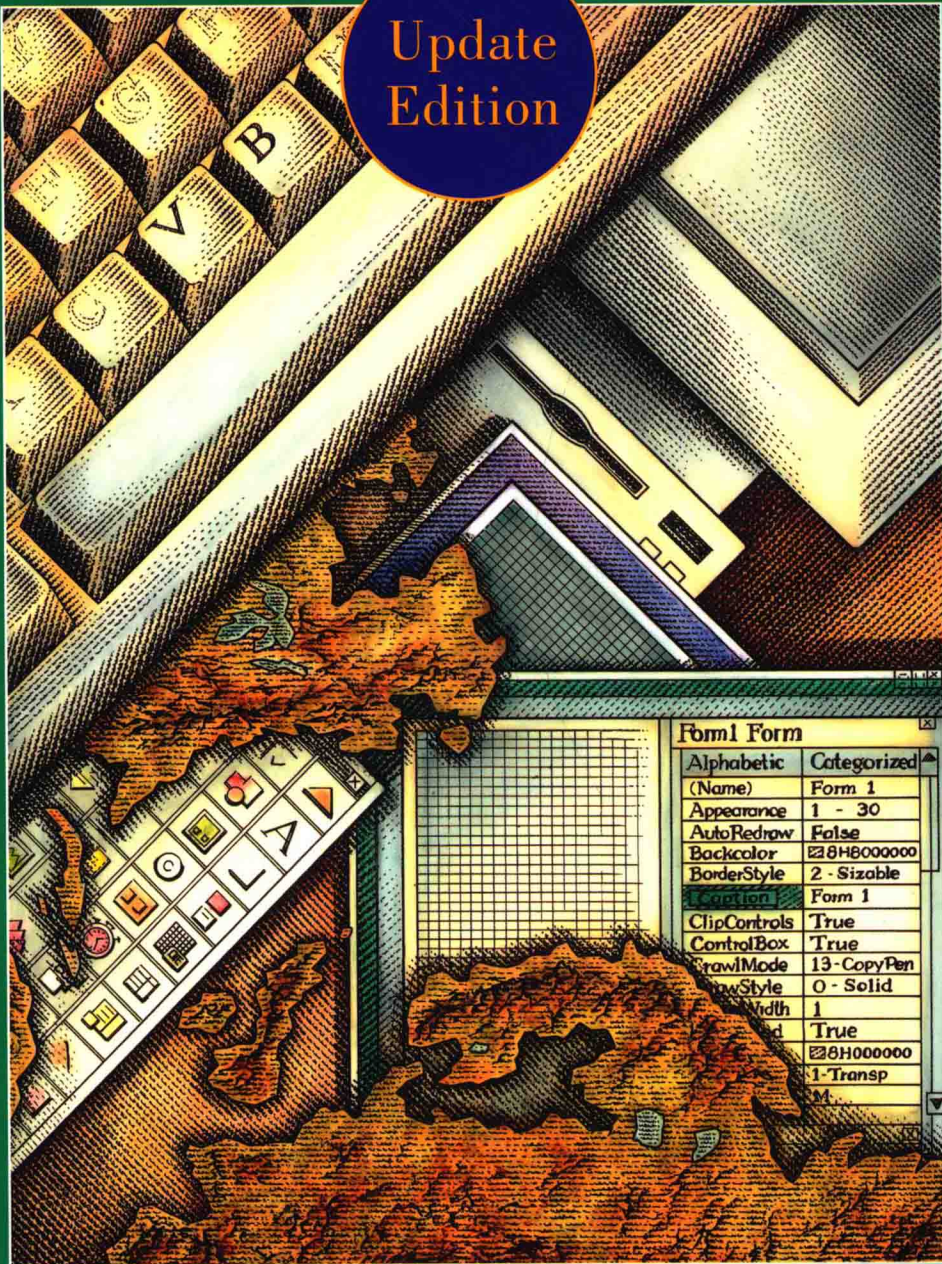


Programming in Visual Basic® 6.0

Update
Edition



Julia Case Bradley • Anita C. Millspaugh

Programming in Visual Basic

Version 6.0

Update Edition

Julia Case Bradley
Mt. San Antonio College

Anita C. Millsbaugh
Mt. San Antonio College



**McGraw-Hill
Irwin**

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

McGraw-Hill Higher Education

A Division of The McGraw-Hill Companies

PROGRAMMING IN VISUAL BASIC 6.0 UPDATE EDITION

Published by McGraw-Hill/Irwin, an imprint of The McGraw-Hill Companies, Inc. 1221 Avenue of the Americas, New York, NY, 10020. Copyright © 2002 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 QPD/QPD 0 9 8 7 6 5 4 3 2 1

ISBN 0-07-251381-0

Publisher: *George Werthman*

Sponsoring editor: *Steve Schuetz*

Developmental editor: *Craig S. Leonard*

Senior marketing manager: *Jeff Parr*

Associate project manager: *Destiny Rynne*

Production supervisor: *Debra R. Sylvester*

Senior designer: *Jenny El-Shamy*

Lead supplement producer: *Marc Mattson*

Senior producer, Media technology: *David Barrick*

Cover design: *Peter Siu*

Typeface: *11/13 Bodoni*

Compositor: *GAC Indianapolis*

Printer: *Quebecor World Dubuque Inc.*

Library of Congress Cataloging-in-Publication Data

Bradley, Julia Case.

Programming in Visual Basic version 6.0 update edition / Julia Case Bradley, Anita C. Millspough.
p. cm.

ISBN 0-07-251381-0 (alk. paper)

1. BASIC (Computer program language) 2. Microsoft Visual BASIC. I. Millspough,

A. C. (Anita C.) II. Title.

QA76.73.B3 B698 2002

005.26'8—dc21

2001044260

www.mhhe.com

Programming in Visual Basic

Version 6.0

Update Edition

Preface

As the world turns to graphical user interfaces, computer programming languages are changing to accommodate the shift. Visual Basic 6 is designed to allow the programmer to develop applications that run under Windows without the complexity generally associated with Windows programming. With very little effort, the programmer can design a screen that holds standard Windows elements such as command buttons, check boxes, option buttons, text boxes, and list boxes. Each of these Windows objects operates as expected, producing a “standard” Windows user interface.

Visual Basic 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.

About This Text

