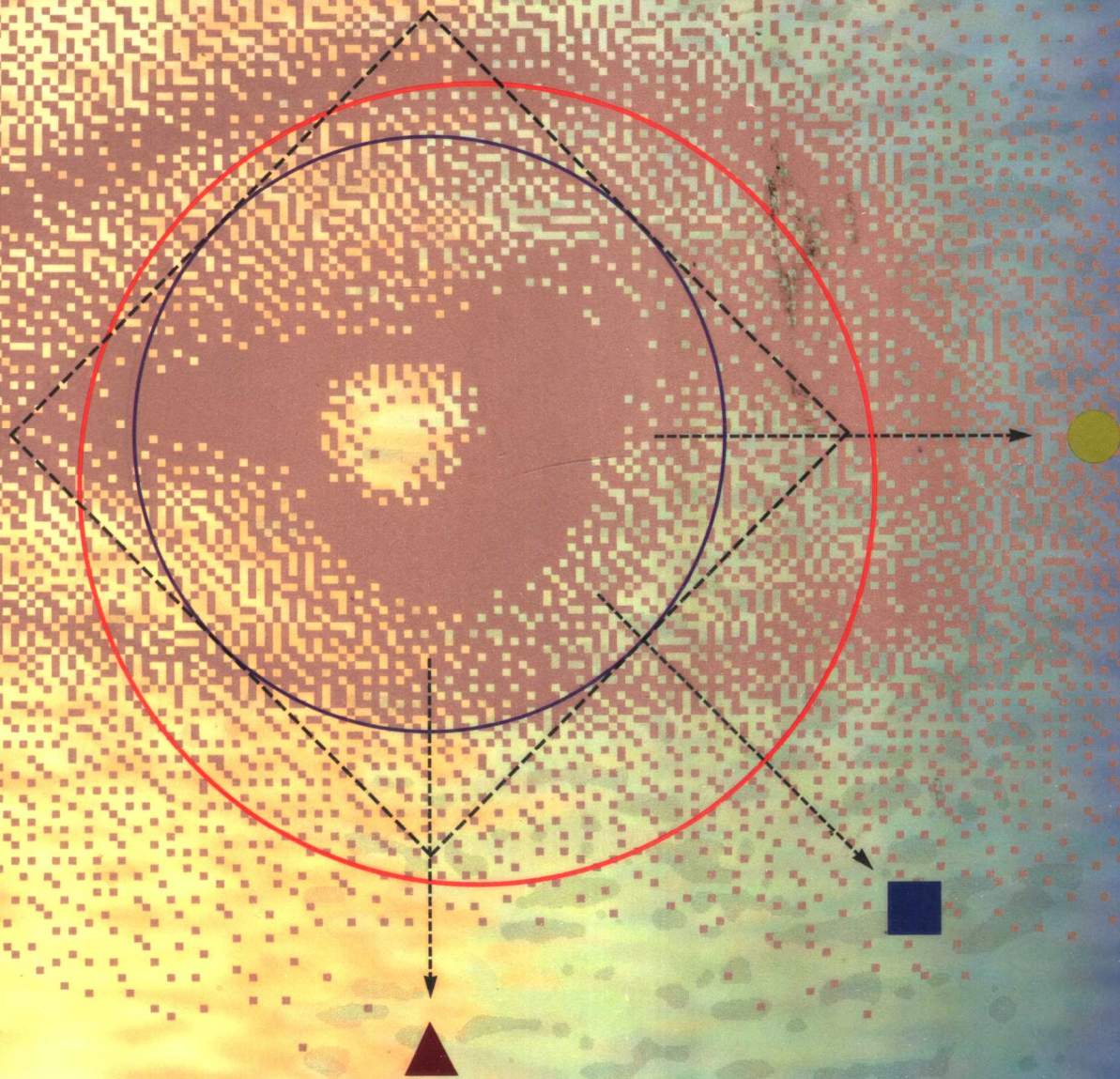


# Programming in Visual Basic

Version 4.0



JULIA CASE BRADLEY  
ANITA C. MILLSPAUGH



01

# **Programming in Visual Basic**

## **Version 4.0**

JULIA CASE BRADLEY  
ANITA C. MILLSPAUGH

**IRWIN**



Chicago • Bogotá • Boston • Buenos Aires • Caracas  
London • Madrid • Mexico City • Sydney • Toronto

