Preface

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

We wove a web in childhood,
A web of sunny air.
Charlotte Brontë

Welcome to C# and the world of Windows, Internet and World-Wide-Web programming with Visual Studio and the .NET platform! This book is the second in our new *.NET How to Program* series, which presents various leading-edge computing technologies in the context of the .NET platform.

C# is the next phase in the evolution of C and C++ and was developed expressly for Microsoft's .NET platform. C# provides the features that are most important to programmers, such as object-oriented programming, strings, graphics, graphical-user-interface (GUI) components, exception handling, multithreading, multimedia (audio, images, animation and video), file processing, prepackaged data structures, database processing, Internet and World-Wide-Web-based client/server networking and distributed computing. The language is appropriate for implementing Internet- and World-Wide-Web-based applications that seamlessly integrate with PC-based applications.

The .NET platform offers powerful capabilities for software development and deployment, including independence from a specific language or platform. Rather than requiring developers to learn a new programming language, programmers can contribute to the same software project, but write code using any (or several) of the .NET languages (such as C#, Visual Basic .NET, Visual C++ .NET and others) with which they are most competent. In addition to providing language independence, .NET extends program portability by enabling .NET applications to reside on, and communicate across, multiple platforms—thus facilitating the delivery of Web services over the Internet. The .NET platform enables Web-based applications to be distributed to consumer-electronic devices, such as cell

phones and personal digital assistants, as well as to desktop computers. The capabilities that Microsoft has incorporated into the .NET platform create a new software-development paradigm that will increase programmer productivity and decrease development time.

New Features in *C# How to Program*

This edition contains many new features and enhancements, including:

- ***Full-Color Presentation.*** This book is now in full color. Full color enables readers to see sample outputs as they would appear on a color monitor. Also, we now syntax color the C# code, similar to the way Visual Studio .NET colors the code in its editor window. Our syntax-coloring conventions are as follows:

```
comments appear in green
keywords appear in dark blue
literal values appear in light blue
text, class, method and variable names appear in black
errors and ASP .NET directives appear in red
```

- ***"Code Washing."*** This is our term for the process we use to format the programs in the book so that they have a carefully commented, open layout. The code appears in full color and is grouped into small, well-documented pieces. This greatly improves code readability—an especially important goal for us, considering that this book contains approximately 23,500 lines of code.
- ***Web Services and ASP .NET.*** Microsoft's .NET strategy embraces the Internet and Web as integral to the software development and deployment processes. Web services—a key technology in this strategy—enables information sharing, commerce and other interactions using standard Internet protocols and technologies, such as Hypertext Transfer Protocol (HTTP), Simple Object Access Protocol (SOAP) and Extensible Markup Language (XML). Web services enable programmers to package application functionality in a form that turns the Web into a library of reusable software components. In Chapter 21, ASP .NET and Web Services, we present a Web service that allows users to make airline seat reservations. In this example, a user accesses a Web page, chooses a seating option and submits the page to the Web server. The page then calls a Web service that checks seat availability. We also present information related to Web services in Appendix P, Crystal Reports® for Visual Studio® .NET, which discusses popular reporting software for database-intensive applications. Crystal Reports, which is integrated into Visual Studio .NET, provides the ability to expose a report as a Web service. The appendix provides introductory information and directs readers to a walkthrough of this process on the Crystal Decisions Web site (**`www.crystaldecisions.com/net`**).
- ***Web Forms, Web Controls and ASP .NET.*** Application developers must be able to create robust, scalable Web-based applications. The .NET platform architecture supports such applications. Microsoft's .NET server-side technology, Active Server Pages (ASP) .NET, allows programmers to build Web documents that respond to client requests. To enable interactive Web pages, server-side programs process information users input into HTML forms. ASP .NET is a significant de-

parture from previous versions of ASP, allowing developers to program Web-based applications using the powerful object-oriented languages of .NET. ASP .NET also provides enhanced visual programming capabilities, similar to those used in building Windows forms for desktop programs. Programmers can create Web pages visually, by dragging and dropping Web controls onto Web forms. Chapter 20, ASP .NET, Web Forms and Web Controls, introduces these powerful technologies.