This textbook is intended for use in an introductory programming course, which assumes no prior knowledge of computer programming. However, many of the later chapters are appropriate for an advanced-level course. 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.

Approach

This text incorporates the basic concepts of programming, problem solving, and programming logic, as well as the design techniques of an 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 loops, decisions, and data management.

The later chapters may be used in various sequences to accommodate the needs of beginning and advanced-level courses, as well as a shorter quarter system or a semester-long course. For a shorter course, the professor may choose to skip the chapter on data files and cover only the first of the two database chapters.

New in This Edition of the Text

This updated edition of the text provides some new material and some reorganization. The biggest changes are:

- A new appendix introducing VB.NET, the next version of Visual Basic.
- The inclusion of ActiveX Data Objects (ADO) in the two database chapters: Chapters 11 and 12. The older-technology DAO controls were replaced with ADO.
- The addition of selecting and sorting Recordsets using the Find method and Filter and Sort properties in Chapter 12.
- The modification of the random file project in Chapter 10 to allow coverage without first covering Chapter 9.
- The addition of the Data Environment to Chapter 12, along with displaying related tables in a grid. SQL selection was removed from the chapter—postponed to the advanced text.
- The addition to Chapter 9 of a discussion of multitier applications.
- The addition of two case studies to each chapter.

Introduced in the Previous Edition

The features introduced in the Visual Basic 6.0 Edition have been retained. The text places an emphasis on the planning steps of project design to encourage students to develop good programming habits from the start.

The programs conform to Microsoft's newly published coding conventions, which define a three-character prefix for variable names to indicate data type.

The new formatting functions (`FormatNumeric`, `FormatCurrency`, `FormatPercent`, and `FormatDateTime`) simplify the formatting of output. And the new `Validate` event and `CausesValidation` properties simplify the validation of input data.