© Richard D. Irwin, a Times Mirror Higher Education Group, Inc. company, 1996

*All rights reserved.* No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

**Irwin Book Team**

Publisher: *Tom Casson*  
Associate editor: *Garrett Glanz*  
Editorial assistant: *Carrie Berkshire*  
Project editor: *Karen M. Smith*  
Production supervisor: *Dina L. Genovese*  
Assistant manager, graphics: *Charlene R. Perez*  
Designer: *Crispin Prebys*  
Compositor: *Douglas & Gayle, Ltd.*  
Typeface: *11/13 Bodo*  
Printer: *Von Hoffmann Press, Inc.*

 **Times Mirror**  
 **Higher Education Group**

**Library of Congress Cataloging-in-Publication Data**

Bradley, Julia Case.  
Programming in Visual Basic, version 4.0 / Julia Case Bradley,  
Anita C. Millspough  
p. cm.  
Includes index.  
ISBN 0-697-20273-9  
1. BASIC (Computer program language) 2. Microsoft Visual BASIC.  
I. Millspough, A. C. (Anita C.) II. Title.  
QA76.73.B3B723 1996  
005.26'2—dc20

95-42175

*Printed in the United States of America*  
3 4 5 6 7 8 9 0 V H 3 2 1 0 9 8 7 6

# PREFACE

As the world turns to graphical user interfaces, computer programming languages are changing to accommodate the shift. Visual Basic 4.0 has been 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.

It is assumed that the student is familiar with the Windows operating environment.

## APPROACH

The authors have successfully taught Visual Basic to beginning and advanced programming students for several years. This text is designed to incorporate the basic concepts of programming, problem solving, programming logic, as well as the design techniques of an event-driven language.

The student will learn to deal with a visual interface while acquiring the skills of programming such as loops, decisions, and data management.

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 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.

## CHAPTER ORGANIZATION

Each chapter begins with identifiable objectives and a brief overview. There are numerous coding examples as well as a hands-on project with guidance for the coding. Thought-provoking feedback questions are interspersed throughout the chapter to allow the students time to reflect on the current topic and their understanding of the details. The end of the chapter items include a chapter review, questions, programming exercises, and two case studies. The case studies provide a continuing theme exercise that may be used throughout the course.

The first chapter 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.

Chapter 2 continues coverage of controls, including option buttons, check boxes, frames, and images. 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 a naming convention 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. Control arrays and message boxes are also covered. Chapter 5 provides the programmer with the knowledge for creating sub functions and sub procedures. The Case structure is provided as an alternative for the If statement. Menus and multiple forms are first discussed 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 grids in chapter 8, and to writing the information to disk in chapter 9. Chapter 9 covers both sequential and random files, but it may be covered in portions.

Chapters 10 and 11 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 11 includes writing queries in SQL.

The drag-and-drop feature of Windows programming is introduced in chapter 12. This chapter normally brings great enthusiasm from students as they learn to deal with the source and target objects. The chapter examples and assignments provide a blend of practical and just-for-fun applications. This is also true of chapter 13, which provides an introduction to the graphics methods and graphic controls.

Chapter 14, the final chapter, covers a variety of topics that bridge from Visual Basic into other applications. These include the Windows API, DLLs, OLE, and Visual Basic for Applications.