- ***Object-Oriented Programming*.** Object-oriented programming is the most widely employed technique for developing robust, reusable software, and C# offers enhanced object-oriented programming features. This text offers a rich presentation of object-oriented programming. Chapter 8, Object-Based Programming, introduces how to create classes and objects. These concepts are extended in Chapter 9, Object-Oriented Programming: Inheritance, which discusses how programmers can create new classes that "absorb" the capabilities of existing classes. Chapter 10, Object-Oriented Programming: Polymorphism, familiarizes the reader with the crucial concepts of polymorphism, abstract classes, concrete classes and interfaces, which facilitate powerful manipulations among objects belonging to an inheritance hierarchy.
- ***XML*.** Use of Extensible Markup Language (XML) is exploding in the software-development industry, the e-business and e-commerce communities, and is pervasive throughout the .NET platform. Because XML is a platform-independent technology for describing data and for creating markup languages, XML's data portability integrates well with C#'s portable applications and services. Chapter 18, Extensible Markup Language (XML), introduces XML. In this chapter, we introduce basic XML markup and discuss the technologies such as DTDs and Schema, which are used to validate XML documents' contents. We also explain how to programmatically manipulate XML documents using the Document Object Model (DOM™) and how to transform XML documents into other types of documents via Extensible Stylesheet Language Transformations (XSLT).
- ***Multithreading*.** Computers enable us to perform many tasks in parallel (or concurrently), such as printing documents, downloading files from a network and surfing the Web. Multithreading is the technology through which programmers can develop applications that perform concurrent tasks. Historically, a computer has contained a single, expensive processor, which its operating system would share among all applications. Today, processors are becoming so inexpensive that it is possible to build affordable computers that contain many processors that work in parallel—such computers are called multiprocessors. Multithreading is effective on both single-processor and multiprocessor systems. C#'s multithreading capabilities make the platform and its related technologies better prepared to deal with today's sophisticated multimedia-intensive, database-intensive, network-based, multiprocessor-based distributed applications. Chapter 14, Multithreading, provides a detailed discussion of multithreading.
- ***ADO .NET*.** Databases store vast amounts of information that individuals and organizations must access to conduct business. As an evolution of Microsoft's ActiveX Data Objects (ADO), ADO .NET represents a new approach for building

applications that interact with databases. ADO .NET uses XML and an enhanced object model to provide developers with the tools they need to access and manipulate databases for large-scale, extensible, mission-critical multi-tier applications. Chapter 19, Database, SQL and ADO .NET, details the capabilities of ADO .NET and the Structured Query Language (SQL) to manipulate databases.

- ***Visual Studio .NET Debugger*.** Debuggers are programs that help programmers find and correct logic errors in program code. Visual Studio .NET contains a powerful debugging tool that allows programmers to analyze their programs line-by-line as those programs execute. In Appendix D, Visual Studio .NET Debugger, we explain how to use key debugger features, such as setting breakpoints and "watches," stepping into and out of procedures, and examining the procedure call stack.
- ***COM (Component Object Model) Integration*.** Prior to the introduction of .NET, many organizations spent tremendous amounts of time and money creating reusable software components called COM components, which include ActiveX® controls and ActiveX DLLs (dynamic link libraries) for Windows applications. In Appendix H, COM Integration, we discuss some of the tools available in Visual Studio .NET for integrating these legacy components into .NET applications. This integration allows programmers to use existing sets of COM-based controls with .NET components.
- ***XML Documentation.*** Documenting program code is crucial for software development, because different programmers often work on an application during the software's lifecycle, which usually includes multiple versions and can span many years. If programmers document software code and methods, other programmers working on the application can learn and understand the logic underlying the code, thus saving time and avoiding misunderstandings. To automate documenting programs, Visual Studio .NET provides an XML tool for C# programmers. Appendix E, XML Documentation, explains how a programmer can insert comments in the code, which produces a separate file providing the code documentation.
- ***Career Opportunities*.** Appendix C, Career Opportunities, introduces career services available on the Internet. We explore online career services from both the employer's and employee's perspectives. We list many Web sites at which you can submit applications, search for jobs and review applicants (if you are interested in hiring someone). We also review services that build recruiting pages directly into e-businesses. One of our reviewers told us that he had used the Internet as a primary tool in a recent job search, and that this appendix would have helped him expand his search dramatically.
- ***Unicode*.** As computer systems evolved worldwide, computer vendors developed numeric representations of character sets and special symbols for the local languages spoken in different countries. In some cases, different representations were developed for the same languages. Such disparate character sets hindered communication among computer systems. C# supports the *Unicode Standard* (maintained by a non-profit organization called the *Unicode Consortium*), which maintains a single character set that specifies unique numeric values for characters and special symbols in most of the world's languages. Appendix G, Unicode, discusses the standard, overviews the Unicode Consortium Web site (**`www.uni-`**

