determine the bonus earned by this employee, and the *Clear* button will clear only the name, hours-worked, and bonus amount fields. A *Print* button allows the user to print the form.

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/ for instructor and student resources.

Programming in Visual C# 2008
Julia Case Bradley, Mt. San Antonio College
Anita C. Millsbaugh, Mt. San Antonio College
ISBN: 0073517216
Copyright year: 2010

Feature Summary

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# 2005," 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.

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 conditions 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 logging 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, "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 latest 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.

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 the Power Pack including PrintForm and the Shape and Line controls.

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

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 13, “Graphics, Animation, Sound, and Drag-and-Drop,” 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. Students learn to play sounds using the `SoundPlayer` class. Video files are played using Windows Media Player. Drag-and-drop events are used to transfer images and the contents of a text box to a list box.

Chapter 14, “Additional Topics in C#,” introduces some advanced programming topics. This final chapter covers validating user input using Error Providers and the Validating event of controls. Students learn to create applications using multiple document interfaces (MDI), create toolbars and status bars using `ToolStrip` and `StatusStrip` controls, and add Web content to a Windows Form using the `WebBrowser` control. The code-snippet feature is introduced. Reading and writing XML text files are covered. The chapter also covers LINQ to XML.

An introduction to Windows Presentation Framework (WPF) includes using WPF Interoperability with a standard Windows Form and creating a WPF Form project.

The appendixes 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 programming style, actions are accomplished with methods of the `Math` class and `String` class. Appendix C gives tips and shortcuts for mastering the Visual Studio environment, and Appendix D discusses security issues for both Windows and Web programming.

Thank You

Many people have worked very hard to design and produce this text. We would like to thank our editors, Scott Davidson and Alaina Grayson. Our thanks also to the many people who produced this text, including Marlena Pechan and Betsy Blumenthal.

We greatly appreciate Robert Price and Peter van der Goes for their thorough technical reviews, constructive criticism, and many valuable suggestions. Thank you to Theresa Berry for her work on the exercise solutions. And, most importantly, we are grateful to Dennis and Richard 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 Visual 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 Visual C#

For the programs in this text, you need to install the .NET Framework v 3.5, Visual C# 2008, and the MSDN (Microsoft Developers Network) library, which contains all of Help and many instructive articles.

You can download the Express Edition of Visual C# and Visual Web Developer from msdn.microsoft.com/express. Using these two products, you can complete most of the exercises in this text.

Format Used for Visual C# Statements

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

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.

About the Authors

Julia Bradley is a professor emeritus of Computer Information Systems at Mt. San Antonio College. She developed and taught computer programming courses for 25 years and then took early retirement from teaching in order to write full time. Most recently she has taught courses in introductory and advanced Visual Basic, Access programming, and Microsoft Office. She began writing BASIC textbooks in 1984 using MS-BASIC (GW-BASIC) and has authored or co-authored texts in Macintosh Basic, QuickBasic, QBasic, Visual Basic, C#, Java, the Internet, and desktop publishing.

Anita Millsbaugh teaches programming courses in Visual Basic, C#, and Java at Mt. San Antonio College and has served as chair of the department for eight years. She received her MBA from California State Polytechnic University, with a bachelor's degree in Computer Information Systems. She has taught faculty at the National Computer Educator's Institute and also has led Great Teacher's Conferences for Mt. SAC and for California Vocational Faculty.