Preface

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

Welcome to the Internet & World Wide Web! At Deitel & Associates, we write college-level programming-language textbooks and professional books. We work hard to keep our published books up-to-date with a steady flow of new editions. Writing *Internet & World Wide Web How to Program*, *Third Edition* was a joy. The Web increases the prominence of the Internet in information systems, strategic planning and implementation. Organizations want to integrate the Internet "seamlessly" into their information systems and the Web offers endless opportunity to do so.

This book and its support materials have everything instructors and students need for an informative, interesting, challenging and entertaining educational experience. We have tuned the writing, the pedagogy, our coding style and the book's ancillary package. Also, we have included a Tour of the Book in this Preface to help instructors, students and professionals get a sense of the rich coverage of Web technologies this book provides. We also provide a dependency chart to indicate prerequisites for each chapter in the book.

In this Preface, we overview the teaching conventions used in this book, such as syntax coloring the code examples, "code washing" and highlighting important code segments to help focus students' attention on key concepts introduced in each chapter. We also overview the new features of this edition.

We discuss the software Prentice Hall has bundled with the book as well as the comprehensive suite of educational materials that help instructors maximize their students' learning experience. These include an Instructor's Resource CD with solutions to the book's chapter exercises and a Test-Item File with hundreds of multiple-choice examination questions and answers. Additional instructor resources are available at the book's Companion Web Site (`www.prenhall.com/deitel`), which includes a Syllabus Manager and customizable PowerPoint® Lecture Notes. PowerPoint slides and additional support materials are available for students at the Companion Web Site, as well.

Internet & World Wide Web How to Program, Third Edition, was reviewed by a team of distinguished academics and industry professionals. We list their names and affiliations so you can get a sense of how carefully this book was scrutinized. The Preface concludes with information about the authors and about Deitel & Associates, Inc. As you read this book, if you have any questions, please send an e-mail to `deitel@deitel.com`; we will respond promptly. Please visit our Web site, `www.deitel.com`, regularly and be sure to sign up for the *DEITEL® Buzz Online* e-mail newsletter at `www.deitel.com/newsletter/subscribe.html`. We use the Web site and the newsletter to keep our readers current on all Deitel publications and services.

Features of Internet & World Wide Web How to Program, Third Edition

This edition contains many new features and enhancements including:

Full-Color Presentation

The book enhances live-code examples by using full color. Readers see sample outputs as they would appear on their computer screens.

Syntax Coloring

We syntax color all the program code, as do many integrated development environments and code editors. This greatly improves code readability—an especially important goal, given that this book contains over 17,229 lines of code. Our syntax-coloring conventions are as follows:

```
comments appear in green
keywords appear in dark blue
ASP.NET, ASP, PHP, ColdFusion and JSP delimiters appear in red
constants and literal values appear in light blue
all other code appears in black
```

Code Highlighting

We have added extensive code highlighting to make it easier for readers to spot the featured segments of each program. This feature also helps students review the material rapidly when preparing for exams or labs.

"Code Washing"

"Code washing" is our term for commenting, using meaningful identifiers, applying indentation and using vertical spacing to separate meaningful program units. This process results in programs that are much more readable and self-documenting. We have added extensive and descriptive comments to all of the code, including a comment before and after every major control statement, to help the student clearly understand the flow of the program. We have done extensive code washing of all the source code programs in the text and the ancillaries.

XHTML (Chapters 4 & 5)

This edition uses XHTML as the primary means of describing Web content. The World Wide Web Consortium deprecated HTML 4 and replaced it first with XHTML 1.0 (Extensible Hypertext Markup Language) and now with XHTML 1.1. XHTML is derived from XML (Extensible Markup Language), which allows Web developers to create their own tags and languages. XHTML is replacing HTML as the standard for marking up Web content.

Macromedia Flash™ MX 2004 (Chapters 17 & 18)

Flash is a cutting-edge multimedia application that enables Web developers to create interactive, animated content. Through hands-on examples, we show how to add interactivity, sound and animation to Web sites while teaching the fundamentals of Flash and **ActionScript**—Flash's scripting language. The examples in Chapter 17 show readers how to create interactive buttons, animated banners and animated splash screens (called **animation pre-loaders**). Chapter 18 presents a case study that demonstrates how to build an interactive game in Flash MX 2004 using intermediate to advanced ActionScript.

Macromedia® Dreamweaver® MX 2004 (Chapter 19)

This edition features a new chapter that demonstrates Macromedia Dreamweaver MX 2004, the leading WYSIWYG (What You See Is What You Get) editor used by Web programmers today. While our book advocates a thorough knowledge of XHTML and its accompanying scripting languages, Dreamweaver is a useful tool for creating and maintaining Web pages. By displaying XHTML elements exactly as a browser would, Dreamweaver allows users to apply text formatting and insert images, lists, tables and forms into Web pages with relative ease.

XML (Chapter 20)

Throughout the book we emphasize XHTML, which derived from XML and HTML. XML derives from SGML (Standardized General Markup Language), whose sheer size and complexity limits its use beyond heavy-duty, industrial-strength applications. XML is a technology created by the World Wide Web Consortium for describing data in a portable format. Document authors use XML's extensibility to create entirely new markup languages for describing specific types of data, including mathematical formulas, chemical molecular structures and music. We also introduce Web services and XML-based applications that promote a flow of information between business organizations and individuals.

Server-Side Technology

We present condensed treatments of seven popular Internet/Web programming languages for building the server side of Internet- and Web-based client/server applications. In Chapters 23 and 24, we discuss ASP.NET—Microsoft's technology for server-side scripting. In Chapters 25 and 26, we introduce Perl and PHP—two open-source scripting languages used extensively for programming Web-based applications. Chapter 27 is a new chapter for this edition and introduces Macromedia's server-side technology, ColdFusion®. Chapters 33–34 (both on CD) present Active Server Pages, the predecessor to ASP.NET. Chapter 35 (on CD) presents Python, an interpreted, interactive, object-oriented programming language responsible for the server-side of many Web applications. Finally, in Chapters 36 and 37 (both on CD), we provide two bonus chapters for Java programmers on Java™ servlets and JavaServer Pages™ (JSP).

Macromedia ColdFusion MX (Chapter 27)

This edition features a new server-side technology: Macromedia ColdFusion MX. ColdFusion is a popular Web application server that enables quick and easy dynamic Web-site development. The popularity of ColdFusion lies in its server-side, tag-based **ColdFusion Markup Language** (**CFML**). CFML is based around a set of tags that work like XHTML tags, making it easy to learn for most Web-site developers. In addition, CFML offers many advanced features that make it as rich as a full-scale programming language.

Teaching Approach

Why We Wrote *Internet & World Wide Web How to Program, Third Edition*

Our goal in creating this book was clear: Produce a textbook for introductory university-level courses in computer programming for students with little or no programming experience, yet offer the depth and rigorous treatment of theory and practice demanded by traditional, upper-level programming courses and professionals. We also wanted the book to expose all readers—beginners and experienced traditional programmers alike—to the great opportunities the Web offers the field of Computer Science. To meet these goals, we produced a comprehensive book that teaches the principles of control statements, object-based programming, various markup languages (e.g., XHTML, XML) and scripting languages. Throughout the book, we develop practical Web-based applications from which even the most experienced programmers can benefit. After mastering the material in this book, students entering upper-level programming courses and industry will be well prepared to take advantage of the Internet and the Web.

Dr. Harvey M. Deitel taught introductory programming courses in universities for 20 years with an emphasis on developing clearly written, well-designed programs. Much of what is taught in these courses are the basic principles of programming with an emphasis on the effective use of control statements and functionalization. We present these topics in *Internet & World Wide Web How to Program, Third Edition,* the way HMD has done in his university courses. Students are highly motivated by the fact that they are learning eight leading-edge scripting technologies (JavaScript, JScript .NET, Flash ActionScript, Perl, PHP, ColdFusion, VBScript and Python) and a leading-edge programming paradigm (object-based programming). We also teach Dynamic HTML, a means of adding "dynamic content" to Web pages. Instead of Web pages with only text and static graphics, Web pages "come alive" with audio, video, animation, interactivity and three-dimensional moving images. These programming languages will be useful to students immediately as they leave the university environment and head into a world in which the Internet and Web have massive prominence.

This book is intended for several academic markets, namely, the introductory course sequences in which C++, Java and Visual Basic are traditionally taught; upper-level elective courses for students who already know programming and as a supplement in introductory courses, where students are first becoming familiar with computers, the Internet and the Web. The book offers a solid one- or two-semester introductory programming experience or an extensive one-semester upper-level elective. The book is also intended for professional programmers in corporate training programs or for self study.

Many books about the Web concentrate on developing attractive Web pages. We discuss Web-page design intensely. But more importantly, the key focus of this book is on Web-based applications development. Our audiences want to build real-world, industrial-strength, Web-based applications. These audiences care about good-looking Web pages, but they also care about client/server systems, databases, distributed computing, etc. Many books about the Web are reference manuals with exhaustive listings of features. That is not our style. We concentrate on creating real applications. We provide the live-code examples on the CD accompanying this book (and at `www.deitel.com`) so that you can run the applications and see and hear the multimedia outputs. You can interact with our game and art programs. The Web is an artist's paradise. Your creativity is your only limitation. However, the Web contains so many tools and mechanisms to leverage your abilities that even if you are not artistically inclined, you can create stunning output. Our goal is to help you master these tools so that you can maximize your creativity and development abilities.