The advanced techniques chapter (Chapter 15) includes sections on creating an MDI project with parent and child forms, as well as creating shortcut menus.

Students and instructors will appreciate the appendix, "Tips and Shortcuts for Mastering the VB Environment." This reference brings together many helpful tips that can save a programmer significant amounts of time.

The instructor materials include suggested coding standards and masters of forms for project planning that can be reproduced and distributed to students. The solutions to all exercises are available to instructors for downloading on the Web.

Introduced in Visual Basic 6

The upgrade from Visual Basic version 5 to version 6 is significant, but not as great a change as the upgrade from version 4 to version 5. The major changes allow developers to use VB for Web page development and perform more robust data management.

VB version 6 introduces new formatting functions to simplify formatting output for display or print. Many new string and numeric functions are also introduced, as well as the new `Validate` event for input controls and the `CausesValidation` property for most controls.

Introduced in Visual Basic 5

Visual Basic 5:

- Runs much faster than previous versions and incorporates more features required for professional application development. VB5 is now competitive with C++ for object-oriented program development.
- Incorporates many helpful new features in the editor, making it easier for beginners as well as advanced programmers to enter and edit code. For example:
 - Drag-and-drop editing for moving and copying lines.
 - Pop-up lists of available data types when declaring variables.
 - Pop-up lists of allowable properties and methods for controls.
 - Tips showing formats and arguments for functions and statements that appear automatically as you enter program code.
- Is easier to debug than previous versions. For example:
 - Data tips, similar to tooltips, display the current contents of variables, properties, and expressions, and pop up when you point to the expression during break time.
 - You can easily set breakpoints in code by clicking in the margin of a statement.
 - During break time, you can drag the highlighted line to set the next statement to execute.
- Includes many new controls. For example:
 - Many ActiveX controls for programming on the Web.
 - A new Web Browser control that allows you to retrieve and display Web pages in an application.

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 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 walks the student through the creation of a first Visual Basic project, incorporating command buttons and labels. The programming environment is introduced along with the concepts of objects and their related properties, methods, and events. Students are taught to create a new folder and store their project inside the folder.

Chapter 2 continues coverage of controls, including text boxes, option buttons, check boxes, frames, images, lines, and shapes. It also covers some of the finer points of using the environment and working with keyboard access keys, multiple controls, and alignment. The color constants are used at this point to lead the novice programmer into Chapter 3, which introduces variables and constants. The text includes naming conventions to make the scope and data type of a variable or constant easier to determine from the coding syntax. Standards also provide for the use of `Option Explicit` to force the declaration of all variables and constants.

Chapter 4 introduces the relational and logical operators and their use with the `If` statement. Input validation and message boxes are also covered. In Chapter 5 students learn to set up custom menus and to write their own sub functions and sub procedures. Multiple forms, global variables, and standard code modules are presented in Chapter 6.

Chapter 7 incorporates list boxes and combo boxes into the projects, providing the opportunity to discuss looping procedures and printing lists of information. The list concept leads logically into the use of variable arrays and control arrays in Chapter 8.

Chapter 9 introduces the terminology of object-oriented programming and provides a step-by-step tutorial for creating a new class, instantiating objects of the new class, creating a collection class to hold references to the objects, and displaying and modifying objects from the collection. It also introduces the concepts of multitier applications.

Chapter 10 covers both sequential and random files, but the material may be covered in sections.

Chapters 11 and 12 deal with the use of Visual Basic as a front end for database programming. The projects display and update tables created by a database application such as Microsoft Access. Chapter 12 includes using a data-bound grid, error trapping, and the Data Environment.

The drag-and-drop feature of Windows programming is introduced in Chapter 13. This chapter normally brings great enthusiasm from students as they learn to deal with the source and target objects. The examples and assignments provide a blend of practical and just-for-fun applications. This approach is also true of Chapter 14, which introduces the graphics methods and graphics controls.

Chapter 15, the final chapter, covers various topics that build a bridge from Visual Basic to other applications. These include using and creating ActiveX controls, the Windows API, DLLs, OLE, Visual Basic for Applications, and MDI and SDI applications, and creating shortcut menus.