`code.org`) and presents a C# application that displays "Welcome to Unicode!" in several languages.

- ***XHTML.*** The World Wide Web Consortium (W3C) has declared HTML to be a legacy technology that will undergo no further development. HTML is being replaced by the Extensible Hypertext Markup Language (XHTML)—an XML-based technology that is rapidly becoming the standard for describing Web content. We use XHTML in Chapter 18, Extensible Markup Language (XML), and offer an introduction to the technology in Appendix K, Introduction to XHTML: Part 1, and Appendix L, Introduction to XHTML: Part 2. These appendices overview headers, images, lists, image maps and other features of this emerging markup language. (We also present a treatment of HTML in Appendices I and J, because ASP .NET, used in Chapters 20 and 21, generates HTML content).
- ***Accessibility.*** Although the World Wide Web has become an important part of many people's lives, the medium currently presents many challenges to people with disabilities. Individuals with hearing and visual impairments, in particular, have difficulty accessing multimedia-rich Web sites. In an attempt to improve 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. Chapter 24, Accessibility, describes these guidelines and highlights various products and services designed to improve the Web-browsing experiences of individuals with disabilities. For example, the chapter introduces VoiceXML and CallXML—two XML-based technologies for increasing the accessibility of Web-based content for people with visual impairments.
- ***Bit Manipulation.*** Computers work with data in the form of binary digits, or bits, which can assume the values 1 or 0. Computer circuitry performs various simple bit manipulations, such as examining the value of a bit, setting the value of a bit and reversing a bit (from 1 to 0 or from 0 to 1). Operating systems, test-equipment, networking software and many other kinds of software require that programs communicate "directly with the hardware" by using bit manipulation. Appendix O, Bit Manipulation, overviews the bit manipulation capabilities that the .NET Framework provides.

Some Notes to Instructors

Students Enjoy Learning a Leading-Edge Language

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 such courses represents the basic principles of programming, concentrating on the effective use of data types, control structures, arrays and functions. Our experience has been that students handle the material in this book in about the same way that they handle other introductory and intermediate programming courses. There is one noticeable difference, though: Students are highly motivated by the fact that they are learning a leading-edge language, C#, and a leading-edge programming paradigm (object-oriented programming) that will be immediately useful to them as they enter the business world. This increases their enthusiasm for the material—which is essential when you consider that there is much more to learn in a C# course now that students must master both the base language

and substantial class libraries as well. Although C# is a new language that may require programmers to revamp their skills, programmers will be motivated to do so because of the powerful range of capabilities that Microsoft is offering in its .NET initiative.

A World of Object Orientation

In the late 1990s, universities were still emphasizing procedural programming. The leading-edge courses were using object-oriented C++, but these courses generally mixed a substantial amount of procedural programming with object-oriented programming—something that C++ lets programmers do. Many instructors now are emphasizing a pure object-oriented programming approach. This book—the first edition of *C# How to Program* and the second text in our .NET series—takes a predominantly object-oriented approach because of the object orientation provided in C#.

Focus of the Book

Our goal was clear: Produce a C# textbook for introductory university-level courses in computer programming aimed at students with little or no programming experience, yet offer the depth and the rigorous treatment of theory and practice demanded by both professionals and students in traditional, upper-level programming courses. To meet these objectives, we produced a comprehensive book that patiently teaches the principles of computer programming and of the C# language, including control structures, object-oriented programming, C# class libraries, graphical-user-interface concepts, event-driven programming and more. After mastering the material in this book, students will be well-prepared to program in C# and to employ the capabilities of the .NET platform.