People want to communicate. Sure, they have been communicating since the dawn of civilization, but computer communications have been limited mostly to digits, alphabetic characters and special characters. The next major wave of communication technology is multimedia. People transmit color pictures and they transmit voices, sounds and audio clips as well. They transmit full-motion color video. At some point, they will insist on three-dimensional, moving-image transmission. Our current flat, two-dimensional televisions eventually will 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 the Internet will 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. Why stop there? It could also replace radio and television as we know them today. It is not hard to imagine the Internet and the Web replacing newspapers with electronic news media. Many newspapers and magazines already offer Web-based versions, some fee based and some free. Increased bandwidth makes it possible to stream audio and video over the Web. Both companies and individuals run their own Web-based radio and television stations. Just a few decades ago, there were only a few television stations. Today, standard cable boxes accommodate about 100 stations. In a few more years, we will have access to thousands of stations broadcasting over the Web worldwide. This textbook may someday appear in a museum alongside radios, TVs and newspapers in an “early media of ancient civilization” exhibit.

Internet & World Wide Web How to Program, Third Edition 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 software engineering and stresses program clarity. We are educators who teach edge-of-the-practice topics in industry classrooms worldwide. This text emphasizes good pedagogy.

Font Conventions

We use fonts to distinguish on-screen components (such as menu names and menu items) from other elements that appear on screen. Our convention is to emphasize on-screen components in a sans-serif bold **Helvetica** font (for example, **Properties**), and to emphasize program text in a serif `Lucida` font (for example, `int x = 5`).

Live-Code Approach

Internet & World Wide Web How to Program, Third Edition, is loaded with numerous live-code examples—each new concept is presented in the context of a complete, working program that is immediately followed by one or more sample executions showing the program’s input/output dialog. This style exemplifies the way we teach and write about programming. We call this method of teaching and writing the live-code approach. *We use programming languages to teach programming languages*. Reading the examples in the text is much like typing and running them on a computer.

This textbook “jumps right in” with XHTML in Chapter 4, then rapidly proceeds with programming in JavaScript, Microsoft’s Dynamic HTML, Flash ActionScript, XML, ASP.NET, Perl, PHP, ColdFusion, VBScript (on CD), Python (on CD), Java Servlets (on CD) and JavaServer Pages (on CD). Many students wish to “cut to the chase;” there is great stuff to be done in these languages so let’s get to it! Web programming is not trivial by any means, but it is fun, and students can see immediate results. Students can get graphical, animated, multimedia-based, audio-intensive, database-intensive, network-based programs running quickly through “reusable components.” They can implement impressive projects. They can be more creative and productive in a one- or two-semester course than is possible in introductory courses taught in conventional programming languages, such as C, C++, Visual Basic and Java. [*Note*: This book includes Java Servlets and JavaServer Pages as “bonus chapters;” it does not teach the fundamentals of Java programming. Readers who want to learn Java may want to consider reading our book *Java How to Program, Fifth Edition*. Readers who desire a deeper, more developer-oriented treatment of Java may want to consider reading our book, *Advanced Java 2 Platform How to Program.*]

Web Access

All of the source-code examples for *Internet & World Wide Web How to Program*, *Third Edition*, (and our other publications) are available on the Internet as downloads from the following Web sites:

```
www.deitel.com
www.prenhall.com/deitel
```

Registration is quick and easy and the downloads are free. We suggest downloading all the examples, then running each program as you read the corresponding text. Making changes to the examples and immediately seeing the effects of those changes is a great way to enhance your learning experience. [*Note*: The examples are also on the CD that accompanies the book.]

Objectives

Each chapter begins with a statement of objectives. This tells students what to expect and gives students an opportunity, after reading the chapter, to determine if they have met these objectives. This 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. Many of the quotations are worth a “second look” *after* reading the 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.

17,229 Lines of Code in 282 Example LIVE-CODE Programs (with Program Outputs)

We present Internet & Web technologies in the context of complete, working programs, that range from just a few lines of code to substantial examples with several hundred lines of code. The examples are available on the accompanying CD and as downloads from our Web site (`www.deitel.com`) and Prentice Hall’s Web site (`www.prenhall.com\deitel`).

701 Illustrations/Figures

An abundance of charts, line drawings and program outputs is included.

457 Programming Tips

We have included 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*, *Error-Prevention Tips*, *Performance Tips*, *Portability Tips*, *Software Engineering Observations* and *Look-and-Feel Observations*. These tips and practices represent the best we have gleaned from decades of programming and teaching experience. One of our students—a mathematics major—told us that she feels this approach is like the highlighting of axioms, theorems and corollaries in mathematics books; it provides a foundation on which to build good software programs.

82 Good Programming Practices

Good Programming Practices *are tips for writing clear programs. These techniques help students produce programs that are more readable, self-documenting and easier to maintain.*

137 Common Programming Errors

Students learning a language—especially in their first programming course—tend to make certain kinds of errors frequently. Focusing on these Common Programming Errors *helps students avoid making the same errors. It also helps reduce long lines outside instructors' offices during office hours*!

45 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 highlight opportunities for improving program performance—making programs run faster or minimizing the amount of memory that they occupy.

30 Portability Tips

Software development is a complex and expensive activity. Organizations that develop software often produce versions customized to a variety of computers and operating systems. So there is a strong emphasis today on portability, i.e., on producing software that will run on a variety of computer systems with few, if any, changes. Some people assume that if they implement an application in a programming language, the application will automatically be portable. This is simply not the case. Achieving portability requires careful and cautious design. There are many pitfalls. We include numerous Portability Tips *to help students write portable code.*

106 Software Engineering Observations

The Software Engineering Observations *highlight techniques, architectural issues and design issues, etc. that affect the architecture and 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.*

43 Error-Prevention Tips

When we first designed this "tip type," we thought we would use them strictly to tell people how to test and debug programs and in previous editions we labelled these as "Testing and Debugging Tips." In fact, many of the tips describe aspects of programming that reduce the likelihood of "bugs" and thus simplify the testing and debugging processes.

14 Look-and-Feel Observations

We provide Look-and-Feel Observations *to highlight graphical user interface conventions. These observations help students design their own graphical user interfaces to conform with industry norms.*

Summary (1413 Summary bullets)

Each chapter includes additional pedagogical devices. We present a thorough, bullet-list-style summary of the chapter. On average, each chapter contains 37 summary bullets that help students review and reinforce important concepts.

Terminology (3096 Terms)

We include an alphabetized list of the important terms defined in each chapter—again, for further reinforcement. There is an average of 81 terms per chapter. Each term also appears in the index, so the student can locate terms and definitions quickly.

605 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 to prepare for the regular exercises.

657 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 statements; writing small portions of functions; writing complete functions and scripts; and writing major term projects. The large number of exercises across a wide variety of topics 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 vast majority of 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**. **We regret that we cannot provide the solutions to professionals**.**]** The student should have two key projects in mind while reading through this book as well—developing a personal Web site using XHTML markup and JavaScript coding, and developing a complete client/server, database-intensive Web-based application by using techniques taught throughout this book.

Approximately 6800 Index Entries (with approximately 9500 Page References)

We have included an extensive index at the back of the book. This helps students 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 terms in the *Terminology* sections appear in the *Index* (along with many more index items from each chapter). Thus, students can use the *Index* in conjunction with the *Terminology* sections to be sure they have covered the key material of each chapter.

"Double Indexing" of All Live-Code Examples and Exercises

Internet & World Wide Web How to Program, Third Edition has 282 live-code examples and 657 exercises (including parts). We have double indexed each of the live-code examples and most of the more challenging projects. For every source-code program in the book, we took the file name and indexed it both alphabetically and as a subindex item under "Examples." This makes it easier to find examples using particular features. The more substantial exercises are indexed both alphabetically and as subindex items under "Exercises."

Bibliography

An extensive bibliography of books, articles and online documentation is included to encourage further reading.

Software Included with Internet & World Wide Web How to Program, Third Edition

[*Note:* We assume readers have Microsoft's Internet Explorer 6 installed on their machines. If you do not have IE 6, you can download for free at `www.microsoft.com/windows/ie/downloads/critical/ie6sp1/default.asp`.] The CD that accompanies this book contains Microsoft Agent 2.0, Adobe® Reader 6.0, MySQL 4.0, PHP 4.3.3, Apache HTTP Server 2.0.47, Microsoft .NET Framework Redist version 1.1, .NET Framework Software Development Kit (SDK) 1.1, Macromedia Flash MX 2004 (30-day full-trial version), Dreamweaver MX 2004 (30-day full-trial version) and ColdFusion MX 6.1 (30-day full-trial version that turns into a free developer edition). The CD also contains the book's examples and a Web page with links to the Deitel & Associates, Inc. Web site, to the Prentice Hall Web site and to a Web page containing links to the Web resources mentioned in the chapters. If you have access to the Internet, this Web page can be loaded into your World Wide Web browser to give you quick access to all the resources.

For the Perl and Python chapters, you will need to download and install appropriate versions of these languages. We developed these chapters using ActiveState ActivePerl 5.8.0 (`www.activestate.com/Products/ActivePerl`) and ActiveState ActivePython 2.2.2 (`www.activestate.com/Products/ActivePython`). Also, a 30-day evaluation copy of Adobe Photoshop Elements 2.0, used in Chapter 3 to illustrate how to create Web graphics, can be downloaded at `www.adobe.com/products/photoshopel`. [*Note*: All of the software included on the CD that accompanies this text and the links to download software were current and active at the time of publication. However, please be aware that Web links have a tendency to change or break, and companies upgrade software to newer versions on a regular basis.]