Acknowledgments

We would like to express our appreciation to the many people who have contributed to the successful completion of this text. Most especially, we thank our students at Mt. San Antonio College who helped class-test the material and who greatly influenced the manuscript.

Many people have worked very hard to design and produce this text, including George Werthman, Steve Schuetz, Craig Leonard, Destiny Rynne, Jenny El-Shamy, Betsy Blumenthal, and Bette Ittersagen.

We greatly appreciate Diane Murphey, Theresa Berry, and Dennis Fraser for their thorough technical reviews, constructive criticism, and many valuable suggestions. Thanks also to Theresa for writing the Instructors Manual and the exercise solutions. And most important, we are grateful to our families for their support and understanding through the long days and busy phone lines.

We want to thank our reviewers, who have made many helpful suggestions:

Gary R. Armstrong
Shippensburg University

Dennis Clarke
Hillsborough Community College

Charles Massey
University of North Carolina—Asheville

Ronald L. Burgher
Metropolitan Community College

Diane Murphey
Oklahoma Panhandle State University

Sheila J. Pearson
Southern Arkansas University

Thomas S. Pennington
Maple Woods Community College

Anita Philipp
Oklahoma City Community College

Hwang Santai
Purdue University—Fort Wayne

Debbie Tesch
Xavier University

Judy Yaeger
Western Michigan University

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 Windows applications. Reading the examples without trying to run them is like trying to learn a foreign language or mathematics just by reading about it. Enter the projects, look up your questions in Visual Basic's excellent Help files, and make those projects *run*.

Format Used for Visual Basic Statements

Visual Basic statements and functions are shown in *this font*. Any values 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}.

Example:

```
[Let] {Variable|Object.Property} = Expression
```

As you work your way through this textbook, note that you may see a subset of the available options for a Visual Basic statement. Generally, the options that are included will reflect those covered in the chapter. If you want to see the complete format for any statement, refer to online Help in the MSDN library or on the Web.

J.C.B.
A.C.M.

Contents

I			
Introduction to Visual Basic	1		
Writing Windows Applications with Visual Basic	2		
<i>The Windows Graphical User Interface</i>	2		
Programming Languages—Procedural, Object Oriented, and Event Driven	3		
<i>The Object Model</i>	3		
<i>Versions of Visual Basic</i>	4		
Writing Visual Basic Projects	4		
<i>The Three-Step Process</i>	4		
<i>Visual Basic Projects</i>	5		
The Visual Basic Environment	6		
<i>The Form Window</i>	6		
<i>The Project Explorer Window</i>	6		
<i>The Properties Window</i>	7		
<i>The Form Layout Window</i>	7		
<i>The Toolbox</i>	7		
<i>The Main Visual Basic Window</i>	7		
<i>The Toolbar</i>	7		
<i>The Form Location and Size Information</i>	8		
<i>Help</i>	8		
<i>Design Time, Run Time, and Break Time</i>	8		
Writing Your First Visual Basic Project	8		
<i>Set Up Your Visual Basic Workspace</i>	9		
<i>Plan the Project</i>	13		
<i>Define the User Interface</i>	13		
<i>Set Properties</i>	17		
<i>Write Code</i>	22		
<i>Visual Basic Code Statements</i>	22		
<i>Code the Event Procedures for Hello World</i>	24		
<i>Run the Project</i>	26		
<i>Save the Project</i>	28		
<i>Open the Project</i>	30		
		<i>Modify the Project</i>	30
		<i>Print the Project Documentation</i>	35
		Sample Printout	36
		<i>The Form Image</i>	36
		<i>The Code</i>	37
		<i>The Form as Text</i>	38
		Finding and Fixing Errors	39
		<i>Compile Errors</i>	39
		<i>Run-Time Errors</i>	40
		<i>Logic Errors</i>	40
		<i>Project Debugging</i>	41
		<i>Naming Rules and Conventions for Objects</i>	41
		Visual Basic Help	42
		<i>The MSDN Viewer</i>	43
		<i>Context-Sensitive Help</i>	44
		2	
		More Controls	51
		Introducing More Controls	52
		<i>Text Boxes</i>	53
		<i>Frames</i>	53
		<i>Check Boxes</i>	53
		<i>Option Buttons</i>	54
		<i>Images</i>	54
		<i>Setting a Border and Style</i>	56
		<i>The Shape Control</i>	56
		<i>The Line Control</i>	57
		Working with Multiple Controls	57
		<i>Selecting Multiple Controls</i>	57
		<i>Deselecting a Group of Controls</i>	58
		<i>Moving Controls as a Group</i>	58
		<i>Setting Properties for Multiple Controls</i>	59
		<i>Aligning Controls</i>	59
		Designing Your Applications for User Convenience	61
		<i>Designing the User Interface</i>	61
		<i>Defining Keyboard Access Keys</i>	62