#### 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 greatly influenced the manuscript. A special thank you goes to Theresa Berry for her thorough technical review, constructive criticism, and many valuable suggestions. Thanks to Tricia for her work on the index. And most important, we are grateful to Dennis, Andy, and Eric 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 example projects, you will be on your way to writing Windows applications. Just reading the example projects is like trying to learn a foreign language or mathematics by reading alone. Enter the projects, look up your questions in Visual Basic's excellent Help, 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:

```
Open "FileName" For {Input|Output|Append|Random} As #FileNumber [Len=Reclength]
```

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

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

# CONTENTS

---

<b>1</b>	<b>INTRODUCTION TO VISUAL BASIC</b>	<b>1</b>	Define the User Interface	11
	<b>WRITING WINDOWS APPLICATIONS WITH VISUAL BASIC</b>	<b>2</b>	Set Properties	15
	The Windows Graphical User Interface	2	Write Code	18
	<b>PROGRAMMING LANGUAGES—PROCEDURAL, OBJECT ORIENTED, AND EVENT DRIVEN</b>	<b>3</b>	Visual Basic Code Statements	19
	The Object Model	3	Code the Event Procedures for Hello World	21
	Versions of Visual Basic	4	Run the Project	23
	<b>WRITING VISUAL BASIC PROJECTS</b>	<b>4</b>	Save the Project	24
	The Three-Step Process	4	Open the Project	26
	Planning a Visual Basic Project	5	Modify the Project	27
	Visual Basic Projects	5	Print the Project Documentation	32
	<b>THE VISUAL BASIC ENVIRONMENT</b>	<b>6</b>	<b>FINDING AND FIXING ERRORS</b>	<b>35</b>
	The Form Window	7	Compile Errors	35
	The Project Window	7	Run-Time Errors	35
	The Properties Window	7	Logic Errors	35
	The Toolbox	7	Project Debugging	35
	The Main Visual Basic Window	7	Naming Conventions	36
	The Toolbar	8	<b>VISUAL BASIC HELP</b>	<b>36</b>
	The Form Location and Size Information	8		
	Help	8	<b>2</b>	
	Design Time, Run Time, and Break Time	8	<b>MORE CONTROLS</b>	<b>43</b>
	<b>WRITING YOUR FIRST VISUAL BASIC PROJECT</b>	<b>8</b>	<b>INTRODUCING MORE CONTROLS</b>	<b>44</b>
	Set Up Your Visual Basic Workspace	9	Text Boxes	45
			Frames	45
			Check Boxes	46
			Option Buttons	46
			Images	46
			Giving a Three-Dimensional Appearance	48
			<b>WORKING WITH MULTIPLE CONTROLS</b>	<b>48</b>
			Selecting Multiple Controls	48
			Deselecting a Group of Controls	49
			Moving Controls as a Group	49