If you have any questions about the software or Web links on the CD, please read the introductory documentation on the CD. We will post additional information and additional software installation instructions on the book's Web site, `www.deitel.com/books/iw3HTP3/index.html`. If you have any technical questions about CD installation or about any of the software supplied with Deitel/Prentice Hall products, please e-mail `media.support@pearsoned.com`. They will respond promptly.

Ancillary Package for Internet & World Wide Web How to Program, Third Edition

Internet & World Wide Web How to Program, Third Edition, has extensive ancillary materials for instructors. The *Instructor's Resource CD (IRCD)* contains solutions to most of the end-of-chapter exercises. This CD is available only to instructors through their Prentice

Hall representatives. **[*Note*: Please do not write to us requesting the instructor's CD. Distribution of this CD is limited strictly to college professors teaching from the book. Instructors may obtain the solutions manual only from their Prentice Hall representatives.]** The ancillaries for this book also include a *Test Item File* of multiple-choice questions. In addition, we provide PowerPoint slides containing all the code and figures in the text and bulleted items that summarize the key points in the text. Instructors can customize the slides. The PowerPoint slides are downloadable from `www.deitel.com` and are available as part of Prentice Hall's Companion Web Site (`www.prenhall.com/deitel`) for *Internet & World Wide Web How to Program, Third Edition*, which offers resources for both instructors and students. For instructors, the Companion Web Site offers a Syllabus Manager, which helps instructors plan courses interactively and create online syllabi.

Students also benefit from the functionality of the *Companion Web Site*. Book-specific resources for students include:

- Customizable PowerPoint slides
- Source code for all example programs
- Reference materials from the book appendices (such as operator-precedence chart, character set and Web resources)

Chapter-specific resources available for students include:

- Chapter objectives
- Highlights (e.g., chapter summary)
- Outline
- Tips (e.g., *Common Programming Errors*, *Good Programming Practices*, *Portability Tips*, *Performance Tips*, *Look-and-Feel Observations*, *Software Engineering Observations* and *Error-Prevention Tips*)
- Online Study Guide—contains additional short-answer self-review exercises (e.g., true/false questions) with answers and provides immediate feedback to the student

Students can track their results and course performance on quizzes using the *Student Profile* feature, which records and manages all feedback and results from tests taken on the *Companion Web Site*. To access *DEITEL® Companion Web Sites*, visit `www.prenhall.com/deitel`.

DEITEL® Buzz Online E-mail Newsletter

Our free e-mail newsletter, the *DEITEL® Buzz Online*, includes commentary on industry trends and developments, links to free articles and resources from our published books and upcoming publications, product-release schedules, errata, challenges, anecdotes, information on our corporate instructor-led training courses and more. To subscribe, visit

```
www.deitel.com/newsletter/subscribe.html
```

DEITEL® e-Learning Initiatives

Wireless devices will have an enormous role in the future of the Internet. Given recent bandwidth enhancements and the emergence of 2.5 and 3G technologies, it is projected that, within a few years, more people will access the Internet through wireless devices than through desktop computers. Deitel & Associates is committed to wireless accessibility and has published *Wireless Internet & Mobile Business How to Program*. We are investigating new electronic formats, such as wireless e-books so that students and professors can access content virtually anytime, anywhere. For periodic updates on these initiatives subscribe to the *DEITEL® Buzz Online* e-mail newsletter, `www.deitel.com/newsletter/subscribe.html` or visit `www.deitel.com`.

DEITEL® Developer Series

Deitel & Associates, Inc., is making a major commitment to covering leading-edge technologies for industry software professionals through the launch of our *DEITEL® Developer Series*. The first books in the series are *Web Services A Technical Introduction* and *Java Web Services for Experienced Programmers*. We are working on *ASP .NET with Visual Basic .NET for Experienced Programmers*, *ASP .NET with C# for Experienced Programmers* and many more. Please visit `www.deitel.com` or subscribe to our e-mail newsletter at `www.deitel.com/newsletter/subscribe.html` for continuous updates on all published and forthcoming *DEITEL Developer Series* titles.

The *DEITEL Developer Series* is divided into three subseries. The *A Technical Introduction* subseries provides IT managers and developers with detailed overviews of emerging technologies. The *A Programmer's Introduction* subseries is designed to teach the fundamentals of new languages and software technologies to programmers and novices from the ground up; these books discuss programming fundamentals, followed by brief introductions to more sophisticated topics. The *For Experienced Programmers* subseries is designed for seasoned developers seeking a deeper treatment of new programming languages and technologies, without the encumbrance of introductory material; the books in this subseries move quickly to in-depth coverage of the features of the programming languages and software technologies being covered.

Tour of the Book

In this section, we take a tour of the subjects you will study in *Internet and World Wide Web How to Program, Third Edition*. Many of the chapters end with a Web Resources section that provides a listing of resources through which you can enhance your knowledge and use of the Web. In addition, you may want to visit the book's Web site `www.deitel.com/books/iw3HTP3/index.html` for additional resources. Due to the volume of information presented in this book, Chapters 30 and higher and Appendix G can be found on the accompanying CD as Adobe PDF documents. [*Note*: The table of contents and index indicate material located on the CD with red text.]

Chapter 1—Introduction to Computers and the Internet

In Chapter 1, we present historical information about computers and computer programming and introductory information on the Internet and the Web. We also overview the technologies and concepts discussed in the remaining chapters of the book.

Chapter 2—Microsoft Internet Explorer 6

Prior to the explosion of interest in the Internet and the Web, if you heard the term **browser**, you probably thought about browsing at a bookstore. Today "browser" has a whole new meaning—an important piece of software that enables you to view Web pages. The two most popular browsers are Microsoft's Internet Explorer and Netscape. Throughout this book, we use Internet Explorer 6, but we provide a solid introduction to Netscape 7.1 at `www.deitel.com`. Using tools included with Internet Explorer, we demonstrate how to use the Web. This chapter shows readers unfamiliar with the Web how to browse the Web with Internet Explorer. We demonstrate several commonly used features for searching the Web, keeping track of the sites you visit and transferring files between computers. The chapter also covers e-mail and instant messaging technologies, which allow users to communicate over the Internet in real time. The chapter concludes with a discussion of alternative browsers and their features and functionality.

Chapter 3—Adobe Photoshop Elements: Creating Web Graphics

The Internet and Web are rich in multimedia content. Web pages contain colorful graphics, sounds and text. Graphics are an essential element of Web-page design that convey visual information about a site's contents. In this chapter, we introduce **Adobe Photoshop Elements**—a graphics software package that contains an extensive set of tools and features for creating high-quality graphics and animations. These tools include filters for applying special effects and screen capturing for taking "snap shots" of the screen. Chapter examples demonstrate creating title images for a Web page, creating a navigation bar that contains a series of buttons used to connect a Web site's pages and manipulating images by using advanced photographic effects. We focus on creating and manipulating the two most popular image formats used in Web documents: **Graphics Interchange Format (GIF)** and **Joint Photographic Expert Group (JPEG)** files. We also discuss the **Portable Network Graphic (PNG)** format, an increasingly popular image format. [*Note*: Readers can download a 30-day evaluation copy of Photoshop Elements from `www.adobe.com/products/photoshopel`. The chapter examples were developed using that version of Photoshop Elements.]

Chapter 4—Introduction to XHTML: Part 1

In this chapter, we unlock the power of Web-based application development by introducing **XHTML**—the **Extensible HyperText Markup Language**. XHTML is a **markup language** for identifying the elements of an XHTML document (or Web page) so that a browser can render (i.e., display) that document on a computer screen. We introduce basic XHTML Web-page creation using a technique we call the live-code approach. Every concept is presented in the context of a complete, working XHTML document. We render each working example in Internet Explorer and show the screen outputs. We present many short Web pages that demonstrate XHTML features. Later chapters introduce more sophisticated XHTML techniques, such as **tables**, which are useful for formatting information retrieved from a database. We introduce XHTML **tags** and **attributes**, which describe the document's information. A key issue when using XHTML is the separation of the **presentation** of a document (i.e., how the document is rendered on the screen by a browser) from the **structure** of the information in that document (i.e., the information the document contains). This chapter begins our in-depth discussion of this issue. As the book proceeds, you will be able to create appealing and powerful Web pages and Web-based applications. Other topics in this chapter include incorporating text, images and special characters (such as copyright and trademark symbols) into an XHTML document, validating an XHTML document to ensure that it is written correctly, placing information inside **lists**, separating parts of an XHTML document with horizontal lines (called **horizontal rules**) and linking to other XHTML documents on the Web.

Chapter 5—Introduction to XHTML: Part 2

In this chapter, we discuss more substantial XHTML elements and features. We demonstrate how to present information in tables and how to gather user input. We explain and demonstrate **internal linking** and **image maps** to make Web pages more navigable and how to use **frames** to display multiple XHTML documents in a browser. **XHTML forms** are one of the most important features introduced in this chapter—forms display information to the user and accept user input. By the end of this chapter, readers should be familiar with the most popular XHTML tags and features used to create Web sites.

Chapter 6—Cascading Style Sheets (CSS™)

Web browsers control the appearance (i.e., the rendering) of every Web page. For instance, one browser may render an `h1` (i.e., a large heading) element in an XHTML document differently than another browser. With the advent of **Cascading Style Sheets (CSS)**, Web developers can control the appearance of their Web pages more accurately. CSS allows Web developers to specify the style of their Web page's elements (spacing, margins etc.) separately from the structure of their pages (section headers, body text, links etc.). This **separation of structure from content** allows greater manageability and makes changing document styles easier and faster. We introduce **inline**, **embedded** and **external** style sheets. Inline style sheets are applied to individual XHTML elements, embedded style sheets are entire style sheets placed directly inside an XHTML document and external style sheets are style sheets located outside an XHTML document.

