Preface

*Live in fragments no longer. Only connect.*
Edward Morgan Forster

Welcome to Visual Basic 6 and the exciting world of Internet and World Wide Web programming! This book is by an old guy and two young guys. The old guy [HMD; Massachusetts Institute of Technology (MIT) 1967] has been programming and/or teaching programming for 37 years. The two young guys [PJD; MIT 1991 and TRN; MIT 1992] have each been programming for 15 years and have caught the teaching and writing “bug.” The old guy programs and teaches from experience; the young guys do so from an inexhaustible reserve of energy. The old guy wants clarity; the young guys want performance. The old guy seeks elegance and beauty; the young guys want results. We got together to produce a book we hope you will find informative, interesting, challenging and entertaining.

**Why We Wrote Visual Basic 6 How to Program**

Dr. Harvey M. Deitel taught introductory programming courses in universities for 20 years with an emphasis on developing clearly written, well-structured programs. Much of what is taught in these courses is the basic principles of programming with an emphasis on the effective use of control structures, primitive data types, arrays, functions and the program development process.

We present these topics in *Visual Basic 6 How to Program* exactly the way HMD has done in his university courses. Our experience has been that students handle the material in the early chapters on primitive data types, control structures, procedures and arrays in about the same manner as they handle introductory Pascal or C courses. There is one noticeable difference though: students are highly motivated by the fact that they are learning a leading-edge language (Visual Basic 6) and leading-edge programming paradigms (event-
driven programming and object-oriented programming) that will be immediately useful to them as they leave the university environment. This increases their enthusiasm for the material. Students quickly discover that they can do great things with Visual Basic 6, so they are willing to put incredible effort into their courses.

Our goal was clear: produce a Visual Basic 6 textbook for introductory university-level courses in computer programming for students with little or no programming experience, yet offer the depth and the rigorous treatment of theory and practice demanded by traditional, upper-level C and C++ courses and that satisfies professionals’ needs. To meet these goals, we produced a comprehensive book because our text also patiently teaches the principles of control structures, event-driven programming, object-oriented programming and the Visual Basic 6 language.

Visual Basic 6 How to Program was written fresh on the heels of C How to Program: Second Edition, C++ How to Program: Second Edition and Java How to Program: Second Edition, each of which has become the world’s leading introductory textbook in its respective field. Hundreds of thousands of students and professional people worldwide have learned C, C++ and Java from these texts. Upon publication in November, 1998 Visual Basic 6 How to Program will be used in universities, professional schools and corporate training programs worldwide.

We have prepared an interactive multimedia CD-ROM edition of this book—the Visual Basic 6 Multimedia Cyber Classroom. Prentice Hall offers a “value pack” edition of both Visual Basic 6 How to Program and the Visual Basic 6 Multimedia Cyber Classroom called The Complete Visual Basic 6 Training Course at a discount for people who want to use both the book and the multimedia CD. Please see the last few pages of this book for ordering instructions. The Complete Visual Basic 6 Training Course is discussed in more detail later in this Preface.

We believe in Visual Basic 6. Its conceptualization by Microsoft®, the creators of Visual Basic 6, is brilliant: Base a language on one of the world’s most widely known languages, Basic. Endow the language with the ability to conveniently build applications for Microsoft Windows®—the world’s most widely used platform. Provide the kinds of heavy duty, high-performance capabilities needed for enterprise systems development. Make the language appropriate for implementing Internet-based and World-Wide-Web-based applications, and build in the features people really need such as Strings, graphics, graphical user interface components, error handling, multimedia (audio, images, animation and video), file processing, database processing, Internet-based client/server networking, World Wide Web browsing, World Wide Web document enhancement with Visual Basic Script (VBScript), and prepackaged components. Make the language extensible so that independent software vendors (ISVs) can provide componentry for a vast array of application arenas. These features are precisely what businesses and organizations need to meet today’s information processing requirements.

Visual Basic 6 empowers programmers to unleash their creativity. Once Visual Basic 6 students enter lab, instructors will not be able to hold the students back. They will be eager to experiment and explore. They will quickly produce applications that go well beyond anything they would have produced in introductory programming courses in procedural languages like C, Pascal and non-visual versions of Basic.
The computer field has never seen anything like the Internet/World Wide Web “explosion” occurring today. People want to communicate. People need to communicate. Sure they have been doing that since the dawn of civilization, but computer communications have been mostly limited to digits, alphabetic characters and special characters passing back and forth. The next major wave is surely multimedia. People want to transmit pictures and they want those pictures to be in color. They want to transmit voices, sounds and audio clips. They want to transmit full-motion color video. And at some point, they will insist on three-dimensional, moving-image transmission. Our current flat, two-dimensional televisions will eventually be replaced with three-dimensional versions that turn our living rooms into “theaters in the round.” Actors will perform their roles as if we were watching live theater. Our living rooms will be turned into miniature sports stadiums. Our business offices will enable video conferencing among colleagues half a world apart as if they were sitting around one conference table. The possibilities are intriguing and Visual Basic 6 is sure to play a key role in making many of these possibilities become reality.

There have been predictions that the Internet will eventually replace the telephone system. Well, why stop there? It could also replace radio and television as we know them today. It’s not hard to imagine the Internet replacing the newspaper with completely electronic news media. This textbook you are reading may someday appear in a museum alongside radios, TVs and newspapers in an “early media of ancient civilization” exhibit.

Teaching Approach

*Visual Basic 6 How to Program* contains a rich collection of examples, exercises, and projects drawn from many fields to provide the student with a chance to solve interesting real-world problems. The book concentrates on the principles of good software engineering and stresses program clarity. We avoid arcane terminology and syntax specifications in favor of teaching by example. Each of our code examples has been carefully tested.

This book is written by three educators who spend most of their time teaching edge-of-the-practice topics in industry classrooms worldwide. The text emphasizes pedagogy.

**Live-Code Teaching Approach**

The book is loaded with live-code examples. This is the focus of the way we teach and write about programming, and the focus of each of our multimedia Cyber Classrooms as well. Virtually every new concept is presented in the context of a complete, working Visual Basic 6 program immediately followed by one or more windows showing the program’s output. We call this style of teaching and writing our live-code approach. *We use the language to teach the language.* Reading these programs is much like entering and running them on a computer.

**World Wide Web Access**

All of the code for *Visual Basic 6 How to Program* is on the Internet at the Prentice Hall Web site [http://www.prenhall.com/deitel](http://www.prenhall.com/deitel) and at the Deitel & Associates, Inc. Web site [http://www.deitel.com](http://www.deitel.com). Please download all the code then run each program as you read the text. Make changes to the code examples and see what happens. See how the Visual Basic 6 compiler “complains” when you make various kinds of errors. Im-
Immediately see the effects of making changes to the code. It’s a great way to learn programming by doing programming. [You must respect the fact that this is copyrighted material. Feel free to use it as you study Visual Basic 6, but you may not republish any portion of it without explicit permission from the authors and Prentice Hall.]