Setting Properties for Multiple Controls	50	<b>CALCULATIONS</b>	<b>80</b>
Aligning Controls	50	Arithmetic Operators	81
		Using Calculations in Code	82
<b>SETTING PROPERTIES FOR USER CONVENIENCE</b>	<b>51</b>	<b>COUNTING AND ACCUMULATING SUMS</b>	<b>83</b>
Defining Keyboard Access Keys	51	<b>VAL FUNCTION</b>	<b>84</b>
Setting the Default and Cancel Properties of Command Buttons	52	<b>FORMATTING DATA</b>	<b>84</b>
Setting the Tab Order for Controls	52	Creating Custom Formats	86
		Defining String Formats	86
		Defining Numeric Formats	87
<b>CODING FOR THE CONTROLS</b>	<b>53</b>	<b>PROGRAMMING HINTS</b>	<b>91</b>
Clearing Text Boxes and Labels	53		
Resetting the Focus	53	<b>4</b>	
Setting the Value Property of Option Buttons and Check Boxes	54	<b>DECISIONS AND CONDITIONS</b>	<b>99</b>
Changing the Font Properties of Controls	55		
Changing the Color of Text	55	<b>IF STATEMENTS</b>	<b>100</b>
Changing Multiple Properties of a Control	55	Flowcharting If Statements	102
Concatenating Text	56	<b>CONDITIONS</b>	<b>102</b>
Continuing Long Program Lines	56	Comparing Numeric Variables and Constants	103
Using the Value of Controls	57	Comparing Strings	104
		Comparing the Text Property of Text Boxes	106
		Comparing Uppercase and Lowercase Characters	107
		Compound Conditions	107
<b>PROGRAMMING HINTS</b>	<b>62</b>	<b>USING IF STATEMENTS WITH OPTION BUTTONS AND CHECK BOXES</b>	<b>108</b>
		<b>NESTED IF STATEMENTS</b>	<b>108</b>
<b>3</b>		<b>CONTROL ARRAYS</b>	<b>111</b>
<b>VARIABLES, CONSTANTS, AND CALCULATIONS</b>	<b>71</b>	<b>FINDING THE HIGHEST OR LOWEST VALUE</b>	<b>113</b>
		<b>INPUT VALIDATION</b>	<b>114</b>
<b>DATA—VARIABLES AND CONSTANTS</b>	<b>72</b>	Checking the Data Type	114
Data Types	73	Checking for a Range of Values	115
Naming Rules	74	<b>MESSAGE BOXES</b>	<b>115</b>
Naming Conventions	74	The Message String	116
Constants—Named and Intrinsic	75		
Declaring Variables	77		
Scope of Variables	79		

<b>PROGRAMMING HINTS</b>	<b>122</b>	Adding and Removing	
Testing for True or False	122	Forms	162
Debugging Visual Basic		The Hide and Show	
Projects	122	Methods	163
Pausing Execution with the		The Load and Unload	
Break Button	123	Statements	164
Forcing a Break	123	Referring to Objects on a	
Using the Debug Window	124	Different Form	164
Watch Expressions	125	<b>STANDARD CODE MODULES</b>	<b>165</b>
Instant Watch	126	<b>VARIABLES AND CONSTANTS</b>	
Stepping through Code	126	<b>IN MULTIPLE FORM PROJECTS</b>	<b>165</b>
		Global Variables and	
		Constants	166
		Static Variables	166
		Guidelines for Declaring	
		Variables and Constants	167
		<b>AN ABOUT BOX</b>	<b>169</b>
		<b>MENUS</b>	<b>169</b>
		Defining Menus	170
		Creating a Menu—	
		Step-by-Step	172
		Modifying a Menu	174
		Coding for Menu Commands	174
		Toggling Check Marks On	
		and Off	175
		Standards for Windows	
		Menus	175
		<b>COMMON DIALOG BOXES</b>	<b>175</b>
		Using a Common Dialog	
		Box	176
		Using the Information from	
		the Dialog Box	177
		Setting Current Values	179
		<b>PROGRAMMING HINTS</b>	<b>190</b>
		Setting the Startup Form	190
		Adding an Existing Form to	
		a Project	190
		Working with Maximized	
		Forms	191
<b>5</b>			
<b>SUB PROCEDURES</b>			
<b>AND FUNCTIONS</b>	<b>133</b>		
<b>THE SHAPE CONTROL</b>	<b>134</b>		
<b>THE LINE CONTROL</b>	<b>135</b>		
<b>THE SCROLL BAR CONTROLS</b>	<b>135</b>		
Scroll Bar Events	136		
<b>THE CASE STRUCTURE</b>	<b>140</b>		
Testing Option Buttons with			
the Case Structure	142		
<b>FUNCTIONS FOR FINANCIAL</b>			
<b>CALCULATIONS</b>	<b>144</b>		
<b>WRITING GENERAL</b>			
<b>PROCEDURES</b>	<b>146</b>		
Passing Variables to			
Procedures	148		
Function Procedures versus			
Sub Procedures	148		
<b>PROGRAMMING HINTS</b>	<b>153</b>		
Creating Executable Files	153		
<b>6</b>			
<b>MULTIPLE</b>			
<b>FORMS AND</b>			
<b>MENUS</b>	<b>161</b>		
<b>MULTIPLE FORMS</b>	<b>162</b>		
Creating New Forms	162		