Chapter 7—JavaScript:[1] Introduction to Scripting

Chapter 7 presents our first JavaScript **programs**[2] (also called **scripts**). Scripting helps Web pages "come alive." Web developers dynamically manipulate Web-page elements through scripting as clients browse Web pages. Chapters 7–12 present JavaScript, which is then used in Chapters 13–16 and in Chapters 30–31 (which are on the CD) to manipulate Web-page content. We present the key fundamental computer-science concepts of JavaScript at the same depth as we do in our other books on conventional programming languages (such as C, C++, C#, Java and Visual Basic), but in the exciting context of the Web. Using our live-code approach, we present every concept in the context of a working JavaScript program that is immediately followed by the screen output. The chapter introduces nonprogrammers to basic programming concepts and constructs. The scripts in this chapter illustrate how to output text to a browser and how to obtain user input through the browser. Some of the input and output is performed using the browser's capability to display predefined **graphical user interface (GUI)** windows (called **dialogs**). This allows nonprogrammers to concentrate on fundamental programming concepts and constructs rather than

[1] JavaScript was originally created by Netscape; Microsoft's version is called JScript. The two scripting languages have a common subset between them known as ECMA-262. ECMA-262 defines a universal, client-side scripting language that ECMA International (originally the European Computer Manufacturers Association) developed in conjunction with Netscape, Microsoft and other companies. JavaScript and JScript each conform to the ECMA-262 standard. We typically refer to ECMA-262 as JavaScript.

[2] The book's JavaScript examples execute in Microsoft Internet Explorer 6. We have tested these examples on Internet Explorer 6 and Netscape 7.1. For those few examples that do not execute in Netscape 7.1, we have (when possible) created Netscape 7.1 equivalent examples. These examples and the test results are available at `www.deitel.com`.

on GUI components and on GUI **event handling**. Chapter 7 also provides detailed treatments of **decision making** and **arithmetic operations**.

Chapter 8—JavaScript: Control Statements I

Chapter 8 focuses on the program-development process. The chapter discusses how to develop a working JavaScript program from a **problem statement** (i.e., a **requirements document**). We show the intermediate steps using, a program development tool called **pseudocode**. The chapter introduces some simple control statements used for decision making (`if` and `if…else`) and repetition (`while`). We examine counter-controlled repetition and sentinel-controlled repetition and introduce the increment, decrement and assignment operators. Simple flowcharts illustrate graphically the flow of control through each of the control statements. This chapter helps the student develop good programming habits in preparation for the more substantial programming tasks in the remainder of the book.

Chapter 9—JavaScript: Control Statements II

Chapter 9 discusses much of the material JavaScript has in common with the Java programming language, especially the **sequence**, **selection** and **repetition** control statements. Here, we introduce one additional control statement for decision making (`switch`) and two additional control statements for repetition (`for` and `do…while`). This chapter also introduces several operators that allow programmers to define complex conditions in their decision-making and repetition statements. The chapter uses flowcharts to illustrate the flow of control through each of the control statements, and concludes with a summary that enumerates each of the statements. The techniques discussed in this chapter and in Chapter 10 constitute a large part of what has been traditionally taught in universities under the topic of structured programming.

Chapter 10—JavaScript: Functions

Chapter 10 takes a deeper look inside scripts. Scripts contain data called **global** (or **script-level**) **variables** and executable units called **functions**. We discuss JavaScript functions, programmer-defined functions and **recursive** functions (i.e., functions that call themselves). The techniques presented in Chapter 10 are essential to produce properly structured programs, especially large programs that Web developers are likely to build in real-world, Web-based applications. The **divide-and-conquer** strategy is presented as an effective means for solving complex problems by dividing them into simpler, interacting components. The chapter offers an introduction to recursion and includes a table summarizing the recursion examples and exercises found in Chapters 10–13. We introduce **events** and event handling—elements required for programming graphical user interfaces (GUIs) in XHTML forms. Events are notifications of state changes, such as button clicks, mouse clicks, pressing keyboard keys, etc. JavaScript allows programmers to respond to various events by coding functions called **event handlers**. This begins our discussion of **event-driven programming**—the user drives the program by interacting with GUI components (causing **events such as mouse clicks**), and the scripts respond to the events by performing appropriate tasks (**event handling**). The event-driven programming techniques introduced here are used in scripts throughout the book. Dynamic HTML event handling is introduced in Chapter 14. Chapter 10 contains a rich set of exercises that include the Towers of Hanoi, computer-aided instruction and a guess-the-number game.

Chapter 11—JavaScript: Arrays

Chapter 11 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. The chapter presents numerous examples of both single-subscripted arrays and double-subscripted arrays. It is widely recognized that structuring data properly is as important as using control statements effectively in the development of properly structured programs. Examples in the chapter investigate various common array manipulations, searching arrays, sorting data and passing arrays to functions. This chapter introduces JavaScript's `for…in` control statement, which interacts with collections of data stored in arrays. The end-of-chapter exercises include a variety of interesting and challenging problems, such as the **Sieve of Eratosthenes** and the design of an airline reservations system. The chapter exercises also include a delightful simulation of the classic race between the tortoise and the hare.

Chapter 12—JavaScript: Objects

This chapter discusses **object-based programming** with JavaScript's built-in objects. The chapter introduces the terminology of objects. We overview the methods (functions associated with particular objects) of the JavaScript `Math` object and provide several examples of JavaScript's string-, date- and time-processing capabilities with the `String` and `Date` objects. An interesting feature of the `String` object is a set of methods that help a programmer output XHTML from a script by enclosing strings in XHTML elements. The chapter also discusses JavaScript's `Number` and `Boolean` objects, useful for performing mathematical and logical operations, and JavaScript's `document`, `window`, `history` and `location` objects specifically designed for the Web. In addition, we demonstrate how to personalize Web sites using cookies (small pieces of data) stored on client computers. At the end of the chapter, we build an interesting example that combines many of the JavaScript concepts presented in Chapters 7–12. Many challenging, yet entertaining, string-manipulation exercises are included.

Chapter 13—Dynamic HTML:[3] DHTML Object Model and Collections

A massive switch is occurring in the computer industry. The procedural programming style used since the inception of the industry is being replaced by the object-oriented style of programming. The vast majority of new software efforts use object technology in one form or another. The scripting languages we discuss in this book usually manipulate existing objects by sending messages that either inquire about the objects' attributes or ask the objects to perform certain actions. In this chapter, we continue the discussion of object technology by presenting Microsoft's Dynamic HTML object model. As Internet Explorer downloads a Web page from a server, it converts each element to an object. Objects store data (their attributes) and perform functions (their methods). Through scripting languages such as JavaScript, you can write commands that *get* or *set* (i.e., read or write) an object's attributes. You can also write commands that invoke an object's methods. The chapter exercises provide the opportunity to program the classic "15-puzzle" game.

[3] Microsoft Dynamic HTML and Netscape Dynamic HTML are incompatible. In this book, we focus on Microsoft Dynamic HTML. We have tested all of the Dynamic HTML examples in Internet Explorer 6 and Netscape 7.1. All of these examples execute in Internet Explorer, but do not execute in Netscape 7.1. We have posted the testing results at `www.deitel.com/books/iw3HTP3/index.html`. In this book, we also present Macromedia Flash content which executes in Internet Explorer and Netscape.

Chapter 14—Dynamic HTML: Event Model

We have discussed how scripting can control XHTML pages. Dynamic HTML includes an **event model** that enables scripts to respond to user actions. This allows Web applications to be more responsive and user friendly. With the event model, scripts can respond to a user moving or clicking the mouse, scrolling up or down the screen or entering keystrokes. Content becomes more dynamic, while interfaces become more intuitive. We discuss how to use the event model to respond to user actions. We provide examples of event handling, which range from mouse capture to error handling to form processing. For example, we call the `onreset` event to confirm that a user wants to reset the form (i.e., the GUI in which the user inputs data). For one of the chapter exercises, the reader creates an interactive script that displays an image alongside the mouse pointer. When the mouse pointer is moved, the image moves with it.

Chapter 15—Dynamic HTML: Filters and Transitions

Internet Explorer includes a set of **filters** that allow developers to perform complex image transformations entirely in the Web browser without the need for additional downloads from a Web server. Filters are scriptable, so the developer can create stunning, customized animations with a few lines of client-side JavaScript. We introduce the `fliph` and `flipv` filters, which mirror text and images horizontally and vertically. We explain the `gray`, `xray` and `invert` filters, which all apply simple transformations to images. We introduce many of the filters that apply effects such as shadows, transparency gradients and distortions. Internet Explorer enables **transitions** that are similar to transitions between slides in PowerPoint-like presentations. The `revealTrans` filter applies visual effects such as box in, circle out, wipe left, vertical blinds, checkerboard across, random dissolve, split horizontal in, strips right up and random bars horizontal. This chapter also introduces the `blendTrans` filter, which allows you to fade in or fade out of an XHTML element over a set interval.