Objectives
Each chapter begins with a statement of Objectives. This tells the student what to expect and gives the student an opportunity, after reading the chapter, to determine if he or she has met these objectives. It is a confidence builder and a source of positive reinforcement.

Quotations
The learning objectives are followed by quotations. Some are humorous, some are philosophical and some offer interesting insights. Our students enjoy relating the quotations to the chapter material. The quotations are worth a “second look” after you read each chapter.

Outline
The chapter Outline helps the student approach the material in top-down fashion. This, too, helps students anticipate what is to come and set a comfortable and effective learning pace.

7806 lines of code in 178 Example Programs (with Program Outputs)
We present Visual Basic 6 features in the context of complete, working Visual Basic 6 programs. This is the focus of our teaching and our writing. We call it our “live-code” approach. Each program is followed by a window with the output produced when the program runs. This enables the student to confirm that the programs run as expected. Reading the book carefully is much like entering and running these programs on a computer. The programs range from just a few lines of code to substantial examples with several hundred lines of code. Students should download all the code for the book from our Web sites and run each program while studying that program in the text. The programs are available at both http://www.deitel.com and http://www.prenhall.com/deitel.

558 Illustrations/Figures
An abundance of charts, line drawings and program outputs is included. The discussion of control structures, for example, features carefully drawn flowcharts. [Note: We do not teach the use of flowcharting as a program development tool, but we do use a brief, flowchart-oriented presentation to specify the precise operation of Visual Basic 6’s control structures.]

412 Programming Tips
We include programming tips to help students focus on important aspects of program development. We highlight hundreds of these tips in the form of Good Programming Practices, Common Programming Errors, Look-and-Feel Observations, Testing and Debugging Tips, Performance Tips, Portability Tips, and Software Engineering Observations. These tips and practices represent the best we have gleaned from a combined six decades of programming and teaching experience. One of our students—a mathematics major—told us recently that she feels this approach is like the highlighting of axioms, the-
orems and corollaries in mathematics books; it provides a basis on which to build good software.

78 Good Programming Practices
When we teach introductory courses, we state that the “buzzword” of each course is “clarity,” and we highlight as Good Programming Practices techniques for writing programs that are clearer, more understandable, more debuggable and more maintainable.

155 Common Programming Errors
Students learning a language tend to make certain errors frequently. Focusing the students’ attention on these Common Programming Errors helps students avoid making the same errors. It also helps reduce the long lines outside instructors’ offices during office hours!

14 Look-and-Feel Observations
We provide Look-and-Feel Observations to highlight Windows graphical user interface conventions. These observations help students design their applications to “look” and “feel” like typical Windows programs.

39 Performance Tips
In our experience, teaching students to write clear and understandable programs is by far the most important goal for a first programming course. But students want to write the programs that run the fastest, use the least memory, require the smallest number of keystrokes, or dazzle in other nifty ways. Students really care about performance. They want to know what they can do to “turbo charge” their programs, so we have included extensive Performance Tips.

15 Portability Tips
We include Portability Tips to help students write portable code, i.e., code that is designed to run on a variety of platforms. These tips highlight issues that may affect how Visual Basic 6 programs run on Windows systems with different hardware devices. We also consider internationalization issues.

76 Software Engineering Observations
Event-driven programming and object-oriented programming require a complete rethinking about the way we build software systems. Visual Basic 6 is an effective language for performing good software engineering. The Software Engineering Observations highlight architectural and design issues that affect the construction of software systems, especially large-scale systems. Much of what the student learns here will be useful in upper-level courses and in industry as the student begins to work with large, complex real-world systems.

40 Testing and Debugging Tips
When we first designed this “tip type,” we thought we would use it strictly to tell people how to test and debug Visual Basic 6 programs. In fact, many of the tips de-
scribe aspects of Visual Basic 6 that reduce the likelihood of “bugs” and thus simplify the testing and debugging process for Visual Basic 6 programs.

**Summary**
Each chapter ends with additional pedagogical devices. We present a thorough, bullet-list-style Summary of the chapter. On average, there are 38 summary bullets per chapter. This helps the students review and reinforce key concepts.

**Terminology**
We include in a Terminology section an alphabetized list of the important terms defined in the chapter—again, further reinforcement. On average, there are 94 terms per chapter.

**Summary of Tips, Practices, and Errors**

**406 Self-Review Exercises and Answers (Count Includes Separate Parts)**
Extensive self-review exercises and answers are included for self study. This gives the student a chance to build confidence with the material and prepare for the regular exercises. Students should be encouraged to do all the self-review exercises and to check their answers.

**819 Exercises (Solutions in Instructor’s Manual; Count Includes Separate Parts)**
Each chapter concludes with a substantial set of exercises including simple recall of important terminology and concepts; writing individual Visual Basic 6 statements; writing small portions of Visual Basic 6 procedures and modules; writing complete Visual Basic 6 procedures, classes, applications; and writing major term projects. The large number of exercises across a wide variety of areas enables instructors to tailor their courses to the unique needs of their audiences and to vary course assignments each semester. Instructors can use these exercises to form homework assignments, short quizzes, and major examinations. The solutions for the exercises are included in the Instructor’s Manual and on the disks available only to instructors through their Prentice-Hall representatives. [NOTE: Please do not write to us requesting the instructor’s manual. Distribution of this publication is strictly limited to college professors teaching from the book. Instructors may obtain the solutions manual only from their regular Prentice Hall representatives.] Solutions to many of the exercises are included on the Visual Basic 6 Multimedia Cyber Classroom CD; please see the last few pages of this book for ordering instructions [or call 1-800-811-0912 and ask for ISBN# 0-13-083116-6].

**Approximately 5150 Index Entries (with approximately 7400 Page References)**
We have included an extensive Index at the back of the book. This helps the student find any term or concept by keyword. The Index is useful to people reading the book for the first time and is especially useful to practicing programmers who use the book as a reference.
Most of the 2078 terms in the Terminology sections appear in the Index (along with many more index items from each chapter). Students can use the Index in conjunction with the Terminology sections to be sure they have covered the key material of each chapter.

**Bibliography**

A bibliography is included to encourage further reading. Excellent additional bibliographic information on Visual Basic 6 is available on the World Wide Web. The bibliography includes a summary of the Visual Basic 6 on-line documentation that comes with the Standard, Professional and Enterprise editions of Visual Basic 6 (but which, unfortunately, does not come with the Working Model Edition software included with this book).

**Additional Resources for Instructors and Students**

We have prepared an Adobe Acrobat PDF-format slide show including all the illustrations, tables, program listings and program output windows in *Visual Basic 6 How to Program*. We will post regular announcements on our web sites

http://www.deitel.com  
http://www.prenhall.com/deitel

about additional instructor and student resources, including errata, FAQs, test banks, syllabi and other forthcoming resources.