## 7 LISTS, LOOPS, AND PRINTING 199

### LIST BOXES AND COMBO BOXES 200

Filling the List	201
The ListIndex Property	203
The ListCount Property	203
The List Property	203

### FOR/NEXT LOOPS 204

Negative Increment or Counting Backward	206
Conditions Satisfied before Entry	206
Altering the Values of the Loop Control Variables	207
Endless Loops	207
Exiting For/Next Loops	207

### SENDING INFORMATION TO THE PRINTER 209

Printing to the Printer	209
Formatting Lines	209
Selecting the Font	212
Terminating the Page or the Job	212

### PROGRAMMING HINTS 217

## 8 ARRAYS AND GRIDS 225

### SINGLE-DIMENSION ARRAYS 226

Subscripts	226
More on Subscripts	227

### FOR EACH/NEXT STATEMENTS 228

Initializing an Array Using For Each	228
---	-----

### USER-DEFINED DATA TYPES 229

Accessing Information with User-Defined Data Types	230
---	-----

Using Array Elements for Accumulators	231
Table Lookup	232
Coding a Table Lookup	233
Coding a Table Lookup with Error Validation	235

### USING LIST BOXES WITH ARRAYS 235

The ItemData Property	236
Using Nonsequential ItemData values	237
Adding Items with ItemData to a List	237

### MULTIDIMENSIONAL ARRAYS 238

Initializing Two- Dimensional Arrays	239
Printing a Two-Dimensional Table	239
Summing a Two- Dimensional Table	240
Lookup Operation for Two- Dimensional Tables	240

### THE GRID CONTROL 242

Appearance	243
Contents	244

### PROGRAMMING HINTS 249

Filling a Grid Control	249
The Array Function	249

## 9 DATA FILES 257

### DATA FILES 258

Data Files and Project Files	258
Data File Terminology	258
File Organizations	259
Opening and Closing Data Files	259
Viewing the Data in a File	260