<i>Setting the Default and Cancel Properties of Command Buttons</i>	63	<i>Counting</i>	111
<i>Setting the Tab Order for Controls</i>	63	<i>Calculating an Average</i>	112
<i>Setting the Form's Location on the Screen</i>	64	Programming Hints	118
<i>Creating ToolTips</i>	65		
Coding for the Controls	66	4	
<i>Clearing Text Boxes and Labels</i>	66	Decisions and Conditions	127
<i>Resetting the Focus</i>	66		
<i>Setting the Value Property of Option Buttons and Check Boxes</i>	66	If Statements	128
<i>Changing the Font Properties of Controls</i>	67	<i>Flowcharting If Statements</i>	130
<i>Changing the Color of Text</i>	68	Conditions	130
<i>Changing Multiple Properties of a Control</i>	68	<i>Comparing Numeric Variables and Constants</i>	131
<i>Concatenating Text</i>	69	<i>Comparing Strings</i>	132
<i>Continuing Long Program Lines</i>	69	<i>Testing for True or False</i>	135
<i>Using the Default Property of a Control</i>	70	<i>Comparing the Text Property of Text Boxes</i>	135
Programming Hints	77	<i>Comparing Uppercase and Lowercase Characters</i>	136
		<i>Compound Conditions</i>	136
		Nested If Statements	137
		Using If Statements with Option Buttons and Check Boxes	140
3		<i>A "Simple Sample"</i>	141
Variables, Constants, and Calculations	87	<i>Checking the Value of an Option Button Group</i>	142
Data—Variables and Constants	88	<i>Checking the Values of Multiple Check Boxes</i>	142
<i>Data Types</i>	89	Displaying Messages in Message Boxes	143
<i>Naming Rules</i>	90	<i>Selecting the MsgBox Icon</i>	144
<i>Naming Conventions</i>	90	<i>Displaying a Message String</i>	144
<i>Constants—Named and Intrinsic</i>	92	Input Validation	145
<i>Declaring Variables</i>	94	<i>Checking for Numeric Values</i>	145
<i>Scope of Variables</i>	96	<i>Checking for a Range of Values</i>	146
Calculations	99	<i>Checking for a Required Field</i>	146
Val Function	99	<i>Performing Multiple Validations</i>	147
Arithmetic Operations	100	Calling Event Procedures	147
<i>Order of Operations</i>	101	Debugging Visual Basic Projects	156
<i>Using Calculations in Code</i>	102	<i>Pausing Execution with the Break Button</i>	156
Formatting Data	102	<i>Forcing a Break</i>	156
A Calculation Programming Example	106	<i>Using the Immediate Window</i>	157
Planning the Project	106	<i>Checking the Current Values of Expressions</i>	158
The Project Coding Solution	110	<i>Stepping through Code</i>	158
Counting and Accumulating Sums	111	Debugging Step-by-Step Tutorial	159
<i>Summing Numbers</i>	111	<i>Test the Project</i>	159

<i>Break and Step Program Execution</i>	160
<i>View the Contents of Properties, Variables, and Conditions</i>	161
<i>Continue Project Execution</i>	162
<i>Test the White Total</i>	163
<i>Correct the Red Total Error</i>	163
<i>Correct the White Total Error</i>	164
<i>Force a Run-Time Error</i>	165