**A Tour of the Book**

**Chapter 1—Computers Concepts**—discusses what computers are, how they work and how they are programmed. It introduces the notion of structured programming and explains why this set of techniques has fostered a revolution in the way programs are written. The chapter gives a brief history of the development of programming languages from machine languages to assembly languages to high-level languages. The origin of the Visual Basic 6 programming language is discussed.

**Chapter 2—Integrated Development Environment**—overviews the Visual Basic 6 Integrated Development Environment (IDE) which is used to create Windows programs. The IDE is a graphical environment containing an editor for writing Visual Basic 6 code, a compiler, and a debugger. The Toolbox, Properties window, Project window, form window, tool bars and menus are discussed. This chapter familiarizes students with the IDE. Students are also introduced to the concept of visual programming—the ability to create programs by writing little code and sometimes no code at all.

**Chapter 3—Introduction to Visual Basic Programming**—gives a concise introduction to writing Visual Basic 6 programs. Detailed discussions of event-driven programming, decision making, arithmetic operations and operator precedence are presented. After studying this chapter, the student will understand how to write simple, but complete, Visual Basic 6 programs.

**Chapter 4—Control Structures: Part I**—focuses on the program development process. The chapter discusses how to take a problem statement (i.e., a requirements document) and from it develop a working Visual Basic 6 program, including performing intermediate steps in pseudocode. The chapter introduces some fundamental data types and simple control structures used for decision making (*If/Then* and *If/Then/Else*) and
repetition (While Do While and Do Until). The chapter uses simple flowcharts to show the flow of control through each of the control structures. We examine counter-controlled repetition and sentinel-controlled repetition. We have had a positive experience assigning problems 4.16 through 4.19 in our introductory courses. Since these four problems have similar structure, doing all four is a nice way for students to “get the hang of” the program development process. Students will enjoy the challenges of the “mystery programs.” The more mathematically inclined students will enjoy problems on encryption and decryption and determining how many digits in a five-digit number are 7s.

Chapter 5—Control Structures: Part II—continues the discussion of Visual Basic 6’s sequence, selection (Select Case) and repetition (For/Next, Do/Loop While and Do/Loop Until) control structures. This chapter also uses simple flowcharts to show the flow of control through each of the control structures. The techniques discussed in Chapter 4 and Chapter 5 constitute a large part of what has been traditionally taught in the universities under the topic of structured programming. Chapter 5 examines repetition in detail and compares the alternatives of counter-controlled loops and sentinel-controlled loops. This chapter helps the student develop good programming habits in preparation for dealing with the more substantial programming tasks in the remainder of the text. The chapter also introduces logical operators—And (logical AND), Or (logical OR), and Not (logical NOT), constants, and Visual Basic 6 data types. There is a substantial exercise set including mathematical and graphical applications. The more mathematically inclined students will enjoy problems on calculating the trigonometric sine of a number with an infinite series, Pythagorean triples and De Morgan’s Laws. Our students particularly enjoy the challenges of triangle-printing and diamond-printing in Exercises 5.16, 5.17 and 5.21; these problems really help students learn to program nested repetition structures—a complex topic to master in introductory courses.

Chapter 6—Sub Procedures and Function Procedures—takes a deeper look at procedures. We explore procedures in depth and include a discussion of procedures that “call themselves,” so-called recursive procedures. We discuss Visual Basic 6 procedures, programmer-defined procedures and recursion. The techniques presented in Chapter 6 are essential to the production of properly structured programs, especially the kinds of larger programs and software that application programmers are likely to develop in real-world applications. The “divide and conquer” strategy is presented as an effective means for solving complex problems by dividing them into simpler interacting components. Students enjoy the treatment of random numbers and simulation, and they appreciate the discussion of the dice game of craps that makes elegant use of control structures (this is one of our most successful lectures in our introductory courses). The chapter offers a solid introduction to recursion. Some texts leave recursion for a chapter late in the book; we feel this topic is best covered gradually throughout the text. The extensive collection of exercises at the end of the chapter includes several classical recursion problems such as the Towers of Hanoi; the reader may want to revisit this problem later in the text and employ graphics, animation and sound to make the problem “come alive.” There are many mathematical and graphical examples. Our students particularly enjoy the development of a “Computer-Assisted Instruction” system in Exercises 6.19 through 6.21.
Chapter 7—Arrays—explores the processing of data in lists and tables of values. We discuss the structuring of data into arrays, or groups, of related data items of the same type. The chapter presents numerous examples of both one-dimensional arrays and two-dimensional arrays. It is widely recognized that structuring data properly is just as important as using control structures effectively in the development of properly structured programs. Examples in the chapter investigate various common array manipulations, printing histograms, sorting data, passing arrays to procedures and an introduction to the field of survey data analysis (with simple statistics). A feature of this chapter is the discussion of elementary sorting and searching techniques and the presentation of binary searching as a dramatic improvement over linear searching. The 32 end-of-chapter exercises include a variety of interesting and challenging problems such as improved sorting techniques, the design of an airline reservations system, an introduction to the concept of turtle graphics (made famous in the LOGO programming language) and the Knight’s Tour and Eight Queens problems that introduce the notions of heuristic programming so widely employed in the field of artificial intelligence. The exercises contain a series of recursion problems including the selection sort, linear search, binary search and quick sort.

Chapter 8—Strings, Dates, and Times—deals with processing words, sentences, characters, groups of characters, dates and times. For each example we provide extensive live-code examples demonstrating appropriate Visual Basic 6 functions “in action.” We show output windows so the reader can see the precise effects of each of the string and character manipulations we discuss. Students will enjoy the card shuffling and dealing example (which they will enhance in the exercises). A key feature of the chapter is an extensive collection of challenging string-manipulation exercises related to limericks, text analysis, word processing, printing dates in various formats, check protection, writing the word equivalent of a check amount, Morse Code, card shuffling and dealing, encryption, hangman, cryptograms and metric-to-English conversions. Students will enjoy the challenges of developing their own spell checker and crossword puzzle generator.

Advanced Topics

Chapter 9—Graphics—begins a run of chapters that present the multimedia “sizzle” of Visual Basic 6. We consider Chapters 9 through 21 to be the book’s advanced material. Professors who have been teaching C and/or C++ at the introductory level will find that the Visual Basic 6 topics open all kinds of opportunities to explore new areas. This is “fun stuff.” We discuss drawing graphical shapes (e.g., lines, circles, etc.), manipulating colors, images, and printing. The chapter illustrates each of these graphics capabilities with live-code examples, appealing screen outputs, detailed features tables and detailed line art. The exercises challenge students to draw various geometric shapes, manipulate images and create a graphical version of the game of hangman.