### SEQUENTIAL FILE ORGANIZATION 261

Writing Data to a Sequential Disk File	261
---	-----

Creating a Sequential Data File 262

Reading the Data in a Sequential File 263

Finding the End of a Data File 263

**TRAPPING PROGRAM ERRORS 264**

The On Error Statement 265

The Err Object 266

Coding Error-Handling Routines 267

The Resume Statement 269

The Exit Function and Exit Sub Statements 270

**DO/LOOPS 271**

**SEQUENTIAL FILE PROGRAMMING EXAMPLE 274**

**USING THE MSGBOX FUNCTION 276**

Function Return Values 277

Specifying the Buttons and/or Icons to Display 277

**RANDOM DATA FILES 277**

Fixed-Length Strings 278

Defining a Record for a Random File 278

Opening a Random File 278

Reading and Writing a Random File 279

Accessing Fields in a Random Record 280

Finding the End of a Random File 280

The Seek Function 281

**UPDATING A RANDOM FILE 281**

Adding and Editing Records 281

Deleting a Record 282

The Read and Write Procedures 283

**PROGRAMMING HINTS 291**

The FreeFile Function 291

The Input Box Function 291

**10 ACCESSING DATABASE FILES 297**

**VISUAL BASIC AND DATABASE FILES 298**

Database Formats Supported by Visual Basic 298

Database Terminology 298

Creating Database Files for Use By Visual Basic 299

**USING THE DATA CONTROL 299**

Data Control and Data-Bound Controls 300

**VIEWING A DATABASE FILE—STEP-BY-STEP 303**

Design and Create the Form 303

Set the Properties for the Data Control 304

Set the Properties for the Data-Bound Controls 304

Write the Code 305

Run the Project 306

**NAVIGATING THE DATABASE IN CODE 307**

The Recordset Object 307

Using the MoveNext, MovePrevious, MoveFirst, and MoveLast Methods 307

Checking for BOF and EOF 308

**USING LIST BOXES AND COMBO BOXES AS DATA-BOUND CONTROLS 308**

Setting Up a Lookup Table for a Field 308

**ADDING A LOOKUP TABLE AND NAVIGATION—STEP-BY-STEP 309**

Modify the User Interface 309

Change the Properties 309

Write the Code 311

<b>UPDATING A DATABASE FILE</b>	<b>312</b>	<b>CREATING A NEW DYNASET</b>	<b>345</b>
Adding Records	312	Using SQL	345
Deleting Records	313	Setting a New RecordSource	345
<b>PREVENTING ERRORS</b>	<b>313</b>	Check for an Empty Recordset	347
The Validate Event	314	<b>PROGRAMMING HINTS</b>	<b>356</b>
<b>PROGRAMMING HINTS</b>	<b>321</b>	Trapping Database Errors	356
		Advanced Database Processing	358
		Use Microsoft Access to Help Code SQL Statements	358
<b>11</b>		<b>12</b>	
<b>DATABASE VALIDATION, SELECTION, AND SQL</b>	<b>327</b>	<b>DRAG-AND-DROP</b>	<b>363</b>
<b>VALIDATION STRATEGIES</b>	<b>328</b>	<b>DRAG-AND-DROP TERMINOLOGY</b>	<b>364</b>
Selectively Enable and Disable Command Buttons	328	The Source and the Target	364
Check the Current Operation	329	Source Object Properties	364
Check the DataChanged Property of Data-Bound Controls	330	DragOver and DragDrop Events	364
Trap Errors with On Error	331	<b>A STEP-BY-STEP EXAMPLE</b>	<b>365</b>
Trap Errors with the Database Error Event	332	Create the User Interface	365
Add a Cancel Button	334	Set the Properties	366
Select the Entry in a Text Box	335	Write the Code	366
<b>PROGRAMMING EXAMPLE SHOWING VALIDATION TECHNIQUES</b>	<b>335</b>	Run the Project	368
<b>RECORDSETS</b>	<b>339</b>	Add Command Buttons	369
Searching for a Specific Record	340	Run the Completed Project	370
The NoMatch Property	341	<b>DRAGGING AND DROPPING MULTIPLE OBJECTS</b>	<b>370</b>
Bookmarks	341	Passing the Source Argument	371
Seeking with Table Indexes	342	Changing the Icon of the Target Image	372
<b>REORDERING A TABLE</b>	<b>343</b>	Setting the DragIcon Property of Source Controls	372
Referring to Database Fields	344	Blanking Out an Image	373
Loading Database Fields into a List Box	344	<b>PROGRAMMING HINTS</b>	<b>381</b>
		Manual Drag-and-Drop	381

13  
**GRAPHICS 385**

**THE GRAPHICS ENVIRONMENT 386**  
 The Coordinate System 386  
 PictureBox Controls 387

**COLORS 387**  
 The RGB Function 387  
 The QBColor Function 388

**THE GRAPHICS METHODS 388**  
 The CLS Method 389  
 The PSet Method 389  
 The Line Method 390  
 The Circle Method 392  
 The Step Keyword 394

**LAYERING 395**

**MORE PROPERTIES FOR YOUR GRAPHICS CONTROLS 396**  
 Design Time 397  
 Controlling Pictures at Run Time 397  
 Moving a Picture 397

**SIMPLE ANIMATION 398**

**THE TIMER CONTROL 399**

**PROGRAMMING HINTS 401**  
 Custom Coordinate Systems 401  
 PaintPicture Method 402

14  
**OLE, DLLS, VBA, AND THIRD-PARTY CONTROLS 407**

**OBJECT LINKING AND EMBEDDING 408**  
 Object Linking 408

Object Embedding 408  
 Creating OLE Objects at Run Time 411

**DYNAMIC LINK LIBRARIES 411**  
 The Declare Statement 412  
 Passing Arguments ByVal and ByRef 412  
 Calling a DLL Procedure 413  
 Accessing System Information with a DLL 413  
 Placing Tabs for Columns in a List Box 414