5

Menus, Sub Procedures, and Sub Functions 173

Menus	174
<i>Defining Menus</i>	174
<i>Creating a Menu—Step-by-Step</i>	176
<i>Coding for Menu Commands</i>	179
<i>Modifying a Menu</i>	179
<i>Toggling Check Marks On and Off</i>	181
<i>Standards for Windows Menus</i>	181

Common Dialog Boxes	181
<i>Using a Common Dialog Box</i>	183
<i>Using the Information from the Dialog Box</i>	184
<i>Setting Current Values</i>	186

Writing General Procedures	187
<i>Creating a New Sub Procedure</i>	187
<i>Passing Variables to Procedures</i>	188
<i>Passing Arguments ByVal or ByRef</i>	189
<i>Function Procedures versus Sub Procedures</i>	190
<i>Writing Function Procedures</i>	190
<i>Writing a Function with Multiple Arguments</i>	192
<i>Calling a Function with Multiple Arguments</i>	192
<i>Reusing Procedures</i>	192
<i>Breaking Calculations into Smaller Units</i>	194

Programming Hints	205
<i>Creating Executable Files</i>	205

6

Multiple Forms 213

Multiple Forms	214
<i>Creating New Forms</i>	214

<i>Adding and Removing Forms</i>	215
<i>The Hide and Show Methods</i>	216
<i>The Load and Unload Statements</i>	218
<i>The Me Keyword</i>	218
<i>Referring to Objects on a Different Form</i>	219

Standard Code Modules 219

Variables and Constants in Multiple-Form Projects	220
<i>Global Variables and Constants</i>	221
<i>Static Variables</i>	221
<i>Guidelines for Declaring Variables and Constants</i>	222

An About Box 223

A Splash Screen 225

Using Sub Main for Startup	226
<i>Coding the Standard Code Module</i>	226
<i>Coding the Splash Screen Event Procedure</i>	227
<i>Setting the Startup Form or Procedure</i>	227

Programming Hints	244
<i>Working with Maximized Forms</i>	244

7

Lists, Loops, and Printing 251

List Boxes and Combo Boxes	252
<i>Filling the List</i>	253
<i>Clearing the List</i>	255
<i>The ListIndex Property</i>	255
<i>The ListCount Property</i>	256
<i>The List Property</i>	256
<i>Removing an Item from a List</i>	257
<i>List Box and Combo Box Events</i>	257

Do/Loops	258
<i>The Boolean Data Type Revisited</i>	261
<i>Using a Do/Loop with a List Box</i>	261

For/Next Loops	262
<i>Negative Increment or Counting Backward</i>	264
<i>Conditions Satisfied before Entry</i>	265

<i>Altering the Values of the Loop Control Variables</i>	265
<i>Endless Loops</i>	265
<i>Exiting For/Next Loops</i>	266
Using the MsgBox Function	267
<i>Function Return Values</i>	268
<i>Specifying the Buttons and/or Icons to Display</i>	268
<i>MsgBox Example</i>	269
Using String Functions	269
<i>Examples Using Left, Right, and Mid Functions</i>	270
<i>The Len Function</i>	270
<i>Selecting Entries in a List Box</i>	271
Sending Information to the Printer	271
<i>Printing to the Printer</i>	272
<i>Formatting Lines</i>	272
<i>Selecting the Font</i>	276
<i>Terminating the Page or the Job</i>	276
<i>Printing the Contents of a List Box</i>	276
<i>Printing the Selected Item from a List</i>	277
<i>Aligning Decimal Columns</i>	277
 8	
Arrays	293
Control Arrays	294
The Case Structure	295
<i>Testing Option Buttons with the Case Structure</i>	297
Single-Dimension Arrays	299
<i>Subscripts</i>	300
<i>More on Subscripts</i>	301
For Each/Next Statements	301
<i>Initializing an Array Using For Each</i>	302
User-Defined Data Types	303
<i>Accessing Information with User-Defined Data Types</i>	304
Using Array Elements for Accumulators	305
Table Lookup	306
<i>Coding a Table Lookup</i>	309
Using List Boxes with Arrays	309
<i>The ItemData Property</i>	310
<i>Using Nonsequential ItemData Values</i>	311