Chapter 10—Basic Graphical User Interface Concepts—introduces the creation of applications with user-friendly graphical user interfaces (GUIs). GUI development is a huge topic, so we divided it into two chapters. These chapters cover the material in depth to enable you to build “industrial-strength” GUI interfaces. Through its programs, tables and line drawings, the chapter illustrates GUI design principles, ListBox controls, scrollbars, Sliders, menus, pop-up menus, and
MsgBoxes. Some of the exercises challenge the student to use various controls to format inputs, create GUIs and sort stored strings to facilitate searching. The chapter exercises include a delightful simulation of the classic race between the tortoise and the hare.

A special section entitled “Building Your Own Computer” explains machine-language programming and proceeds with the design and implementation of a computer simulator that allows the reader to write and run machine-language programs. This unique feature of the text will be especially useful to the reader who wants to understand how computers really work. Our students enjoy this project and often implement substantial enhancements; several enhancements are suggested in the exercises.

Chapter 11—Advanced Graphical User Interface Concepts—continues the detailed walkthrough of Visual Basic 6’s GUI capabilities. The examples walk the reader through the manipulation of more advanced GUI controls including RichTextBox,UpDown, ImageList, ImageCombo, and FlatScrollBar. Single document interface (SDI) and multiple document interfaces (MDI) are discussed. The chapter concludes with a discussion of native-code compilation. The exercises encourage the reader to develop more substantial GUIs with the advanced components presented in the chapter.

Chapter 12—Mouse and Keyboard—discusses handling mouse and keyboard events. Topics discussed include drag-and-drop, changing the mouse-pointer shape, distinguishing which mouse button was pressed, intercepting key events (from the keyboard) and distinguishing which keyboard key was pressed. The chapter provides several nice examples—including a drag-and-drop example that allows the user to move a Knight chess piece around a chess board.

Chapter 13—Error Handling and Debugging—is one of the most important chapters in the book from the standpoint of building so-called “mission-critical” or “business-critical” applications that require high degrees of robustness and fault tolerance. Things do go wrong, and at today’s computer speeds—commonly hundreds of millions of operations per second—if they can go wrong they will, and rather quickly at that. Programmers are often a bit naive about using components. They ask, “How do I request that a component do something for me?” They also ask “What value(s) does that component return to me to indicate it has performed the job I asked it to do?” But programmers also need to be concerned with, “What happens when the component I call on to do a job experiences difficulty? How will that component signal that it has a problem?” In Visual Basic 6, when a component (i.e., a procedure) encounters difficulty, it can “raise an error.” The environment of that component is programmed to “handle” that error and deal with it. To use a Visual Basic 6 component, you need to know not only how that component behaves when “things go well,” but also what errors that component raises when “things go poorly.” The chapter discusses the vocabulary of error handling. The On Error block executes program code that may execute properly or may raise an error if something goes wrong. Associated with an On Error block are one or more handlers that process raised errors and attempt to restore order and keep systems “up and running” rather than letting them “crash.” Even if order can not be fully restored, the handlers may perform operations that enable a system to continue executing, albeit at reduced levels of performance—such activity is often referred to as “graceful degradation.”
The chapter also introduces the reader to the Visual Basic 6 debugger. We discuss the *Debug* menu and debug tool bar commands (used to control the debugging session), the *Locals* window (used to display local variable values), the *Watch* window (used to view both local and module variables) and the *Call Stack* window (used to view a list of all the procedures that were called in order to reach the selected line).

**Chapter 14—Sequential File Processing**—deals with input/output that is accomplished through streams of data directed to and from files. This is one of the most important chapters for programmers who will be developing commercial applications. The chapter introduces the data hierarchy from bits, to bytes, to fields, to records, to files. Next, Visual Basic 6’s simple view of files and streams is presented. We show how programs pass data to secondary storage devices like disks and how programs retrieve data already stored on those devices. Sequential-access files are discussed using a series of programs that show how to open and close files, how to store data sequentially in a file and how to read data sequentially from a file. This chapter uses the new *FileSystem* object model of Visual Basic 6 which allows files, directories and drives to be conveniently manipulated. We also discuss controls for graphically manipulating files, directories and drives.

**Chapter 15—Records and Random-Access Files**—Random-access files are discussed using a series of programs that show how to sequentially create a file for random access, how to read and write data to a file with random access, and how to read data sequentially from a randomly accessed file. The last random-access program combines many of the techniques of accessing files both sequentially and randomly into a complete transaction-processing program. We discuss the notion of User-Defined Types (UDTs)—Visual Basic’s equivalent of records. We explain how these records can be output to, and input from, secondary storage devices. Students in our industry seminars have told us that after studying the material on file processing, they were able to produce substantial file-processing programs that were immediately useful to their organizations. The exercises ask the student to implement a variety of programs that build and process random-access files.

**Chapter 16—Object-Oriented Programming**—begins our discussion of class modules. The chapter represents a wonderful opportunity for teaching data abstraction the “right way.” The chapter focuses on the essence and terminology of classes and objects. What is an object? What is a class of objects? What does the inside of an object look like? How are objects created? How are they destroyed? How do objects communicate with one another? Why are classes such a natural mechanism for packaging software as reusable componentry? The chapter discusses implementing abstract data types (ADTs) as Visual Basic 6 classes, accessing class members, enforcing information hiding with *Private* instance variables, separating interface from implementation, using access methods and utility methods, and initializing objects. The chapter discusses composition—the process of building classes that have references to objects as members, dynamic memory allocation, interface inheritance, and delegation. The chapter compares inheritance (“is a” relationships) with composition (“has a” relationships). A feature of the chapter is its several substantial case studies. In particular, a lengthy case study implements a point, circle and cylinder class hierarchy. Another lengthy case study implements an employee, boss, commission worker, piece worker and hourly worker class hierarchy.
The chapter explains polymorphic behavior. In object-oriented languages in general, when many subclasses are related through inheritance to a common superclass, each subclass object may be treated as a superclass object. This enables programs to be written in a general manner independent of the specific types of the subclass objects. New kinds of objects can be handled by the same program, thus making systems more extensible. Polymorphism enables programs to eliminate complex Select Case logic in favor of simpler “straight-line” logic. A screen manager of a video game, for example, can simply send a “draw” message to every object in a linked list of objects to be drawn. Each object knows how to draw itself. A new type of object can be added to the program without modifying that program as long as that new object also knows how to draw itself. This style of programming is typically used to implement today’s popular GUls. Visual Basic 6 does not support conventional implementation inheritance (as Java and C++ do, for example), but does allow polymorphism through interface inheritance. The chapter introduces interfaces—sets of methods that must be defined by any class that Implements the interfaces. Interfaces are useful for providing a basic set of methods to classes throughout the hierarchy. A feature of the chapter is its major polymorphism case studies.