Multimedia-Intensive Communications

People want to communicate. Sure, they have been communicating since the dawn of civilization, but the potential for information exchange has increased dramatically with the evolution of various technologies. Until recently, even computer communications were limited mostly to digits, alphabetic characters and special characters. The current wave of communication technology involves the distribution of multimedia—people enjoy using applications that transmit color pictures, animations, voices, audio clips and even full-motion color video over the Internet. At some point, we will insist on three-dimensional, moving-image transmission.

There have been predictions that the Internet will eventually replace radio and television as we know them today. Similarly, it is not hard to imagine newspapers, magazines and books delivered to “the palm of your hand” (or even to special eyeglasses) via wireless communications. Many newspapers and magazines already offer Web-based versions, and some of these services have spread to the wireless world. When cellular phones were first introduced, they were large and cumbersome. Today, they are small devices that fit in our pockets, and many are Internet-enabled. Given the current rate of advancement, wireless technology soon could offer enhanced streaming-video and graphics-packed services, such as video conference calls and high-power, multi-player video games.

Teaching Approach

C# How to Program contains a rich collection of examples, exercises and projects drawn from many fields and designed to provide students with a chance to solve interesting, real-world problems. The code examples in this text have been tested on Windows 2000 and

Windows XP. The book concentrates on the principles of good software engineering, and stresses program clarity. We are educators who teach edge-of-the-practice topics in industry classrooms worldwide. We avoid arcane terminology and syntax specifications in favor of teaching by example. The text emphasizes good pedagogy.[1]

LIVE-CODE™ Teaching Approach

C# How to Program is loaded with numerous LIVE-CODE™ examples. This style exemplifies the way we teach and write about programming and is the focus of our multimedia *Cyber Classrooms* and Web-based training courses. Each new concept is presented in the context of a complete, working example that is immediately followed by one or more windows showing the program's input/output dialog. 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 entering and running them on a computer.

World Wide Web Access

All of the examples for *C# How to Program* (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 these downloads are free. We suggest downloading all the examples, then running each program as you read the corresponding text. Make changes to the examples and immediately see the effects of those changes—a great way to learn programming. Each set of instructions assumes that the user is running Windows 2000 or Windows XP and is using Microsoft's Internet Information Services (IIS). Additional setup instructions for Web servers and other software can be found at our Web sites along with the examples. [*Note:* This is copyrighted material. Feel free to use it as you study, but you may not republish any portion of it in any form without explicit permission from Prentice Hall and the authors.]

Visual Studio .NET, which includes C#, can be purchased and downloaded from Microsoft. Three different versions of Visual Studio .NET are available—Enterprise, Professional and Academic. Visit **`developerstore.com/devstore/`** for more details and to order. If you are a member of the Microsoft Developer Network, visit **`msdn.microsoft.com/default.asp`**.

Objectives

Each chapter begins with objectives that inform students of what to expect and give them an opportunity, after reading the chapter, to determine whether they have met the intended goals. The objectives serve as confidence builders and as a source of positive reinforcement.

Quotations

The chapter objectives are followed by sets of quotations. Some are humorous, some are philosophical and some offer interesting insights. We have found that students enjoy relat-

1. We use fonts to distinguish between Visual Studio .NET's Integrated Development Environment (IDE) features (such as menu names and menu items) and other elements that appear in the IDE. Our convention is to emphasize IDE features in a sans-serif bold Helvetica font (e.g., **Project** menu) and to emphasize program text in a serif bold Courier font (e.g., **`bool x = true;`**).

ing the quotations to the chapter material. Many of the quotations are worth a "second look" *after* you read each chapter.

Outline
The chapter outline enables students to approach the material in top-down fashion. Along with the chapter objectives, the outline helps students anticipate future topics and set a comfortable and effective learning pace.

Approximately 23,500 Lines of Code in 204 Example Programs (with Program Outputs)
We present C# features in the context of complete, working C# programs. The programs range in size from just a few lines of code to substantial examples containing several hundred lines of code. All examples are available on the CD that accompanies the book or as downloads from our Web site, **`www.deitel.com`**.