<i>Adding Items with ItemData to a List</i>	312
Multidimensional Arrays	313
<i>Initializing Two-Dimensional Arrays</i>	314
<i>Nested For/Next Example</i>	314
<i>Printing a Two-Dimensional Table</i>	314
<i>Summing a Two-Dimensional Table</i>	315
<i>Lookup Operation for Two-Dimensional Tables</i>	316
Programming Hints	324
<i>The Array Function</i>	324

9

OOP—Creating Object-Oriented Programs 333

Visual Basic and Object-Oriented Programming	334
<i>Objects</i>	334
<i>Object-Oriented Terminology</i>	335
<i>Reusable Objects</i>	336
Classes	337
<i>Assigning Property Values</i>	337
Creating a New Class—Step-by-Step	339
<i>Define a New Class Module</i>	339
<i>Define the Class Properties</i>	340
<i>Add Property Procedures</i>	340
<i>Code a Method</i>	343
<i>Add General Remarks</i>	343
<i>Save the Class Module</i>	343
Creating a New Object	
Using a Class	344
<i>Define and Use a New Object</i>	344
<i>Save the Form and the Project</i>	346
<i>Run the Project</i>	347
Choosing When to Create New Objects	347
<i>Using the Set Statement</i>	348
<i>Early Binding versus Late Binding</i>	349
The Initialize and Terminate Events	349
Terminating Projects	350
Collections	351
<i>Creating a Collection</i>	351

<i>Creating a Unique Key in the CProduct Class</i>	352	The Err Object	397
<i>Creating the CProducts Class</i>	352	<i>The Err.Number Property</i>	397
<i>Adding Objects to a Collection</i>	353	<i>Raising Error Conditions</i>	398
<i>Removing a Member from a Collection</i>	354	<i>Coding Error-Handling Routines</i>	398
<i>Accessing a Member of a Collection</i>	355	<i>The Resume Statement</i>	399
<i>Returning the Count Property</i>	355	<i>Handling Errors</i>	400
<i>Setting a Default Property</i>	355	<i>The Exit Function and Exit Sub Statements</i>	401
<i>Using For Each/Next</i>	356	<i>Saving Changes to a File</i>	403
Using a Collection in a Form—Step-by-Step	358	<i>Sequential File Programming Example</i>	404
<i>Modifying the User Interface</i>	358	Random Data Files	408
<i>Declaring the Collection Object</i>	359	<i>Fixed-Length Strings</i>	409
<i>Coding the Add Procedure</i>	359	<i>Defining a Record for a Random File</i>	409
<i>Coding the Display Procedure</i>	359	<i>Opening a Random File</i>	409
<i>Coding the Clear Procedure</i>	360	<i>Reading and Writing a Random File</i>	410
<i>Running the Project</i>	360	<i>Accessing Fields in a Random Record</i>	411
Using the Object Browser	360	<i>Finding the End of a Random File</i>	412
<i>Examining VB Objects</i>	361	<i>The Seek Function</i>	412
<i>Examining Your Own Classes</i>	362	Using a List Box to Store a Key Field	413
Using a List Box to Store the Keys	362	<i>Trimming Extra Blanks from Strings</i>	414
<i>Using the List Box to Display an Object</i>	364	<i>Retrieving a Record from the File</i>	414
<i>Using the List Box to Remove an Object</i>	364	<i>Displaying the Selected Record</i>	415
Avoiding Global Variables	364	Updating a Random File	415
<i>Adding Properties to Forms</i>	365	<i>Locking the Contents of Controls</i>	417
Programming Hints	380	<i>Adding Records</i>	417
<i>Multi-tier Applications</i>	380	<i>Deleting a Record</i>	418
		<i>Editing Records</i>	419
		<i>The Read and Write Procedures</i>	420
		Programming Hints	434
		<i>The InputBox Function</i>	434
		<i>Using the InputBox to Randomly Retrieve a Record</i>	435
10		11	
Data Files	387	Accessing Database Files	441
Data Files	388	Visual Basic and Database Files	442
<i>Data Files and Project Files</i>	388	<i>Database Formats Supported by Visual Basic</i>	442
<i>Data File Terminology</i>	388	<i>Database Terminology</i>	442
<i>File Organizations</i>	389	<i>Creating Database Files for Use by Visual Basic</i>	443
<i>Opening and Closing Data Files</i>	389		
<i>The FreeFile Function</i>	391		
<i>Viewing the Data in a File</i>	391		
Sequential File Organization	391		
<i>Writing Data to a Sequential Disk File</i>	392		
<i>Creating a Sequential Data File</i>	392		
<i>Reading the Data in a Sequential File</i>	393		
<i>Finding the End of a Data File</i>	394		
<i>Locating a File</i>	394		
Trapping Program Errors	395		
<i>The On Error Statement</i>	395		

Using the ADO Data Control	444	<i>Putting It Together</i>	492
<i>The Data Control and Data-Bound Controls</i>	446	Validation and Error Trapping	493
Viewing a Database File—Step-by-Step	453	<i>Locking Text Boxes</i>	493
<i>Design and Create the Form</i>	453	<i>Validating Data in the Validate Event</i>	494
<i>Set the Properties for the Data Control</i>	455	<i>Trap Errors with On Error</i>	495
<i>Set the Properties for the Data-Bound Controls</i>	457	<i>Programming Example Showing Validation Techniques</i>	498
<i>Write the Code</i>	458	Searching for Records	503
<i>Run the Project</i>	459	<i>The Find Method</i>	503
Navigating the Database in Code	460	<i>No Record Found</i>	504
<i>The Recordset Object</i>	460	<i>Bookmarks</i>	505
<i>Using the MoveNext, MovePrevious, MoveFirst, and MoveLast Methods</i>	460	<i>The Filter Property</i>	505
<i>Checking for BOF and EOF</i>	460	Sorting a Recordset	506
Using List Boxes and Combo Boxes as Data-Bound Controls	461	Working with Database Fields	507
<i>Setting up a Lookup Table for a Field</i>	461	<i>Referring to Database Fields</i>	508
Adding a Lookup Table and Navigation—Step-by-Step	462	<i>Loading Database Fields into a List Box</i>	508
<i>Modify the User Interface</i>	462	<i>An Example with Find, Filter, and Sort</i>	508
<i>Change the Properties</i>	463	The Data Environment Designer	512
<i>Write the Code</i>	464	<i>Connection and Command Objects</i>	513
<i>Testing the Navigation and Lookup Tables</i>	466	<i>Adding a Data Environment Designer</i>	513
Updating a Database File	466	<i>Creating Connections</i>	514
<i>Adding Records</i>	466	<i>Adding Commands</i>	514
<i>Deleting Records</i>	467	<i>Creating a Data Environment—Step-by-Step</i>	515
Preventing Errors	468	<i>Navigating Recordsets for Data Environment Objects</i>	517
<i>Protecting an Add Operation</i>	468	<i>One-to-Many Relationships</i>	518
Programming Hints	478	<i>Relation Hierarchies</i>	519
		<i>Creating a Relation Hierarchy—Step-by-Step</i>	519
		<i>Build the Relation Hierarchy</i>	520

12

Data Handling—Grids, Validation, Selection, and Sorting

Displaying Data in Grids	486
<i>A Grid Control—Step-by-Step</i>	486
<i>Displaying the Record Number and Record Count</i>	491
<i>The MoveComplete Event</i>	492

13

Drag-and-Drop

Drag-and-Drop Terminology	532
<i>The Source and the Target</i>	532
<i>Source Object Properties</i>	532
<i>DragOver and DragDrop Events</i>	532
A Step-by-Step Example	533
<i>Create the User Interface</i>	533
<i>Set the Properties</i>	534
<i>Write the Code</i>	534