The chapter exercises ask the student to discuss several conceptual issues and approaches, work with interfaces, modify the chapter’s employee class—and pursue all these projects with polymorphic programming. The exercises also ask the student to compare the creation of new classes by inheritance vs. composition; to extend the inheritance hierarchies discussed in the chapter; to write an inheritance hierarchy for quadrilaterals, trapezoids, parallelograms, rectangles and squares; and to create an automobile hierarchy. Additional exercises challenge the student to develop classes for complex numbers, rational numbers, times, dates, Tic-Tac-Toe, a savings account and sets of integers.

Chapter 17—ActiveX—introduces the student to Microsoft’s broad range of distributed computing technologies called ActiveX. The chapter discusses the foundations of ActiveX—Microsoft’s component object model (COM), Microsoft’s distributed component model (DCOM), and object linking and embedding (OLE).

In previous chapters, the student becomes familiar with using ActiveX controls to create GUls and to perform other tasks. This chapter carefully demonstrates how to create ActiveX controls. We discuss the types of ActiveX controls that can be created using Visual Basic 6. We also discuss the key events in the lifetime of an ActiveX control.

Two key examples in the chapter create a LabelScrollbar control and digital Clock control. The LabelScrollbar combines a Label control and a scrollbar control. The digital Clock control draws the time on the control every second with the assistance of a Timer control. Each of these controls is written from “scratch.”

We also introduce the ActiveX Control Interface Wizard for rapidly creating simple controls and for creating the foundation of more complex controls. We provide a step-by-step example that uses the ActiveX Control Interface Wizard to aid in the creation of the same LabelScrollbar created earlier in the chapter.

Special windows, called property pages, that contain a list of properties are also discussed. Many ActiveX controls provide property pages to allow the developer to conveniently set groups of properties. We demonstrate how to create property pages for the LabelScrollbar control using the Property Page Wizard.
The chapter also demonstrates how a Visual Basic 6 program can control Microsoft Word through a process called automation. In one example, we create a Word document without explicitly opening Word. In another example we use Word’s spelling checker to determine if a word is spelled correctly.

The exercises also ask the reader to create several controls as well as to enhance both the LabelScrollbar and Clock controls.

Chapter 18—Database Management—discusses how Visual Basic 6 can be used to access relational databases. Visual Basic 6 provides a variety of database-aware controls for displaying and manipulating database data. In this chapter we primarily focus on the Microsoft ADO Data control and the Microsoft DataGrid control. The examples in the chapter use the Biblio.mdb sample database provided with Visual Basic 6. For accessing the Biblio.mdb database, we provide an overview of Structured Query Language (SQL). We discuss the Hierarchical FlexGrid control, DataList control and DataCombo controls. We also discuss the Data Environment Designer for rapidly designing and implementing data-driven Visual Basic 6 applications.

Chapter 19—Networking, the Internet and the World Wide Web—deals with programs that can communicate over computer networks. What is a client? What is a server? How do clients ask servers to perform their services? How do servers give results back to clients? What is a URL (uniform resource locator)? How can a program load World Wide Web pages? How can I use Visual Basic 6 to develop collaborative applications?

This chapter gives you what you need to begin implementing client/server networked Visual Basic 6 programs immediately. We show how to write programs that “walk the Web.” We discuss manipulating URLs, using a URL stream connection to read a file on a server, establishing simple clients and simple servers using stream sockets, client/server interaction with stream sockets and connectionless client/server interaction with datagrams.

The chapter concludes with a discussion of Visual Basic Scripting (VBScript) for enhancing World Wide Web documents. VBScript is a subset of Visual Basic 6. The chapter has a nice collection of exercises including several suggested modifications to the server examples.

If this is your first Visual Basic 6 book and you are an experienced computing professional, you may well be thinking, “Hey, this just keeps getting better and better. I can’t wait to get started programming in this language. It will let me do all kinds of stuff I’d like to do, but that was never easy for me to do with the other languages I’ve used.” You’ve got it right. Visual Basic 6 is an enabler. So if you liked the ActiveX and networking discussions, hold onto your hat, because Visual Basic 6 will also let you program multimedia applications.

Chapter 20—Multimedia: Images, Animation and Audio—deals with Visual Basic 6’s capabilities for making computer applications come alive. It is remarkable that students in first programming courses will be writing applications with all these capabilities. The possibilities are intriguing. Imagine having access (over the Internet and through CD-ROM technology) to vast libraries of graphics images, audios and videos and being able to weave your own together with those in the libraries to form creative applications. Already more than half the new computers sold come “multimedia equipped.” Within just a few years,
new machines equipped for multimedia will be as common as machines with floppy disks today. We can’t wait to see the kinds of term papers and classroom presentations students will be making when they have access to vast public domain libraries of images, drawings, voices, pictures, videos, animations and the like.

A “paper” when most of us were in the earlier grades was a collection of characters, possibly handwritten, possibly typewritten. A “paper” in just a few short years will become a multimedia “extravaganza” that makes the subject matter come alive. It will hold your interest, pique your curiosity, make you feel what the subjects of the paper felt when they were making history. Multimedia will make your science labs much more exciting. Textbooks will come alive. Instead of looking at a static picture of some phenomenon, you will watch that phenomenon occur in a colorful, animated, presentation with sounds, videos and various other effects. It will leverage the learning process. People will be able to learn more, learn it in more depth and experience more viewpoints.

The chapter discusses images and image manipulation, audios and animation. A feature of this chapter is an example that uses the Microsoft Agent control which allows the user to choose one of three predefined characters—Merlin, Genie and Robby the Robot. These characters actually speak and move (e.g., Robby the Robot removes his head and fixes himself with a screwdriver). Another key example of this chapter is the Multimedia MCI control—which we use to create a CD player and an AVI file player. We demonstrate the RealAudio control for receiving streaming audio and video from the Internet. We also demonstrate the Marquee control for scrolling an image from a specified URL. The chapter concludes with a discussion of the ActiveMovie control which enables an application to play video and sound in many different formats (i.e., MPEG, AVI, WAV, etc.).

Once you have read the chapter, you will be eager to try out these techniques, so we have included extensive exercises to challenge and entertain you; many of these make interesting term projects:

<table>
<thead>
<tr>
<th>Multimedia exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog Clock</td>
</tr>
<tr>
<td>Artist</td>
</tr>
<tr>
<td>Automated Teller Machine</td>
</tr>
<tr>
<td>Background Audio</td>
</tr>
<tr>
<td>Calendar/Tickler File</td>
</tr>
<tr>
<td>Calling Attention to an Image</td>
</tr>
<tr>
<td>Digital Clock</td>
</tr>
<tr>
<td>Audio/Graphical Kaleidoscope</td>
</tr>
</tbody>
</table>

You are going to have a great time attacking some of these problems! Some will take a few hours and some are great term projects. We see all kinds of opportunities for multimedia electives starting to appear in the university computing curriculum. You might compete with your classmates to develop the best solutions to several of these problems.
Chapter 21—Data Structures, Collections, Dictionaries—is particularly valuable in second- and third-level university courses. The chapter discusses the techniques used to create and manipulate dynamic data structures such as linked lists, stacks, queues (i.e., waiting lines) and trees. The chapter begins with discussions of self-referential classes and dynamic memory allocation. We proceed with a discussion of how to create and maintain various dynamic data structures. For each type of data structure, we present live-code programs and show sample outputs. The **Collection** object—an array-like object that can dynamically grow and shrink—is also discussed. We also discuss the **Dictionary** object for storing key/value pairs.