607 Illustrations/Figures
An abundance of charts, line drawings and program outputs is included. The discussion of control structures, for example, features carefully drawn flowcharts. [*Note:* We do not teach flowcharting as a program-development tool, but we do use a brief, flowchart-oriented presentation to explain the precise operation of each C# control structure.]

509 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*, *Testing and Debugging Tips*, *Performance Tips*, *Portability Tips*, *Software Engineering Observations* and *Look-and-Feel Observations*. These tips and practices represent the best the authors have gleaned from a combined seven 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.

91 Good Programming Practices

Good Programming Practices *are tips that call attention to techniques that will help students produce better programs. When we teach introductory courses to nonprogrammers, we state that the "buzzword" for each course is "clarity," and we tell the students that we will highlight (in these Good Programming Practices) techniques for writing programs that are clearer, more understandable and more maintainable.*

165 Common Programming Errors

Students learning a language—especially in their first programming course—tend to make certain kinds of errors frequently. Pointing out these Common Programming Errors *reduces the likelihood that students will make the same mistakes. It also shortens long lines outside instructors' offices during office hours!*

44 Testing and Debugging Tips

When we first designed this "tip type," we thought the tips would contain suggestions strictly for exposing bugs and removing them from programs. In fact, many of the tips describe aspects of C# that prevent "bugs" from getting into programs in the first place, thus simplifying the testing and debugging process.

57 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 programs that run the fastest, use the least memory, require the smallest number of keystrokes or dazzle in other ways. Students really care about performance and they want to know what they can do to "turbo charge" their programs. We have included 57 Performance Tips *that highlight opportunities for improving program performance—making programs run faster or minimizing the amount of memory that they occupy.*

16 Portability Tips

We include Portability Tips *to help students write portable code and to provide insights on how C# achieves its high degree of portability.*

115 Software Engineering Observations

The object-oriented programming paradigm necessitates a complete rethinking of the way we build software systems. C# is an effective language for achieving good software engineering. The Software Engineering Observations *highlight architectural and design issues that affect the construction of software systems, especially large-scale systems. Much of what the student learns here will be useful in upper-level courses and in industry as the student begins to work with large, complex real-world systems.*

21 Look-and-Feel Observations

We provide Look-and-Feel Observations *to highlight graphical-user-interface conventions. These observations help students design attractive, user-friendly graphical user interfaces that conform to industry norms.*

Summary (1277 Summary bullets)
Each chapter ends with additional pedagogical devices. We present a thorough, bullet-list-style summary of the chapter. On average, there are 39 summary bullets per chapter. This helps the students review and reinforce key concepts.

Terminology (2932 Terms)
We include an alphabetized list of the important terms defined in the chapter in a *Terminology* section. Again, this serves as further reinforcement. On average, there are 89 terms per chapter. Each term also appears in the index, so the student can locate terms and definitions quickly.

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

367 Exercises (Solutions in Instructor's Manual; Count Includes Separate Parts)
Each chapter concludes with a substantial set of exercises that involve simple recall of important terminology and concepts; writing individual C# statements; writing small portions of C# methods and classes; writing complete C# methods, classes and applications; and writing major projects. These exercises cover a wide variety of topics, enabling instructors to tailor their courses to the unique needs of their audiences and to vary course assignments

each semester. Instructors can use the exercises to form homework assignments, short quizzes and major examinations. The solutions for the exercises are included in the *Instructor's Manual* and on the disks *available only to instructors* through their Prentice-Hall representatives. **[NOTE: Please do not write to us requesting the instructor's manual**. **Distribution of this publication is strictly limited to college professors teaching from the book. Instructors may obtain the solutions manual from their regular Prentice Hall representatives**. **We regret that we cannot provide the solutions to professionals.]** Solutions to approximately half the exercises are included on the *C# Multimedia Cyber Classroom* CD-ROM (available in April 2002 at **www.InformIT.com/cyberclassrooms**; also see the last few pages of this book or visit **www.deitel.com** for ordering instructions). Also available in April 2002 is the boxed product, *The Complete C# Training Course*, which includes both our textbook, *C# How to Program* and the *C# Multimedia Cyber Classroom*. All of our *Complete Training Course* products are available at bookstores and online booksellers, including **www.InformIT.com**.