Chapter 16—Dynamic HTML: Data Binding with Tabular Data Control

This is one of the most important chapters in the book for people who want to build substantial, real-world Web-based applications. Businesses thrive on data, and Dynamic HTML helps Web developers build data-intensive applications. With **data binding**, data does not reside exclusively on the server. Data are sent from the server to the client, and all subsequent manipulations of the data occur on the client. Data can be maintained on the client in a manner that distinguishes the data from the XHTML markup. Manipulating data on the client improves performance by eliminating server activity and network delays. Once data is available on the client, the data can be **sorted** (i.e., arranged into ascending or descending order) and **filtered** (i.e., selected according to some criterion) in various ways. We present examples of each of these operations. To bind external data to XHTML elements, Internet Explorer employs software capable of connecting the browser to live data sources, known as **Data Source Objects (DSOs)**. Several DSOs are available in Internet Explorer—in this chapter, we discuss the most popular DSO, **Tabular Data Control (TDC)**.

Chapter 17—Macromedia Flash MX 2004: Building Interactive Animations

Macromedia Flash is a cutting-edge multimedia application that creates interactive content for the World Wide Web. Through hands-on examples, this chapter shows how to add interactivity, sound and animation to Web sites, while teaching the fundamentals of Macromedia Flash and **ActionScript**, Flash’s scripting language. The chapter examples include creating interactive buttons, animated banners and animated splash screens (called **animation preloaders**). The exercises ask the reader to create a navigation bar, a spotlight effect and a morphing effect. The morphing effect exercise in particular is a wonderful illustration of the power of Flash. From this chapter, the reader will gain a thorough knowledge of Flash's graphic and animation capabilities, as well as an introductory understanding of ActionScript.

Chapter 18— Macromedia Flash MX 2004: Building an Interactive Game

While Chapter 17 provides a comprehensive introduction to Flash animation, it just begins to introduce the student to ActionScript's vast capabilities. Chapter 18 leads the student, step-by-step, through creating an interactive game. The skills learned in this case study include, but are not limited to, creating global variables, accessing objects on different layers, creating ActionScript functions, handling various events with event handlers, detecting object collisions with collision detectors and creating a time counter. Ultimately, the student is rewarded with a game in which the user has a limited amount of time to knock out pieces of a moving target. The exercises include adding a final score display and creating multiple levels of play.

Chapter 19— Macromedia Dreamweaver MX 2004

Macromedia Dreamweaver is the leading WYSIWYG (What You See Is What You Get) editor used by Web programmers today. While our book advocates a thorough knowledge of XHTML and its accompanying scripting languages, we acknowledge Dreamweaver as a useful tool for creating and maintaining Web pages and complete Web sites. This chapter walks the reader through using Dreamweaver to add elements, such as images, lists, line breaks, special characters and text formatting to a Web page. We provide comprehensive hands-on examples, including table and form creation. We also discuss Dreamweaver's powerful site creation and management capabilities. Finally, we discuss incorporating JavaScript into Dreamweaver pages. Since server-side scripting languages are discussed in later chapters of the book, we do not cover them in detail in this chapter. Still, Dreamweaver's integration with ASP.NET, ColdFusion, JSP and PHP is explained briefly. The exercises include creating complex tables and forms.

Chapter 20—Extensible Markup Language (XML)

Throughout the book, we have been emphasizing XHTML. This language derives from SGML (Standardized General Markup Language), which became an industry standard in 1986. SGML is employed in publishing applications worldwide, but it has not been incorporated into mainstream computing and information technology curricula. Its sheer size and complexity limit its use beyond heavy-duty, industrial-strength applications. The Extensible Markup Language (XML) is an effort to make SGML-like technology available to a much broader community. XML, a condensed subset of SGML, contains additional features for usability. XML differs in concept from XHTML. XHTML is a markup language, and XML is a language for *creating* markup languages. XML enables document authors to create their own markup for virtually any type of information. As a result, document authors use this extensibility to create entirely new markup languages to describe specific types of data, including mathematical formulas, chemical molecular structures, music and recipes. Markup languages created with XML include XHTML (Chapters 4 and 5), MathML (for mathematics), VoiceXML (for speech), SMIL™ (the Synchronized Multimedia Integration Language, for multimedia presentations), SVG (Scalable Vector Graphics, for vector and raster graphics), CML (Chemical Markup Language, for chemistry) and XBRL (Extensible Business Reporting Language, for financial data exchange). XML is a technology created by the World Wide Web Consortium for describing data in a portable format. XML is one of most important technologies in industry today and is being integrated into almost every field. Every day, companies and individuals are finding new and exciting uses for XML. In this chapter, we present examples that illustrate the basics of marking up data using XML. We demonstrate several XML-derived markup languages, such as MathML, CML, MusicXML, RSS, **XML Schema** (for checking an XML document's grammar) and **XSLT (Extensible Stylesheet Language Transformations**, for transforming an XML document's data into XML or XHTML documents). We also introduce Web services and Water (an XML-native programming language that simplifies Web service programming). The reader interested in a deeper treatment of XML may want to consider our book, *XML How to Program*.

Chapter 21—Web Servers (IIS and Apache)
Through Chapter 20, we focus on the client side of Web-based applications. Chapters 21–27 and Chapters 33–37 (on the CD) focus on the server side, discussing many technologies crucial to implementing successful Web-based systems. A Web server is part of a **multitiered application**—sometimes referred to as an *n*-tier application. A three-tier application contains a **data tier (bottom tier)**, **middle tier** and **client tier (top tier)**. The bottom tier is an organization's database. The middle tier receives client requests from the top tier, references the data stored in the bottom tier and sends the requested information to the client. The client tier renders a Web page and executes any scripting commands contained in the Web page. A crucial decision in building Web-based systems is which Web server to use. The *Apache Web Server* and Microsoft *Internet Information Services (IIS)* are the two most popular Web servers used in industry. Each of these is an "industrial-strength" server designed to handle the high volumes of transactions that occur in real-world systems. They require considerable system resources and administrative support. In this chapter, we provide a brief introduction to IIS and Apache. We discuss how to request XHTML, ASP.NET, Perl, Python and PHP documents from these Web servers when using Internet Explorer. The chapter concludes by listing some additional Web servers that are available on the Internet. [*Note*: Server software is complex and evolving quickly. Our goal in this chapter is to give you a "handle" on setting up and using server-side software. Deitel & Associates, Inc., does not provide software support for these servers. We suggest that you browse the Web sites we list at the end of this chapter for organizations that may provide such support.]

Chapter 22—Database: SQL, MySQL, DBI and ADO.NET
The vast majority of an organizations' data is stored in databases. In this chapter, we introduce databases as well as **Structured Query Language (SQL)** for making database queries. The chapter also introduces *MySQL*, an open source, enterprise-level database server, and highlights several key features of this database server. We provide a list of data objects that access MySQL through various programmatic libraries called **database interfaces (DBIs)**. We specifically discuss DBIs for Perl, Python and PHP. In addition, a brief discussion of Microsoft's version of data storage, called **universal database access (UDA)**, is provided. A key UDA component is ActiveX Data Objects .NET (ADO.NET), which we introduce in this chapter and use in Chapter 23, ASP.NET. We list additional resources related to MySQL and Microsoft Access at `www.deitel.com/books/iw3HTP3/index.html`.

Chapter 23—ASP.NET

This chapter introduces Microsoft's ASP.NET (Active Server Pages .NET), the first of the seven server-side software development paradigms we discuss. ASP.NET applications can be programmed in a variety of languages including JScript .NET, C# and VBScript (VBScript is Chapter 32 on the CD). We use JScript .NET as our programming language. ASP.NET implements middle-tier business logic while taking advantage of Microsoft's .NET framework, which provides useful Web service components. In this chapter, we introduce the reader to **dynamic content generation** (i.e., the process by which a scripting language generates an XHTML document, an XML document etc.). Chapter examples include several server-side programming topics, such as writing text files, querying an Access database, Web security and Web services. Key examples include an ASP.NET document that allows users to create Web pages, a guestbook application, an ASP.NET document that displays information about the client's browser and a simple Web service. This is a crucial chapter for those readers who want to implement Web-based applications by using Microsoft technologies.

Chapter 24—Case Study: ASP.NET and XML

In this chapter, we build on the material presented in Chapter 23 by creating an online message forum using ASP.NET. Message forums are "virtual" bulletin boards in which users discuss a variety of topics. The case study presented allows users to post messages to an existing forum and to create new forums. Each forum's data are stored in XML documents that are dynamically manipulated using ASP.NET. This chapter integrates many of the technologies presented earlier in the book, including XHTML, CSS, ASP.NET, XML and XSLT. Chapter exercises ask the reader to modify the case study to delete individual messages from a forum and to delete individual forums.

Chapter 25—Perl and CGI[4] (Common Gateway Interface)

Historically, the most widely used server-side technology for developing Web-based applications has been Perl/CGI. Despite the emergence of newer technologies such as ASP.NET (Chapters 23 and 24), PHP (Chapter 26), ColdFusion (Chapter 27), Python (Chapter 35 on the CD), Java servlets (Chapter 36 on the CD) and JavaServer Pages (Chapter 37 on the CD), the Perl community is well entrenched, and Perl will remain popular for the foreseeable future. Chapter 25 presents an introduction to Perl/CGI, including many real-world, live-code examples and discussions, and demonstrations of some of the most recent features of each of these technologies. Key examples demonstrate how to interact with a MySQL database and how to use **regular expressions** (i.e., expressions that efficiently search strings for patterns of characters).

Chapter 26—PHP

In this chapter, we introduce PHP, another popular server-side scripting language for Web-based application development. Similar to Perl (Chapter 25) and Python (Chapter 35 on the CD), PHP has a large community of users and developers. We begin the chapter by introducing basic syntax, data types, operators and arrays, string processing and regular expressions. Chapter coverage includes form processing and business logic, connecting to a

[4] The reader interested in a deeper treatment of Perl and CGI may want to consider our book, *Perl How to Program*.

database and writing cookies. The chapter examples include a three-tier Web-based application that queries a MySQL database and a dynamically generated form that allows users to submit information to a mailing list database.