One problem when working with references is that students may have trouble visualizing the data structures and how their nodes are linked together. So we have included illustrations that show the links and the sequence in which they are created. The binary tree example is a nice capstone for the study of references and dynamic data structures. This example creates a binary tree; enforces duplicate elimination; and introduces recursive pre-order, inorder and postorder tree traversals. Students have a genuine sense of accomplishment when they study and implement this example. They particularly appreciate seeing that the inorder traversal prints the node values in sorted order. The chapter exercises include a supermarket simulation using queueing, recursively searching a list, recursively printing a list backwards, binary tree deletion, level-order traversal of a binary tree, printing trees, inserting/deleting anywhere in a linked list, analyzing the performance of binary tree searching and sorting and implementing an indexed list class.

**Appendix A—Operator Precedence Chart**—lists each of the Visual Basic 6 operators and indicates their relative precedence. We list each operator and its full name on a separate line.

**Appendix B—ANSI Character Set**—lists the characters of the ANSI (American National Standards Institute) character set and indicates the character code value for each. ANSI includes ASCII (American Standard Code for Information Interchange) as a subset.

**Appendix C—Visual Basic Internet and World Wide Web Resources**—presents the best resources Abbey Deitel was able to track down for you on the Web. This is a great way for you to get into the “world of Visual Basic.” The appendix lists various Visual Basic resources such as consortia, journals and companies that make various key Visual Basic-related products. The following chart summarizes the resources you will find by visiting the URLs Abbey recommends.

<table>
<thead>
<tr>
<th>add-ins</th>
<th>examples and demos</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveX controls</td>
<td>FAQs</td>
</tr>
<tr>
<td>books</td>
<td>games</td>
</tr>
<tr>
<td><strong>Carl and Gary’s Visual Basic Home Page</strong></td>
<td><strong>Inside Visual Basic</strong></td>
</tr>
<tr>
<td>COM</td>
<td>jobs</td>
</tr>
<tr>
<td><strong>comp.lang.basic.visual</strong></td>
<td>links</td>
</tr>
<tr>
<td>newsgroups</td>
<td>mailing lists</td>
</tr>
<tr>
<td>custom controls</td>
<td></td>
</tr>
<tr>
<td><strong>Developer.com</strong></td>
<td></td>
</tr>
<tr>
<td>DCOM</td>
<td></td>
</tr>
<tr>
<td>COM+</td>
<td></td>
</tr>
<tr>
<td>conferences</td>
<td></td>
</tr>
<tr>
<td>DirectX</td>
<td></td>
</tr>
</tbody>
</table>

**Add-ins**: add-ins examples and demos

**ActiveX controls**: FAQs games

**Books**: links mailing lists

**COM**: inside Visual Basic

**Custom controls**: jobs

**Developer.com**: links

**DCOM**: mailing lists

**COM+**: Microsoft COM home page

**Conferences**: Microsoft Developer Network (MSDN)

**DirectX**: Microsoft Scripting Technologies VBScript site

**Microsoft Transaction Server**:

**Microsoft.public.vb**: newsgroups
Appendix D—Number Systems—discusses the binary (base 2), decimal (base 10), octal (base 8) and hexadecimal (base 16) number systems. This material is valuable for introductory courses in computer science and computer engineering. The appendix is presented with the same pedagogic learnings aids as the chapters of the book. A nice feature of the appendix is its 33 exercises, 19 of which are self-review exercises with answers.

The Complete Visual Basic 6 Training Course

We have implemented an interactive, CD-ROM-based, software version of Visual Basic 6 How to Program called The Visual Basic 6 Multimedia Cyber Classroom. It is loaded with features for learning and reference. The Cyber Classroom is normally wrapped with Visual Basic 6 How to Program in the product called The Complete Visual Basic 6 Training Course. If you have already purchased the textbook, you can get a copy of the Cyber Classroom CD directly from Prentice Hall. Please call 1-800-811-0912 and ask for ISBN# 0-13-083116-6.

The Visual Basic 6 Multimedia Cyber Classroom contains an introductory presentation in which we overview the Cyber Classroom features. The live-code Visual Basic 6 example programs in the textbook truly “come alive” in the Cyber Classroom. We have placed executables for all these example programs “under the hood” of the Cyber Classroom, so if you are viewing a program and want to execute it, you simply click the lightning bolt icon and the program executes. You immediately see—and hear for the audio-based multimedia programs—the program’s outputs. If you want to modify a program and see and hear the effects of your changes, simply click the floppy-disk icon that causes the source code to be “lifted off” the CD and “dropped into” one of your own directories so you can edit the program, recompile the program and try out your new version. Click the audio icon and Tem Nieto or Paul Deitel will talk about the program and “walk you through” the code. We provided line numbers in the source code listings to help you follow the audio discussions.
The Cyber Classroom contains a set of interactive, self-review questions that reinforce your understanding of the material. These questions are provided at the end of most sections. Also, solutions to many of the end-of-chapter exercises are provided. Please note that we cannot provide complete solutions to all the exercises as this product is a college textbook and the exercises are used by professors for homework assignments and exams. For professors, the Instructor’s Manual provides a list of the solved exercises provided with the Cyber Classroom. Again, please do not ask us for the Instructor’s Manual. It is only available to college instructors teaching from the book. It can only be obtained by faculty members from Prentice Hall field representatives.

The Visual Basic 6 Multimedia Cyber Classroom also provides many navigational aids including extensive hyperlinking. The Cyber Classroom remembers in a “history list” recent sections you have visited and allows you to move forward or backward in that history list. The thousands of index entries are hyperlinked to their text occurrences. Using the full-text-search feature, you can type a term and the Cyber Classroom will locate the occurrences of the term throughout the text. The Table of Contents entries are “hot,” so clicking a chapter name immediately takes you to that chapter. The Contents menu provides direct access to all the sections in the current chapter, every chapter, every appendix, the Bibliography and the Index. Many people like to browse through our programming tips—Common Programming Errors, Good Programming Practices, Look-and-Feel Observations, Performance Tips, Portability Tips, Software Engineering Observations and Testing-and-Debugging Tips. The tips are all accessible via icons at the bottom of each chapter page.

Students and professional users of our Multimedia Cyber Classrooms tell us they like the interactivity and that the Cyber Classrooms are effective reference tools because of the extensive hyperlinking and other navigational features. We recently had an email from a person who said that he lives “in the boonies” and cannot take a live course at a university, so the Cyber Classroom was the solution to his educational needs.