**USING THIRD-PARTY CONTROLS 416**

**VISUAL BASIC FOR APPLICATIONS 416**  
 Recording an Excel Macro 417  
 A Sample VBA Application 418  
 Help with Visual Basic in Excel 420

**CREATING YOUR OWN OBJECTS 420**

**APPENDIX ANSWERS TO FEEDBACK QUESTIONS 423**

**GLOSSARY 433**

**INDEX 439**

# INTRODUCTION TO VISUAL BASIC

---

**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 *event-driven programming*.
3. Explain the concepts of objects, properties, and methods.
4. List and describe the three steps for writing a Visual Basic project.
5. Describe the various files that make up a Visual Basic project.
6. Identify the elements in the Visual Basic environment.
7. Explain the differences among design time, run time, and break time.
8. Write, run, save, print, and modify your first Visual Basic project.
9. Identify compile errors, run-time errors, and logic errors.
10. Look up Visual Basic topics in Help.

---

## CHAPTER

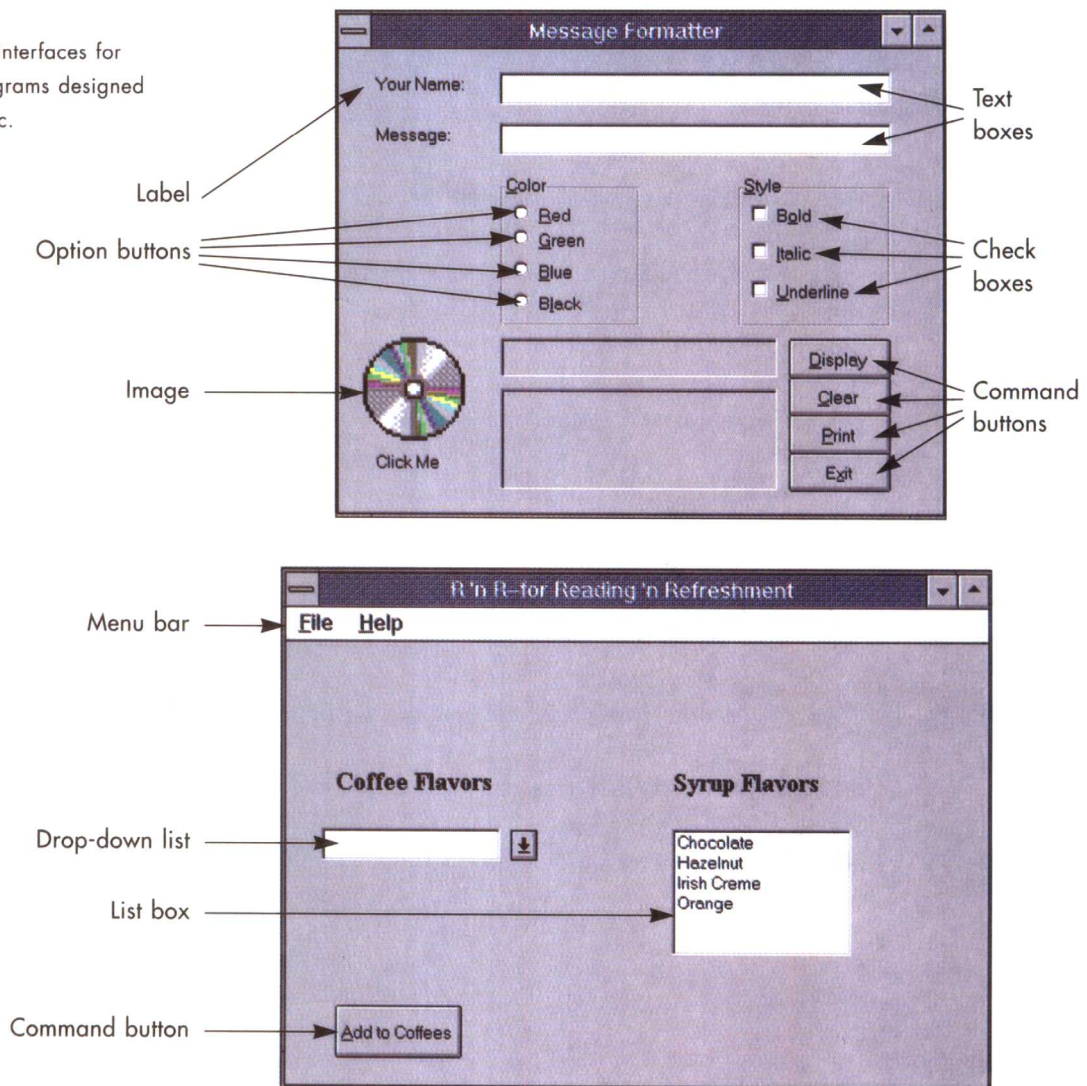
**1**

## WRITING WINDOWS APPLICATIONS WITH VISUAL BASIC

Using this text, you will learn to write computer programs that run in the Microsoft Windows environment. Your projects will look and act like standard Windows programs. Visual Basic (VB) provides the tools you need to create windows with familiar elements like menus, text boxes, command buttons, option buttons, check boxes, list boxes, and scroll bars. Figure 1.1 shows some sample Windows user interfaces.

**Figure 1.1**

Graphical user interfaces for application programs designed with Visual Basic.



### THE WINDOWS GRAPHICAL USER INTERFACE

Microsoft Windows uses a **graphical user interface**, or **GUI** (pronounced *gooy*). The Windows GUI defines how the various elements look and function. As a Visual Basic programmer, you have available a toolbox of these elements. You will create new windows, called **forms**.



Then you will use the toolbox to add the various elements, called **controls**. The projects that you will write follow a relatively new type of programming, called *event-driven programming*.