Chapter 27— Macromedia ColdFusion MX

This chapter describes the ColdFusion Markup Language (CFML), a popular server-side markup language for building dynamic Web sites. Macromedia's ColdFusion MX implements the CFML language and is used in this chapter to demonstrate basic and intermediate CFML capabilities. The language provides many markup tags, as well as support for functions, variables and expressions. Similar to ASP.NET and PHP, CFML enables developers to create database-driven Web sites, but with a quicker learning curve and shorter development time. The chapter includes a two-section case study on building an online bookstore with a shopping cart that tracks users' book selections.

Chapter 28—Multimedia: Audio, Video, Speech Synthesis and Recognition

This chapter focuses on the explosion of audio, video and speech technology appearing on the Web. We discuss adding sound, video and animated characters to Web pages (primarily using existing audio and video clips). Your first reaction may be a sense of caution, because these are complex technologies about which most readers have had little education. You quickly will see how easy it is to incorporate multimedia into Web pages and control multimedia components with Dynamic HTML. Multimedia files can be large. Some multimedia technologies require that an entire multimedia file be downloaded to the client before the audio or video begins playing. With **streaming audio** and **streaming video** technologies, audio and video can begin playing while the files are downloading, thus reducing delays. Streaming technologies are popular on the Web. This chapter demonstrates how to incorporate the **RealNetworks RealOne Player** into a Web page to receive streaming media. The chapter also includes an extensive set of Internet and Web resources that discuss interesting ways in which designers use multimedia-enhanced Web pages. This chapter introduces an exciting technology called **Microsoft Agent** for adding **interactive animated characters** to an XHTML document. **Agent characters** include *Peedy the Parrot*, *Genie*, *Merlin* and *Robby the Robot*, as well as those created by third-party developers. Each character allows users to interact with the application, using more natural human communication techniques such as speech. The agent characters accept mouse and keyboard interaction, speak and hear (i.e., they support speech synthesis and speech recognition). With these capabilities, your Web pages can speak to users and can actually respond to their voice commands! Microsoft Agent is included on the CD-ROM that accompanies this book. The chapter exercises ask the reader to create a karaoke machine and to incorporate an agent character into a Web page.

Chapter 29—Accessibility

Currently, the Web presents a challenge to individuals with disabilities. Multimedia-rich Web sites are difficult for text readers and other programs to interpret, especially for deaf users and users with visual impairments. To rectify this situation, the World Wide Web Consortium (W3C) launched the **Web Accessibility Initiative (WAI)**, which provides guidelines for making Web sites accessible to people with disabilities. This chapter provides a description of these guidelines, such as using the `alt` attribute of the `img` element to describe images and XHTML and using CSS to ensure that a page can be viewed on any type of display or reader. We discuss accessibility options available in Macromedia Flash, a predominantly visual and aural medium. We also introduce **VoiceXML** and **CallXML**, two technologies for increasing the accessibility of Web-based content. VoiceXML helps people with visual impairments to access Web content via speech synthesis and speech recognition. CallXML allows users with visual impairments to access Web-based content through a telephone. In the chapter exercises, readers create their own voicemail applications using CallXML.

Chapter 30 (on CD)—Dynamic HTML: Structured Graphics ActiveX Control
Although high-quality content is important to a Web site, it does not attract or maintain visitors' attention like eye-catching, animated graphics. This chapter explores the **Structured Graphics ActiveX Control** included with Internet Explorer. The Structured Graphics Control is a Web interface for the **DirectAnimation** subset of Microsoft's DirectX software. DirectAnimation is used in many popular video games and graphical applications. This control allows you to create complex graphics containing lines, shapes, textures and fills. In addition, scripting allows the graphics to be manipulated dynamically. The exercises at the end of the chapter ask the reader to create three-dimensional shapes and rotate them.

Chapter 31 (on CD)—Dynamic HTML: Path, Sequencer and Sprite ActiveX Controls
In this chapter, we discuss three additional DirectAnimation ActiveX controls available for Internet Explorer: the **Path Control**, the **Sequencer Control** and the **Sprite Control**. Each of these controls allows Web developers to add animated multimedia effects to Web pages. The Path Control allows the user to determine the positioning of elements on the screen. This is more elaborate than CSS absolute positioning, because the user can define lines, ovals and other shapes as paths along which objects move. Every aspect of motion is controllable through scripting. The Sequencer Control performs tasks at specified time intervals. This is useful for presentation-like effects, especially when used with the transitions discussed in Chapter 15. The Sprite Control creates Web animations. We also discuss, for comparison purposes, animated GIFs—another technique for producing Web-based animations.

Chapter 32 (on CD)—VBScript
Visual Basic Scripting Edition (VBScript) is a scripting language developed by Microsoft. Although it is not supported by many leading browsers, **plug-ins** help some of those browsers understand and process VBScript. Before the development of the Microsoft .NET Framework, however, VBScript was the most widely used language for writing Active Server Pages (ASP) prior to ASP.NET. This chapter prepares you to use VBScript on the client side in Microsoft communities and in Microsoft-based **intranets** (i.e., internal networks that use the same communications protocols as the Internet). This chapter will prepare readers to develop ASP pages in Chapters 33 and 34, as well as maintain existing ASP pages previously written in VBScript.

Chapter 33 (on CD)—Active Server Pages (ASP)
This chapter presents the original Active Server Pages technology, Microsoft's server-side technology before the advent of .NET. The chapter covers the same topics as Chapter 23 but uses ASP and VBScript instead of ASP.NET and JScript .NET.

Chapter 34 (on CD)—Case Study: Active Server Pages and XML
Chapter 34 presents the same message forum case study as Chapter 24 but uses ASP and VBScript instead of ASP.NET and JScript .NET. This chapter integrates many of the technologies presented earlier in the book, including XHTML, CSS, ASP, XML and XSLT.

Chapter 35 (on CD)—Python[5]

In this chapter, we introduce Python, an interpreted, cross-platform, object-oriented, general-purpose programming language. We begin by presenting basic syntax, data types, control statements and functions. We then introduce **lists** (i.e., data structures similar to a JavaScript array), **tuples** (i.e., immutable lists) and **dictionaries**, which are high-level data structures that store pairs of related data items. String processing and regular expressions are discussed, as is **exception handling**, which provides a structured mechanism for recovering from run-time errors. Chapter examples include implementing an XHTML registration form and showing how to use **cookies** (i.e., small text files written to the client machine). In addition, a three-tier Web-based example queries a MySQL database for author information.

Chapter 36 (on CD)—Servlets: Bonus for Java™ Developers

Java servlets represent a fifth popular way of building server-side Web-based applications. Servlets are written in Java (not JavaScript), which requires a substantial book-length treatment to learn. We do not teach Java in *Internet and World Wide Web How to Program, Third Edition*. This chapter (from our book *Java How to Program, 5/e*) is provided as a “bonus chapter” for readers familiar with Java. Readers who want to learn Java may want to consider reading this textbook.

Chapter 37 (on CD)—JavaServer Pages (JSP): Bonus for Java™ Developers

In this chapter (from our book *Java How to Program, 5/e*), we introduce **JavaServer Pages (JSP)**—an extension of Java servlet technology. JavaServer Pages enable Web-application programmers to create dynamic Web content, using familiar XML syntax and scripting with Java. Using JavaServer Pages, Web-application programmers can create **custom tag libraries** that encapsulate complex and dynamic functionality in XML tags. Web-page designers who are not familiar with Java can use these custom tag libraries to integrate information from databases, business-logic components and other resources into dynamically generated Web pages. This chapter is provided as a “bonus” chapter for readers familiar with Java. Readers who want to learn Java may want to consider reading our book *Java How to Program, Fifth Edition*.

Chapter 38 (on CD)—E-Business and E-Commerce

Chapter 38 explores the world of e-business and e-commerce. It begins by discussing the various business models associated with e-businesses. These include storefronts, auctions, portals, dynamic pricing, comparison shopping and demand-sensitive and name-your-price models. We also discuss the management and maintenance of an e-business, which includes advertising and marketing, accepting online payments, securing online transactions and understanding legal issues. We address such topics as branding, e-advertising, customer relationship management, e-wallets, micropayments, privacy and copyright. We also discuss security topics, including public-key cryptography, **Secure Socket Layer (SSL)** and wireless security. This chapter also discusses the emergence of XML and how it enables the standardization of business transactions worldwide. The final section of this chapter introduces wireless technology and mobile business concepts including wireless security, marketing, location-based services, privacy and more.

[5] The reader interested in a deeper treatment of Python may want to consider our book, *Python How to Program*.

Appendix A—XHTML Special Characters

This appendix shows many commonly used XHTML special characters, called **character entity references** by the World Wide Web Consortium (W3C).

Appendix B—XHTML Colors

This appendix explains how to create colors by using either color names or hexadecimal RGB values. Included is a table that matches colors to values.

Appendix C—Operator Precedence Chart

This appendix contains a JavaScript operator precedence chart.

Appendix D—ASCII Character Set

This appendix contains a table of the 128 ASCII alphanumeric symbols.

Appendix E—Number Systems

This appendix explains the binary, octal, decimal and hexadecimal number systems. It shows how to convert between bases and perform mathematical operations in each base.

Appendix F—Unicode®

This appendix introduces the **Unicode Standard**, an encoding scheme that assigns unique numeric values to the world's characters. It includes an XML-based example that uses Unicode encoding to print a welcome message in 10 different languages.

Appendix G (on CD)—Career Opportunities

The Internet presents valuable resources and services for job seekers and employers. Automatic search features allow employees to scan the Web for open positions. Employers also can find job candidates by using the Internet. This greatly reduces the amount of time spent preparing and reviewing resumes, as well as travel expenses for distance recruiting and interviewing. In this appendix, we explore career services on the Web from the perspectives of job seekers and employers. We introduce comprehensive job sites, industry-specific sites (including sites geared specifically for Java and wireless programmers) and contracting opportunities, as well as additional resources and career services designed to meet the needs of a variety of individuals.

Dependency Chart

Figure 1 illustrates the dependencies that exist between chapters in the book. We recommend studying the topics of the book in the order indicated by the arrows, though other orders are certainly possible. Some of the dependencies apply only to sections of chapters, so we advise readers to browse the material before designing a course of study. This book is widely used in courses that teach pure client side Web programming, courses that teach pure server side Web programming, and courses that mix and match some client side and some server side Web programming. Curricula for courses in Web programming are still evolving, and many syllabi exist to organize the material in this book. You can search the Web for "syllabus," "Internet," "Web" and "Deitel" to find syllabi currently used with this book. Readers only interested in studying server side technologies should begin with an understanding of how to build basic Web pages (using XHTML and CSS) and have a firm grasp of introductory object-based programming concepts (presented in the JavaScript

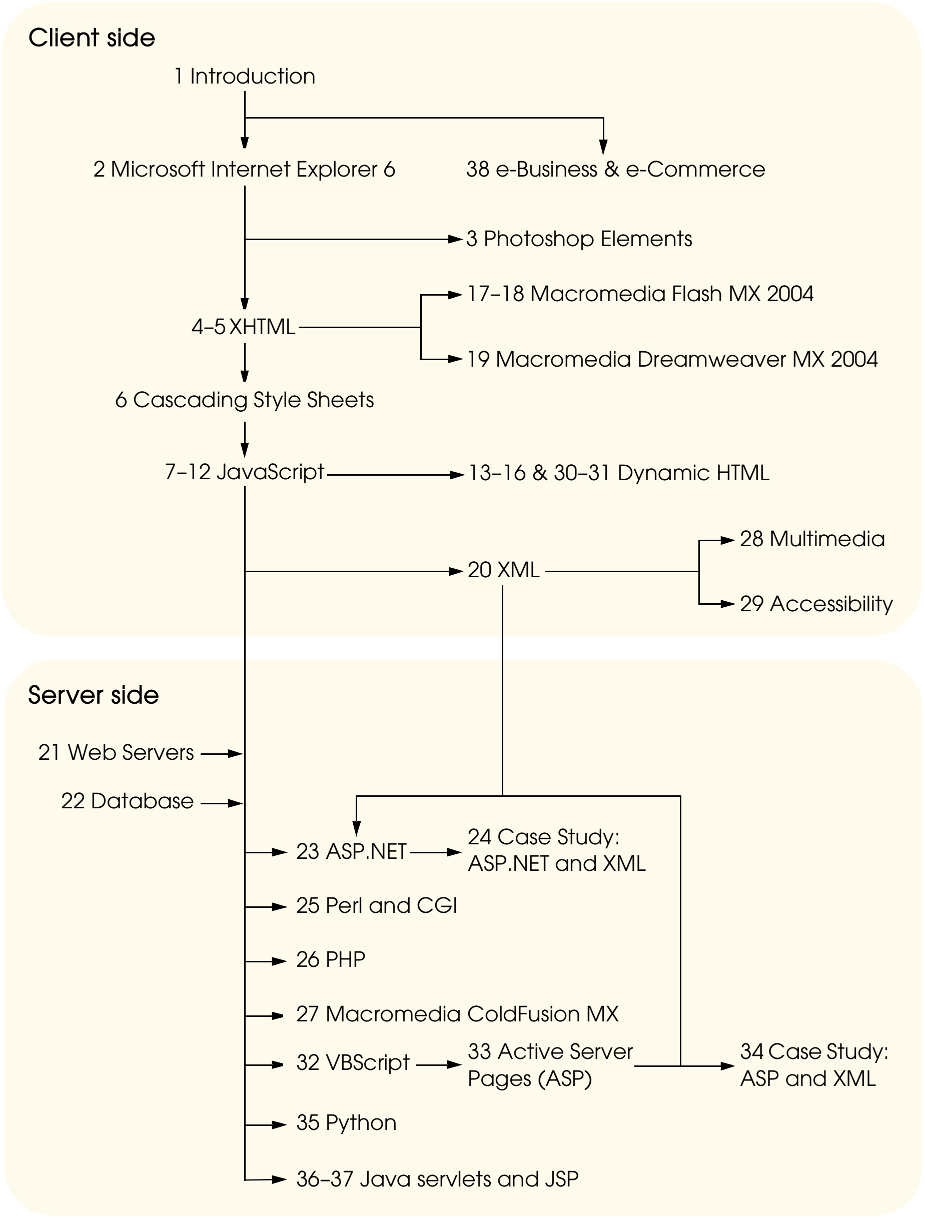
Client side
1 Introduction
2 Microsoft Internet Explorer 6
38 e-Business & e-Commerce
3 Photoshop Elements
17–18 Macromedia Flash MX 2004
4–5 XHTML
19 Macromedia Dreamweaver MX 2004
6 Cascading Style Sheets
7–12 JavaScript
13–16 & 30–31 Dynamic HTML
28 Multimedia
20 XML
29 Accessibility
Server side
21 Web Servers
22 Database
23 ASP.NET
24 Case Study: ASP.NET and XML
25 Perl and CGI
26 PHP
27 Macromedia ColdFusion MX
32 VBScript
33 Active Server Pages (ASP)
34 Case Study: ASP and XML
35 Python
36–37 Java servlets and JSP

Fig. 1 Flowchart illustrating the dependencies between chapters in the book.

chapters of this book). Note the positioning of XML in the dependency chart. XML could be taught in several places: part of a client side unit, beginning of a server side unit or split between the two. Also note that the book is designed to cover both non-Microsoft and Microsoft technologies (e.g., Dynamic HTML, ASP.NET) in a way that makes it convenient for instructors to include or omit the Microsoft material.

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. Many people at Deitel & Associates, Inc. devoted long hours to this project.

- Abbey Deitel, President
- Barbara Deitel, Chief Financial Officer
- Christi Kelsey, Director of Business Development

We would also like to thank the participants in our Deitel & Associates, Inc. College Internship Program[6] who have contributed to this project. Jonathan Emanuele is a graduate of Cornell University and holds a BS in Operations Research. He is currently a Masters Student in Computer Science at Cornell. He co-authored Chapter 27 and revised Chapters 4, 5, 6 and 22. David Zolotusky is a Computer Science major at Cornell University. He revised Chapters 7–12 and contributed to Chapters 21, 29 and 35. Eric Hsu is a Computer Science major with minors in Operations Research and Management Science at Cornell University. He revised Chapters 2, 3 and 32–34, and co-authored Chapters 20, 23 and 24. Andrew McNay is a Computer Science and Visual Arts major at Stanford University. He revised Chapters 13–17, 26, 28 and 30–31, and co-authored Chapters 18 and 19. Bei Zhou is a Computer Science major at Northeastern University. He assisted with the ancillary materials and tested all XHTML and Dynamic HTML examples in Internet Explorer 6 and Netscape 7.1. Jonathan Henry is a Computer Science major at Northeastern University. He assisted with the ancillary materials.

We are fortunate to have worked on this project with the talented and dedicated team of publishing professionals at Prentice Hall. We especially appreciate the extraordinary efforts of our Computer Science Editor, Kate Hargett and her boss and our mentor in publishing—Marcia Horton, Editorial Director of Prentice Hall's Engineering and Computer Science Division. Vince O'Brien, Tom Manshreck and John Lovell did a marvelous job managing the production of the book. Carole Anson and Geoffrey Cassar did a wonderful job on the design of the book, and Sarah Parker managed the publication of the book's extensive ancillary package.

We wish to acknowledge the efforts of our *Third Edition* reviewers and to give a special note of thanks to Carole Snyder and Jennifer Cappello of Prentice Hall who managed the extraordinary review effort for this book.

[6] The Deitel & Associates, Inc. College Internship Program is a highly competitive program that offers a limited number of salaried positions to Boston-area college students majoring in Computer Science, Information Technology, Marketing, Management and English. Students work at our corporate headquarters in Maynard, Massachusetts full-time in the summers and (for those attending college in the Boston area) part-time during the academic year. We also offer full-time internship positions for students interested in taking a semester off from school to gain industry experience. Regular full-time positions are available from time to time to college graduates. For more information, please contact our president, Abbey Deitel—`abbey.deitel@deitel.com`—and visit our Web site, `www.deitel.com`.

Internet & World Wide Web How to Program, Third Edition Reviewers
Americ Azevedo (University of California at Berkeley)
Tim Buntel (Macromedia, Inc.)
Sylvia Candelaria de Ram (Cognizor, LLC; HumanMarkup.org)
Wesley J. Chun (CyberWeb Consulting)
Marita Ellixson (Eglin AFB)
Jay Glynn (American General AIG)
James Greenwood (Poulternet)
Timothy Greer (Middle Tennessee State University)
James Huddleston (Independent Consultant)
Lynn Kyle (Yahoo!, Inc.)
Dan Livingston (Independent Consultant)
Oge Marques (Florida Atlantic University)
Mark Merkow (American Express Technologies)
Dan Moore (Independent Consultant)
George Semeczko (Royal & Sun Alliance Insurance Canada)
Deborah Shapiro (Cittone Institutes)
Matt Smith (Institute of Technology at Blanchardstown)
Narayana Rao Surapaneni (Patni Computer Systems Limited)
Stephanie Tauber (Tufts University)
Yateen Thakkar (Syntel India, Ltd.)
Cynthia Waddell (International Center for Disability Resources on the Internet)
Loran Walker (Lawrence Technological University)
Alnisa White (ILL Designs)

We wish to acknowledge again the efforts of our previous edition reviewers (some first edition and some second edition; the affiliations were current at the time of the review):

Richard Albright (University of Delaware)
Joan Aliprand (Unicode Consortium)
Race Bannon (Information Architects)
Kamaljit Bath (Microsoft)
Sunand Bhattacharya (ITT Technical Schools)
Paul Bohman (WebAIM)
Jason Bronfeld (Bristol-Myers Squibb Company)
Steve Burnett (RSA)
Carl Burnham (Southpoint.com)
Sylvia Candelaria de Ram (Editor, Python Journal)
Shane Carareo (Active State)
Kelly Carey (West Valley College)
Chris Constentino (Cisco Systems Inc., PTR Author)
Kevin Dorff (Honeywell)
Fred Drake (PythonLabs)
Bob DuCharme (XML Author)
Jonathan Earl (Technical Training and Consulting)
Amanda Farr (Virtual-FX.net)
Avi Finkel (WhizBang! Labs)

Seth Fogie (Donecker's, PTR Author)
Steven Franklin (UC Irvine)
Charles Fry (thesundancekid.org)
Jim Gips (Boston College)
Jesse Glick (NetBeans)
Phillip Gordon (Berkeley)
Christopher Haupt (Adobe)
Jesse Heines (UMass Lowell)
Shelly Heller (George Washington University)
Auda Hesham (CUNY)
Damon Houghland (Prentice Hall PTR Author "Essential WAP for Web Professionals")
Bryan Hughes (Adobe)
Jeff Isom (WebAIM)
John Jenkins (Unicode Consortium)
Simon Johnson (Shake Communications Pty Ltd)
Peter Jones (SUN Microsystems)
Alwyn Joy (Whiz Networks Pvt. Ltd.)
Ankur Kapoor (MIND UR Web)
David Kershaw (Art Technology)
Ryan Kuykendall (Amazon)
Hunt LaCascia (Engenius, Inc.)
Yves Lafon (W3C)
Daniel LaLiberte (W3C/Mosaic/NASA)
Elizabeth Lane Lawley (RIT)
Mike Leavy (Adobe)
Ze-Nian Li (Simon Frasier University)
Luby Liao (University of San Diego)
Wen Liu (ITT)
Maxim Loukianov (SoloMio Corp.)
Marc Loy (Consultant)
Dan Lynch (CyberCash)
Massimo Marchiori (W3C)
Rick McGowan (Unicode Consortium)
Julie McVicar (Oakland Community College)
Jasmine Merced (PerlArchive.com)
Mark Michael (Kings College)
Scott Mitchell (Consultant)
Dan Moore (XOR, Inc.)
Charles McCathie Neville (W3C)
Simon North (Synopsys)
Dr. Cyrus Peikari (VirusMD Corp., PTR Author)
Steven Pemberton (CWI, Amsterdam)
Shep Perkins (Fidelity Select Wireless Portfolio)
Corrin Pitcher (DePaul University)
Ashish Prakash (IBM)
Paul Prescod (Active State)

Keith Roberts (Prentice Hall PTR Author "Core CSS")
Rama Roberts (SUN Microsystems)
Chad Rolfs (Adobe)
Robert Rybaric (PRO-INFO Systems)
Arie Schlessinger (Columbia University)
Deb Shapiro (Computer Learning Centers)
Devan Shepherd (Shepherd Consulting Services)
Steve Smith (ASP Alliance)
M.G. Sriram (HelloBrian Corp.)
Dan Steinman (Consultant)
Sumanth Sukumar (IBM Transarc Labs [HTTP / AFS & DCE DFS])
Scott Tilley (University of California, Riverside)
Vadim Tkachenko (Sera Nova)
Guido Van Rossum (python.org)
Nic Van't Schip (vanschip.com)
William Vaughn (Microsoft)
Michael Wallent (Microsoft)
Susan Warren (Microsoft)
Ken Whistler (Sybase; Unicode Consortium)
Monty Widenius (MySQL)
Jesse Wilkins (Metalinear Media)
Michael Willett (wavesys.com)
Bernard Wong (Microsoft)
Ed Wright (Jet Propulsion Laboratory)
Stephen Wynne (IBM Transarc Labs/Carnegie Mellon University)

These reviewers scrutinized every aspect of the text and made countless suggestions for improving the accuracy and completeness of the presentation.

Contacting Deitel & Associates

We would sincerely appreciate your comments, criticisms, corrections and suggestions for improving the text. Please address all correspondence to:

    deitel@deitel.com

We will respond promptly.

Errata

We will post all errata for the *Third Edition* at `www.deitel.com`.

Customer Support

Please direct all software and installation questions to Pearson Education Technical Support:

- By phone: 1-800-677-6337
- By email: `media.support@pearsoned.com`
- On the Web: `247.prenhall.com`

Please direct all Internet & World Wide Web questions to `deitel@deitel.com`. We will respond promptly.

Welcome to the exciting world of the Internet & World Wide Web programming. We sincerely hope you enjoy learning with this book.

Dr. Harvey M. Deitel
Paul J. Deitel
Andrew B. Goldberg

About the Authors

Dr. Harvey M. Deitel, Chairman and Chief Strategy Officer of Deitel & Associates, Inc., has 42 years experience in the computing field, including extensive industry and academic experience. 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 now widely implemented in systems such as UNIX, Linux and Windows XP. 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 his son, Paul J. Deitel. He and Paul are the co-authors of several dozen books and multimedia packages and they are writing many more. With translations published in Japanese, German, Russian, Spanish, Traditional Chinese, Simplified Chinese, Korean, French, Polish, Italian, Portuguese, Greek, Urdu and Turkish, the Deitels' texts have earned international recognition. Dr. Deitel has delivered professional seminars to major corporations, government organizations and the military.

Paul J. Deitel, CEO and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of the MIT's Sloan School of Management, where he studied Information Technology. Through Deitel & Associates, Inc., he has delivered C, C++, Java, Internet and World Wide Web courses to industry clients, including IBM, Sun Microsystems, Dell, Lucent Technologies, Fidelity, NASA at the Kennedy Space Center, the National Severe Storm Laboratory, Compaq, White Sands Missile Range, Rogue Wave Software, Boeing, Stratus, Cambridge Technology Partners, Open Environment Corporation, One Wave, Hyperion Software, Adra Systems, Entergy, CableData Systems and many other organizations. He has lectured on C++ and Java for the Boston Chapter of the Association for Computing Machinery and has taught satellite-based Java courses through a cooperative venture of Deitel & Associates, Prentice Hall and the Technology Education Network. He and his father, Dr. Harvey M. Deitel, are the world's best-selling Computer Science textbook authors.

Andrew B. Goldberg is a recent graduate of Amherst College, where he earned a degree in Computer Science. His senior honors thesis, entitled *"Highly Constrained Sports Scheduling With Genetic Algorithms,"* brought together several of Andrew's research interests, including artificial intelligence, artificial life and genetic algorithms (i.e., algorithms that mimic evolution and the process of natural selection to solve problems). Andrew enjoys database-driven, dynamic Web programming and has developed applications for Amherst College and several independent Web sites. Andrew has contributed to other Deitel publications including *Operating Systems, Third Edition*.

About Deitel & Associates, Inc.

Deitel & Associates, Inc., is an internationally recognized corporate training and content-creation organization specializing in Internet/World Wide Web software technology, e-business/e-commerce software technology, object technology and computer programming languages education. The company provides instructor-led courses on Internet and World Wide Web programming, wireless Internet programming, object technology, and major programming languages and platforms, such as C, C++, Visual C++ .NET, Visual Basic .NET, C#, Java, Advanced Java, XML, Perl, Python and more. The founders of Deitel & Associates, Inc., are Dr. Harvey M. Deitel and Paul J. Deitel. The company's clients include many of the world's largest computer companies, government agencies, branches of the military and business organizations. Through its 27-year publishing partnership with Prentice Hall, Deitel & Associates, Inc. publishes leading-edge programming textbooks, professional books, interactive CD-based multimedia *Cyber Classrooms*, *Complete Training Courses*, Web-based training courses and course management systems e-content for popular CMSs such as WebCT™, Blackboard™ and CourseCompass^{SM}. Deitel & Associates, Inc., and the authors can be reached via e-mail at:

```
deitel@deitel.com
```

To learn more about Deitel & Associates, Inc., its publications and its worldwide corporate on-site training curriculum, see the last few pages of this book or visit:

```
www.deitel.com
```

Individuals wishing to purchase Deitel books, *Cyber Classrooms*, *Complete Training Courses* and Web-based training courses can do so through bookstores, online booksellers and:

```
www.deitel.com
www.prenhall.com/deitel
www.InformIT.com/deitel
www.InformIT.com/cyberclassrooms
```

Bulk orders by corporations and academic institutions should be placed directly with Prentice Hall. See the last few pages of this book for worldwide ordering instructions.