Approximately 5,420 Index Entries (with approximately 6,450 Page References)
We have included an extensive Index at the back of the book. Using this resource, students can search for any term or concept by keyword. The Index is especially useful to practicing programmers who use the book as a reference. Each of the 2932 terms in the Terminology sections appears in the Index (along with many more index items from each chapter). Students can use the index in conjunction with the Terminology sections to ensure that they have covered the key material in each chapter.

"Double Indexing" of All C# LIVE-CODE™ Examples
C# How to Program has 204 LIVE-CODE™ examples, which we have "double indexed." For every C# source-code program in the book, we took the file name with the **.cs** extension, such as **ChessGame.cs**, and indexed it both alphabetically (in this case, under "C") and as a subindex item under "Examples." This makes it easier to find examples using particular features.

C# Multimedia Cyber Classroom and The Complete C# Training Course

We have prepared an interactive, CD-ROM-based, software version of *C# How to Program*, called the *C# Multimedia Cyber Classroom.* This resource is loaded with e-Learning features that are ideal for both learning and reference. The *Cyber Classroom* is packaged with the textbook at a discount in *The Complete C# Training Course*. If you already have the book and would like to purchase the *C# Multimedia Cyber Classroom* separately, please visit **www.InformIT.com/cyberclassrooms**. The ISBN number for the *C# Multimedia Cyber Classroom* is 0-13-064587-7. All Deitel™ *Cyber Classrooms* are available in CD-ROM and Web-based training formats.

The CD provides an introduction in which the authors overview the *Cyber Classroom*'s features. The textbook's 204 LIVE-CODE™ example C# programs truly "come alive" in the *Cyber Classroom*. If you are viewing a program and want to execute it, you simply click the lightning-bolt icon, and the program will run. You immediately will see—and hear, when working with audio-based multimedia programs—the program's outputs. If you want to modify a program and see the effects of your changes, simply click the

floppy-disk icon that causes the source code to be "lifted off" the CD and "dropped into" one of your own directories so you can edit the text, recompile the program and try out your new version. Click the audio icon, and one of the authors will discuss the program and "walk you through" the code.

The *Cyber Classroom* also provides navigational aids, including extensive hyperlinking. The *Cyber Classroom* is browser based, so it remembers sections that you have visited recently and allows you to move forward or backward among these sections. The thousands of index entries are hyperlinked to their text occurrences. Furthermore, when you key in a term using the "find" feature, the *Cyber Classroom* will locate occurrences of that term throughout the text. The Table of Contents entries are "hot," so clicking a chapter name takes you immediately to that chapter.

Students like the fact that solutions to approximately half the exercises in the book are included with the *Cyber Classroom*. Studying and running these extra programs is a great way for students to enhance their learning experience.

Students and professional users of our *Cyber Classrooms* tell us that they like the interactivity and that the *Cyber Classroom* is an effective reference due to its extensive hyperlinking and other navigational features. We received an e-mail from a person who said that he lives "in the boonies" and cannot take a live course at a university, so the *Cyber Classroom* provided an ideal solution to his educational needs.

Professors tell us that their students enjoy using the *Cyber Classroom* and spend more time on the courses and master more of the material than in textbook-only courses. For a complete list of the available and forthcoming *Cyber Classrooms* and *Complete Training Courses*, see the *Deitel™ Series* page at the beginning of this book, the product listing and ordering information at the end of this book or visit **`www.deitel.com`**, **`www.prenhall.com/deitel`** and **`www.InformIT.com/deitel`**.

Deitel e-Learning Initiatives

e-Books and Support for Wireless Devices

Wireless devices will play 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 two years, more people will access the Internet through wireless devices than through desktop computers. Deitel & Associates, Inc., is committed to wireless accessibility and has recently published *Wireless Internet & Mobile Business How to Program*. To fulfill the needs of a wide range of customers, we currently are developing our content both in traditional print formats and in newly developed electronic formats, such as e-books so that students and professors can access content virtually anytime, anywhere. Visit **`www.deitel.com`** for periodic updates on this initiative.

e-Matter

Deitel & Associates, Inc., is partnering with Prentice Hall's parent company, Pearson PLC, and its information technology Web site, **`InformIT.com`**, to launch the Deitel e-Matter series at **`www.InformIT.com/deitel`**. This series will provide professors, students and professionals with an additional source of information on specific programming topics. e-Matter consists of stand-alone sections taken from published texts, forthcoming texts or pieces written during the Deitel research-and-development process. Developing e-Matter based on pre-publication books allows us to offer significant amounts of the material to ear-

ly adopters for use in courses. Some possible C# e-Matter titles we are considering include *Object-Based Programming and Object-Oriented Programming in C#*; *Graphical User Interface Programming in C#*; *Multithreading in C#*; *ASP .NET and Web Forms: A C# View;* and *ASP .NET and Web Services: A C# View.*

Course Management Systems: WebCT, Blackboard, and CourseCompass
We are working with Prentice Hall to integrate our *How to Program Series* courseware into three Course Management Systems: WebCT, Blackboard™ and CourseCompass. These Course Management Systems enable instructors to create, manage and use sophisticated Web-based educational programs. Course Management System features include course customization (such as posting contact information, policies, syllabi, announcements, assignments, grades, performance evaluations and progress tracking), class and student management tools, a gradebook, reporting tools, communication tools (such as chat rooms), a whiteboard, document sharing, bulletin boards and more. Instructors can use these products to communicate with their students, create online quizzes and tests from questions directly linked to the text and automatically grade and track test results. For more information about these upcoming products, visit **`www.deitel.com/whatsnew.html`**. For demonstrations of existing WebCT, Blackboard and CourseCompass courses, visit **`cms.prenhall.com/WebCT`**, **`cms.prenhall.com/Blackboard`** and **`cms.prenhall.com/CourseCompass`**, respectively.

Deitel and InformIT Newsletters

Deitel Column in the InformIT Newsletters
Deitel & Associates, Inc., contributes a weekly column to the popular *InformIT* newsletter, currently subscribed to by more than 800,000 IT professionals worldwide. For opt-in registration, visit **`www.InformIT.com`**.

Deitel Newsletter
Our own free, opt-in newsletter includes commentary on industry trends and developments, links to articles and resources from our published books and upcoming publications, information on future publications, product-release schedules and more. For opt-in registration, visit **`www.deitel.com`**.

The Deitel .NET Series

Deitel & Associates, Inc., is making a major commitment to .NET programming through the launch of our .NET Series. *C# .NET How to Program* and *Visual Basic .NET How to Program, Second Edition* are the first books in this new series. We intend to follow these books with *Advanced C# How to Program* and *Advanced Visual Basic .NET How to Program*, which will be published in December 2002. We also plan to publish *Visual C++ .NET How to Program* in July 2002, followed by *Advanced Visual C++ .NET How to Program* in July 2003.

Advanced C# How to Program

C# How to Program covers introductory through intermediate-level C# programming topics, as well as core programming fundamentals. By contrast, our upcoming textbook *Ad-*

vanced C# How to Program will be geared toward experienced C# developers. This new book will cover enterprise-level programming topics, including: Creating multi-tier, database intensive ASP .NET applications using ADO .NET and XML; constructing custom Windows controls; developing custom Web controls; and building Windows services. The book also will include more in-depth explanations of object-oriented programming (with the UML), ADO .NET, XML Web services, wireless programming and security. *Advanced C# How to Program* will be published in December 2002.

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

- Sean E. Santry, a graduate of Boston College with degrees in Computer Science and Philosophy, Director of Software Development at Deitel & Associates, Inc., and co-author of *Advanced Java 2 Platform How to Program*, contributed to Chapters 1–10, 12–13 and 18–23.
- Matthew R. Kowalewski, a graduate of Bentley College with a degree in Accounting Informations Systems, is the Director of Wireless Development at Deitel & Associates, Inc. He contributed to Chapters 19–20, Appendices B, F, I–N, P and edited the Index.
- