

PROGRAMMING IN VISUAL BASIC 2010

Julia Case Bradley
Mt. San Antonio College

Anita C. Millspaugh
Mt. San Antonio College





PROGRAMMING IN VISUAL BASIC 2010

Published by McGraw-Hill, a business unit of The McGraw-Hill Companies, Inc., 1221 Avenue of the Americas, New York, NY, 10020. Copyright © 2011 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.

1 2 3 4 5 6 7 8 9 0 WDQ/WDQ 1 0 9 8 7 6 5 4 3 2 1 0

ISBN 978-0-07-351725-4
MHID 0-07-351725-9

Vice president/Editor in chief: *Elizabeth Haefele*
Vice president/Director of marketing: *John E. Biernat*
Executive sponsoring editor: *Scott Davidson*
Director of development: *Sarah Wood*
Developmental editor II: *Alaina Grayson*
Editorial coordinator: *Alan Palmer*
Marketing manager: *Tiffany Wendt*
Lead digital product manager: *Damian Moshak*
Digital development editor: *Kevin White*
Director, Editing/Design/Production: *Jess Ann Kosic*
Project manager: *Marlena Pechan*
Buyer II: *Sherry L. Kane*
Senior designer: *Srijan Savanoste*
Manager, Digital production: *Janean A. Utley*
Media project manager: *Cathy L. Tepper*
Cover design: *Jon Resh*
Typeface: *11/13 Bodoni*
Compositor: *Aptara®, Inc.*
Printer: *Worldcolor*
Cover credit: © *Veer*

Library of Congress Cataloging-in-Publication Data

Bradley, Julia Case.
Programming in Visual Basic 2010 / Julia Case Bradley, Anita C. Millsbaugh.
p. cm.
Includes index.
ISBN-13: 978-0-07-351725-4 (alk. paper)
ISBN-10: 0-07-351725-9 (alk. paper)
I. Microsoft Visual BASIC. 2. BASIC (Computer program language) I. Millsbaugh,
A. C. (Anita C.) II. Title.
QA76.73.B3B6968 2011
005.2'768—dc22

2010013744

The Internet addresses listed in the text were accurate at the time of publication. The inclusion of a Web site does not indicate an endorsement by the authors or McGraw-Hill, and McGraw-Hill does not guarantee the accuracy of the information presented at these sites.

PREFACE

Visual Basic (VB) has become such a popular programming language for several reasons. VB is easy to learn, which makes it an excellent tool for understanding elementary programming concepts. In addition, it has evolved into such a powerful and popular product that skilled Visual Basic programmers are in demand in the job market.

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

Visual Basic 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. VB is a fully object-oriented language, which includes inheritance and polymorphism. Object-oriented programming (OOP) is introduced in Chapter 1, and its features appear in every chapter of the book.

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.

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.



Good Programming Habits

1. Always test the tab order on your forms. Fix it if necessary by changing the `TabIndex` property of the controls.
2. Provide visual separation for input fields and output fields, and always make it clear to the user which one which.
3. Make sure that your forms can be navigated and entered from the keyboard. Always set an `Accept` button (`Acceptation` 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`.

INTERFACE GUIDELINES

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

Feedback 2.2

1. Write the Basic statements to clear the text box called `CompanyTextOut` and reset the location pointer into the line.
2. Write the Basic statements to clear the label called `CompanyLabel` and place the location pointer into a text box called `OrderTextOut`.
3. What will be the effect of each of these Basic statements?
 - (a) `PrintInchesOutOnScreen = True`
 - (b) `ColorPrintInScreen.Checked = True`
 - (c) `DrawingsLocation.Visible = False`
 - (d) `LocationLabel.ForeColor = Color.Red`
 - (e) `CityLabel.Text = CityTextOut.Text`
 - (f) `MessageBox.Show.Checked = True`

Setting Properties Based on User Actions

Often you need to change the Enabled or Visible property of a control based on an action of the user. For example, you may have controls that are disabled or invisible until the user signs in. In the following example, when the user logs in and clicks the Sign In button, several controls become visible, others become invisible, and a group box of radio buttons is enabled.

FEEDBACK QUESTIONS

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

TIPS

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

variable. It's usually a good idea to create a variable for the message and format the message before calling the `MessageBox` method, if nothing else, it makes your code easier to read and follow.

Combining Values into a Message String

You can concatenate a literal such as "Total Sales:" with the value from a variable. You may need to include an extra space inside the literal to make sure that the value is separated from the literal.

Use MessageFormat For Formatting

```
MessageBox.Show("Total Sales: " & TotalSales.ToString("C"))  
MessageBox.Show(MessageFormat("Sales Summary: {0}", MessageFormat.C))
```

Creating Multiple Lines of Output

If your message is too long for one line, you can use the `vbCrLf` character to insert a new line character into the string message. Use the Visual Studio Intellisense constant `Environment.NewLine` to determine line endings. You can concatenate this constant into a message string to set up multiple lines.

```
MessageBox.Show("Line 1\nLine 2\nLine 3", MessageBoxButtons.OKCancel)
```

TIP

Specify only the message for a "bold and dim" message box for debugging purposes. It will display an OK button and an empty title bar. `MessageBox.Show(" ", "Error")`

Feedback 3.2

Assume that `FrogInJugger = 10`, `ToadInJugger = 6`, and `FillingInJugger = 4`.

• What will be displayed for each of the following statements?

1. If `FrogInJugger > FillingInJugger` Then `FrogInJugger.Checked = True` Else `FrogInJugger.Checked = False` End If
2. If `FrogInJugger > ToadInJugger & FillingInJugger` Then `ResultTextOut.Text = "It's a frog."` Else `ResultTextOut.Text = "It's a toad and its filling."` End If
3. If `FillingInJugger > ToadInJugger And FrogInJugger <= 5 Or ToadInJugger = 0` Then `ResultTextOut.Text = "It's a frog."` Else `ResultTextOut.Text = "It's a toad."` End If
4. Write the statements necessary to compare the numeric values stored in `AppleInJugger` and `OrangeInJugger`. Display in `MessageBox` which has more, the apples or the oranges.
5. Write the Basic statements that will test the current value of `BalanceDecided`. When `BalanceDecided` is greater than zero, the check box for `Funds Available`, called `FundsAvailable`, should be selected, the `BalanceDecided` set back to zero, and `CreditsInJugger` incremented by one. When `BalanceDecided` is zero or less, `FundsAvailable` should not be selected (do not change the value of `BalanceDecided` or increment the counter).

TIP

Remember to help you catch errors. Most Basic string constants are Else with the last conditional. If dependent on the indentation.

Using If Statements with Radio Buttons and Check Boxes

In Chapter 2 you used the `CheckedChanged` event for radio buttons and check boxes to ensure that only one radio button or check box is selected at a time.

FOR THE STUDENT



STUDENT DATA

available on the text's Web site offers a debugging project, database files for the programming exercises and case studies, graphics, and sound files.



Visit the **VISUAL BASIC 2010** Web site at:
<http://www.mhhe.com/VB2010/> for
instructor and student resources.

FOR THE INSTRUCTOR

INSTRUCTOR WEB SITE

includes: Instructor's Manual with teaching hints, outlines, and a matrix of the chapter features required for each programming exercise; PowerPoint Slides; Testing Files (using EZ Test and in Word files); as well as Solutions to End-of-Chapter Exercises.

PROGRAMMING IN VISUAL BASIC 2010
Instructor Edition

BRADLEY BELLUSCHI

Study Any Where
Online LearningCenter

Second Edition, 2009

Programming in Visual Basic 2010

Julia Case Bradley, Mt. San Antonio College
Julia C. Millonig, Mt. San Antonio College

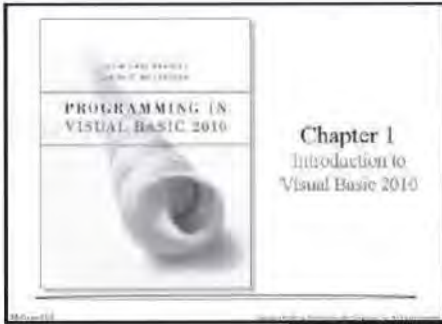
This text incorporates the basic concepts of programming, problem solving, programming logic, as well as the design techniques of an object-oriented, event-driven language, viz. 2010, a fully object-oriented language, which involves inheritance and polymorphic (Object-oriented programming OOP), is introduced in Chapter 1. User interfaces appear in every chapter of the book.

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, classes, forms, 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.

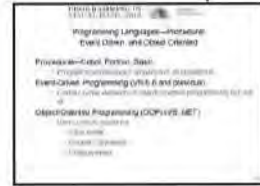
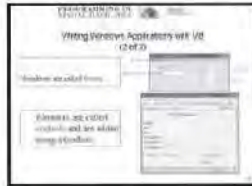
To learn more about the book, the website supports, please visit the Information Center.

© 2009 Pearson Education, Inc. All rights reserved. Printed in the United States of America. This book is published under the name of the Pearson Education Group. All other trademarks are the property of their respective owners.



POWERPOINT PRESENTATIONS

provide instructors with complete, detailed presentations that walk students through the important concepts covered in each chapter.



PROGRAMMING IN VISUAL BASIC 2010

BRACKET
WELLS/BLACK

Information Center

[Home](#) | [Learning Center](#) | [About the Book](#) | [Contact Us](#) | [Feedback](#) | [Privacy Policy](#) | [Terms of Use](#)

Programming in Visual Basic 2010
 Bill Chen, Bradley M. Kim, Barbara Collette
 JAMES C. BRIDGES, JR., Kent State University
 ISBN: 0073017129
 Copyright year: 2011

This book introduces the basic framework of programming, problem solving, programming logic, as well as the design techniques of an object-oriented, fourth-generation language, VB 2010 as a full-featured, object-oriented language, which includes inheritance and polymorphism. Object-oriented programming (OOP) concepts listed in Chapter 1 and its books to assist in every chapter of the book.

Chapters topics are presented in a sequence that allows the programmer to learn how to deal with a visual interface while applying advanced programming techniques in creating projects with objects, arrays, loops, and data management.

A high priority is given to writing applications that are easy for the user to understand and to avoid, whenever possible, with extensive design, structured, algorithm, flowchart, and code.

Visit our on-line site for free (all-in-one) online, with audio, video, and interactive, if you're interested in buying about additional textbooks, lectures, and courses for online.

© 2011, Mhhe.com, All rights reserved.
 Printed in the United States of America. This book is printed on acid-free paper.

Visit the **VISUAL BASIC 2010** Web site at:
<http://www.mhhe.com/VB2010/> for
 instructor and student resources.

Changes in This Edition

This revision of the text is based on Visual Basic Professional 2010. VB 2010 provides for elimination of the line continuation character under most circumstances. The array and object initializers also have been improved.

The narrative, step-by-step exercises, screen captures, and appendices have all been updated to VB 2010. The screen captures are all based on Windows 7. A section covering collection objects has been added to the chapter on arrays.

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 review, questions, programming exercises, and four case studies.

Chapter 1, “Introduction to Visual Basic 2010,” introduces Microsoft’s Visual Studio integrated development environment (IDE). The single environment is used for multiple programming languages. A step-by-step program gets students into programming very quickly (quicker than most books). The PrintForm control is included to allow students to easily submit screen captures of the form at run time. 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, “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, rich text boxes, masked text boxes, group boxes, check boxes, radio buttons, picture boxes, and the new 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 and scope of variables and constants and always to use Option Strict, which forces adherence to strong data typing.

Error handling is accomplished using structured exception handling. The Try/Catch/Finally structure 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 the If statement to validate input data. Multiple decisions are handled with both nested If statements and the Select Case structure.

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, Sub Procedures, and Function Procedures,” covers the concepts of writing and calling general sub procedures and function procedures. Students learn to include both menus and context menus in projects, display the Windows 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. The `Friend` keyword is introduced.

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

Chapter 9, “Web Applications,” introduces Web applications using Web Forms. Students learn to design and develop simple Web applications that consist of Web pages that execute in a browser application. Multiple-page Web sites are covered along with validator controls and an introduction to state management.

Chapter 10, “Database Applications,” 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. LINQ is used to query system processes.

Chapter 11, “Data Files,” presents the VB object-oriented techniques for data file handling. Students learn to save and read small amounts of data using the `My` object and 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. `My.Computer.Audio.Play` is used to provide sound, and drag-and-drop events are used to transfer the contents of a text box to a list box and to move images.

Chapter 14, “Additional Topics in Visual Basic,” introduces some advanced VB 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 is covered using the new XML literals and using LINQ.

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

The appendices offer important additional material. Appendix A holds the answers to all Feedback questions. Appendix B covers methods and functions for math, string handling, and date manipulation. In OOP style, most actions that were formerly done with functions are now accomplished with methods of the Math class and String class.

Appendix C, on mastering the Visual Studio environment, is based on the .NET IDE and includes instructions for using snap lines for form design. Appendix D discusses security issues for both Windows and Web programming.

Acknowledgments

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 Marlana Pechan, David Shapiro, and Betsy Blumenthal.

We greatly appreciate Theresa Berry of Mt. San Antonio College and Peter van der Goes of Rose State College for their thorough technical reviews, constructive criticism, and many valuable suggestions. We would like to thank Brenda Nielsen of Mesa Community College for her work in creating the PowerPoint Presentations that accompany this text and to Theresa Berry for the Instructor’s Manual and Test Bank. 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 teaching and writing about Visual Basic. 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 Basic 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 Basic

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

You can download the Express Edition of Visual Basic 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 Basic Statements

Visual Basic statements, methods, and functions are shown in **this font**. Any values that you must supply are in *italics*. Optional items are in [square brackets]. Braces and a vertical bar indicate that you must choose one or the other value {one | other}.

Examples

```
Const Identifier [As Datatype] = Value  
Do {While | Until} Condition
```

As you work your way through this textbook, note that you may see a subset of the available options for a Visual Basic 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.

This page intentionally left blank

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 and C# at Mt. San Antonio College and 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.

This page intentionally left blank

BRIEF CONTENTS

Chapter 1

Introduction to Visual Basic
2010 1

Chapter 2

User Interface Design 63

Chapter 3

Variables, Constants, and
Calculations 105

Chapter 4

Decisions and Conditions 155

Chapter 5

Menus, Common Dialog Boxes,
Sub Procedures, and Function
Procedures 209

Chapter 6

Multiform Projects 249

Chapter 7

Lists, Loops, and Printing 283

Chapter 8

Arrays and Collections 325

Chapter 9

Web Applications 361

Chapter 10

Database Applications 397

Chapter 11

Data Files 437

Chapter 12

OOP: Creating Object-Oriented
Programs 467

Chapter 13

Graphics, Animation, Sound, and
Drag-and-Drop 523

Chapter 14

Additional Topics in Visual
Basic 559

Appendix A

Answers to Feedback
Questions 601

Appendix B

Methods and Functions for
Working with Dates, Financial
Calculations, Mathematics, and
String Operations 615

Appendix C

Tips and Shortcuts for Mastering
the Environment 629

Appendix D

Security 641

Glossary 645

Index 656