Professors have sent us emails indicating their students enjoy using the Cyber Classrooms, spend more time on the courses and master more of the material than in textbook-only courses. Also, the Cyber Classroom helps shrink lines outside professors’ offices during office hours. Our other Cyber Classrooms include the C & C++ Multimedia Cyber Classroom: Second Edition, the Java Multimedia Cyber Classroom: Second Edition and plan to publish Cyber Classroom editions of our forthcoming Visual J++ 6 How to Program and Visual C++ 6 How to Program textbooks.

Visual Basic 6.0 Working Model Edition Software

This book contains a CD-ROM that includes Microsoft’s Visual Basic 6.0 Working Model Edition and the examples from the book. The CD also contains an HTML web page that you can load into your World Wide Web browser. The page contains links to the Deitel & Associates, Inc. web site, the Prentice Hall web site and the web sites listed in Appendix C.

The Visual Basic 6.0 Working Model Edition allows the reader to create and execute Visual Basic 6 programs. Although the Working Model Edition does not support every Visual Basic 6 feature presented in this book, it does support the vast majority of the features discussed. Some features covered in this book are specific to the Professional Edition or the Enterprise Edition.
We carefully tested every one of the 178 examples in the book using the Visual Basic 6.0 Working Model Edition and have compiled for you in the following table a list of the 7 programs that cannot be created/executed using this software and notes on six other examples that will run but may require special handling:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Figure</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 17</td>
<td>17.2</td>
<td>Requires Microsoft Word to be installed with Service Release 1 (SR-1).</td>
</tr>
<tr>
<td></td>
<td>17.15, 17.33, 17.34, 17.45</td>
<td>These work correctly, but may issue various error messages when the projects are loaded into the Visual Basic 6.0 Working Model. Simply dismiss the dialogs that are displayed.</td>
</tr>
<tr>
<td></td>
<td>17.47 and 17.50</td>
<td>Visual Basic 6.0 Working Model does not allow the programmer to create ActiveX DLLs.</td>
</tr>
<tr>
<td></td>
<td>17.52 and 17.54</td>
<td>The ActiveX features of these examples are only available in the Professional and Enterprise editions of Visual Basic.</td>
</tr>
<tr>
<td></td>
<td>17.55</td>
<td>The ActiveX features of this example are only available in the Professional and Enterprise editions of Visual Basic.</td>
</tr>
<tr>
<td>Chapter 18</td>
<td>18.19</td>
<td>The Data Environment Designer is only provided with the Professional and Enterprise editions of Visual Basic.</td>
</tr>
<tr>
<td>Chapter 20</td>
<td>20.1</td>
<td>Microsoft Agent must be installed before this program can execute. Refer to the discussion in the text for more information on downloading Microsoft Agent.</td>
</tr>
<tr>
<td></td>
<td>20.9</td>
<td>The Animation control is only provided with the Professional and Enterprise editions of Visual Basic.</td>
</tr>
</tbody>
</table>

The following are the minimum system requirements to use the Visual Basic 6 Working Model Edition:

- A personal computer with a 486DX/66 processor. A Pentium 90 or higher microprocessor is recommended.
- Microsoft Windows 95, Windows 98 or Windows NT® 4.0 with service pack 3 (or later) installed.
- The minimum system memory for Windows 95, Windows 98 or Windows NT 4.0 is 24 MB. Microsoft recommends 32 MB or higher for all versions of Windows.
- The hard-disk space required is 52 MB for the typical installation and 65 MB for the maximum installation.
- Microsoft Internet Explorer™ 4.01 with Service Pack 1 (included). The additional hard disk space required for Microsoft Internet Explorer is 43 MB for the typical installation and 59 MB for the maximum installation.
- A CD-ROM drive.
- A monitor with VGA or higher-resolution (Super VGA recommended).
- Microsoft Mouse or a compatible pointing device.
Installing the Visual Basic 6 Working Model Edition
On most Windows 95, Windows 98 and Windows NT 4.0 computers, the Visual Basic 6 Working Model Edition software begins its installation program automatically when the CD-ROM is inserted, so you can simply answer the questions asked by the installation program. In the event that the installation program does not execute automatically, run the Setup.exe program from the Visual Basic 6 Working Model Edition CD-ROM, then answer the questions asked by the installation program.

Documentation for the Visual Basic 6 Working Model Edition
Microsoft provides the full documentation for Visual Basic on the World Wide Web at the Microsoft Developer Network web site

http://msdn.microsoft.com/developer/default.htm

When you browse this site, click the link MSDN Library Online. If you have not already registered for the Microsoft Developer Network, you will be asked to register now. Registration is free and you can only access the documentation if you register first. After you register, go back to the Microsoft Developer Network web site home page and click MSDN Library Online. You should now see the MSDN Library Online page. The documentation is arranged hierarchically. You can find the Visual Basic documentation under

Microsoft Developer Network Library Online
  Developer Products
    Visual Studio 6.0
    Visual Basic 6.0

Acknowledgments
One of the great pleasures of writing a textbook is acknowledging the efforts of many people whose names may not appear on the cover, but whose hard work, cooperation, friendship, and understanding were crucial to the production of the book.

Three other people at Deitel & Associates, Inc. devoted long hours to this project. We would like to acknowledge the efforts of Chris Poirier, Barbara Deitel and Abbey Deitel.

Mr. Chris Poirier, a junior majoring in Computer Science at the University of Rhode Island, created many of the graphical illustrations presented in this book.

Barbara Deitel managed the preparation of the manuscript and coordinated with Prentice Hall the production of the book. Barbara’s efforts are by far the most painstaking of what we do to develop books. She has infinite patience. She handled the endless details involved in publishing a 1000-page, two-color book; a 500-page instructor’s manual and the 650 megabyte CD—Visual Basic 6 Multimedia Cyber Classroom. She used FrameMaker page-composition software to prepare the book. Barbara mastered this complex software package and did a marvelous job giving the book its clean style. She spent long hours researching the quotes at the beginning of each chapter. Barbara prepared every one of the 7800 page references in the 46-page index. She did all this in parallel with handling her extensive financial and administrative responsibilities at Deitel & Associates, Inc.

Abbey Deitel, a graduate of Carnegie Mellon University’s industrial management program, and now pursuing a management career with Deitel & Associates, Inc., wrote
Appendix C, “Visual Basic Internet and World Wide Web Resources” and suggested the title for the book. We asked Abbey to surf the World Wide Web and track down the best Visual Basic 6 sites. She used every major Web search engine and collected this information for you in Appendix C. For each resource and demo, Abbey has provided a brief explanation. She will be maintaining current versions of these resources and demo listings on our Web sites http://www.prenhall.com/deitel and http://www.deitel.com. She asks that you send URLs for your favorite sites to her by email at deitel@deitel.com and she will post links to these on our sites.

We are fortunate to have been able to work on this project with a talented and dedicated team of publishing professionals at Prentice Hall. This book happened because of the encouragement, enthusiasm, and persistence of our computer science editor, Laura Steele, and her boss—our mentor at Prentice Hall one of the best friends we’ve ever had in publishing—Marcia Horton, Editor-in-Chief of Prentice-Hall’s Engineering and Computer Science Division. Camille Trentacoste did a marvelous job as production manager.

The Visual Basic 6 Multimedia Cyber Classroom was developed in parallel with Visual Basic 6 How to Program. We sincerely appreciate the “new media” insight, savvy and technical expertise of our editor Mark Taub, Editor-in-Chief of Multimedia and Distance Learning and his able colleague, Karen McLean. Mark and Karen did a remarkable job bringing the Visual Basic 6 Multimedia Cyber Classroom, to publication under a tight schedule.

We owe special thanks to the creativity of Tamara Newnam Cavallo who did the art work for our programming tips icons and the cover. She created the delightful creature who shares with you the book’s programming tips. Please help us name this endearing little bug.

We sincerely appreciate the efforts of our reviewers:

Sean Alexander (Microsoft Corporation)
Dave Glowacki (Microsoft Corporation)
Phil Lee (Microsoft Corporation)
William Vaughn (Microsoft Corporation)
Scott Wiltamuth (Microsoft Corporation)
Mehdi Abedinejad (Softbank Marketing Services, Inc.)
David Bongiovanni (Bongiovanni Research & Technology, Inc.)
Rockford Lhotka

Under an especially tight time schedule, they scrutinized every aspect of the text and made countless suggestions for improving the accuracy and completeness of the presentation.

We would like to acknowledge the extraordinary efforts of Susanne Peterson, our mentor at Microsoft. All the nice things we said about Susanne on the book’s dedication page are absolutely true!

Our single most important source of information was Microsoft’s Visual Basic 6 Documentation from the Microsoft Developer Network (MSDN) Library for Visual Studio 6.0 which we worked from mostly during the beta releases of Visual Basic 6. We list the key portions of this documentation in the Bibliography at the end of this book. The single most
valuable source of information for us within this documentation set was the *Visual Basic 6 Programmer’s Guide*.

We would sincerely appreciate your comments, criticisms, corrections, and suggestions for improving the text. Please send your suggestions for improving and adding to our list of *Good Programming Practices*, *Common Programming Errors*, *Look-and-Feel Observations*, *Testing-and-Debugging Tips*, *Performance Tips*, *Portability Tips* and *Software Engineering Observations*. We will acknowledge all contributors in the next edition of our book. Please address all correspondence to our email address:

deitel@deitel.com

We will respond immediately. Well, that’s it for now. Welcome to the exciting world of Visual Basic 6 programming. We hope you enjoy this look at contemporary computer applications development using multimedia, databases, ActiveX, the Internet and the World Wide Web. Good luck!

*Dr. Harvey M. Deitel*  
*Paul J. Deitel*  
*Tem R. Nieto*

*Sudbury, Massachusetts*  
*October, 1998*

**About the Authors**

**Dr. Harvey M. Deitel**, CEO of Deitel & Associates, Inc., has 37 years experience in the computing field including extensive industry and academic experience. He is one of the world's leading computer science instructors and seminar presenters. Dr. Deitel earned B.S. and M.S. degrees from the Massachusetts Institute of Technology and a Ph.D. from Boston University. He worked on the pioneering virtual memory operating systems projects at IBM and MIT that developed techniques widely implemented today in systems like UNIX, Windows NT and OS/2. He has 20 years of college teaching experience including earning tenure and serving as the Chairman of the Computer Science Department at Boston College before founding Deitel & Associates, Inc. with Paul J. Deitel. He is author or co-author of several dozen books and multimedia packages and is currently writing many more. With translations published in Japanese, Russian, Spanish, Basic Chinese, Advanced Chinese, Korean, French, Polish and Portuguese. Dr. Deitel’s texts have earned international recognition. Dr. Deitel has delivered professional seminars internationally to major corporations, government organizations and various branches of the military.

**Paul J. Deitel**, Executive Vice President of Deitel & Associates, Inc., is a graduate of the Massachusetts Institute of Technology's Sloan School of Management where he studied Information Technology. Through Deitel & Associates, Inc. he has delivered Java, C and C++ courses for industry clients including Digital Equipment Corporation, Sun Microsystems, Rogue Wave Software, Software 2000, Computervision, Stratus, Fidelity, Cambridge Technology Partners, Open Environment Corporation, One Wave, Hyperion Software, Lucent Technologies, Adra Systems, Entergy, CableData Systems, NASA at the
Kennedy Space Center, the National Severe Storm Laboratory and IBM. He has lectured on C++ and Java for the Boston Chapter of the Association for Computing Machinery. He teaches satellite-based Java courses through a cooperative venture of Deitel & Associates, Inc., Prentice Hall and the Technology Education Network. He is the co-author of fifteen books and multimedia packages with Harvey Deitel and is currently writing five more.


**Tem R. Nieto**, Senior Instructor with Deitel & Associates, Inc., is a graduate of the Massachusetts Institute of Technology where he studied engineering and computing. Through Deitel & Associates, Inc. he has delivered courses for industry clients including Sun Microsystems, Digital Equipment Corporation, Compaq, Stratus, Fidelity, Art Technology, Progress Software, Toys “R” Us, Operational Support Facility of the National Oceanographic and Atmospheric Administration, Jet Propulsion Laboratory, Nynex, Motorola, Federal Reserve Bank of Chicago, Banyan, Schlumberger, University of Notre Dame, NASA, various military installations and many others.

**About Deitel & Associates, Inc.**

Deitel & Associates, Inc. is an internationally recognized corporate training and publishing organization specializing in programming languages, Internet/World Wide Web technology and object technology education. The company provides courses on Visual Basic, C++, Visual C++, Visual J++, C, Java, Internet and World Wide Web programming, and Object-Oriented Analysis and Design. The principals of Deitel & Associates, Inc. are Dr. Harvey M. Deitel and Paul J. Deitel. The company’s clients include some of the world’s largest computer companies, government agencies, branches of the military and business organizations. Through its publishing partnership with Prentice Hall, Deitel & Associates, Inc. publishes leading-edge programming textbooks, professional books, interactive, CD-ROM based multimedia *Cyber Classrooms*, satellite courses and World Wide Web courses. Deitel & Associates, Inc. and each of the authors can be reached via email at

```
deitel@deitel.com
```

To learn more about Deitel & Associates, Inc., its public seminar schedule and its worldwide on-site course curriculum, visit:

```
http://www.deitel.com
```

To learn more about Deitel & Deitel/Prentice Hall publications, visit:

```
http://www.prenhall.com/deitel
```
For a current list of Deitel/Prentice Hall publications including textbooks and multimedia packages, and for complete worldwide ordering information, please see the last few pages of this book.