## PROGRAMMING LANGUAGES — PROCEDURAL, OBJECT ORIENTED, AND EVENT DRIVEN

There are literally hundreds of programming languages. Each was developed to solve a particular type of problem. Most traditional languages, such as BASIC, C, COBOL, FORTRAN, PL/I, and Pascal are considered *procedural* languages. That is, the program specifies the sequence of all operations, step-by-step. Program logic determines the next instruction to execute, in response to conditions and user requests.

The newer programming languages, such as C++ and Visual Basic, use a different approach: **object-oriented programming** (OOP) and **event-driven programming**. Microsoft refers to Visual Basic as an event-driven programming language, which has many (but not all) elements of an object-oriented language such as C++. Each release of Visual Basic moves it a little closer to a true object-oriented language.

In the event-driven model, programs are no longer procedural; they do not follow a sequential logic. You, as the programmer, do not take control and determine the sequence of execution. Instead, the user can press keys and click on various buttons and boxes in a window. Each user action can cause an *event* to occur, which triggers a Basic procedure that you have written. For example, the user clicks on a command button labeled Calculate. The clicking causes the button's click event to occur, and the program automatically jumps to a procedure you have written to do the calculation.

### THE OBJECT MODEL

In Visual Basic you will work with **objects**, which have **properties** and **methods**.

#### OBJECTS

Think of an object as a thing, or a noun. Examples of objects are forms and controls. Forms are the windows and dialog boxes you place on the screen; controls are the elements you place inside a form, such as text boxes, command buttons, and list boxes.

#### PROPERTIES

Properties tell something about an object, such as its name, color, size, location, or how it will behave. You can think of properties as adjectives that describe objects.

When you refer to a property, you first name the object, then a period, then name the property. For example, refer to the Caption property of a form called Form1 as Form1.Caption (say "form1 dot caption").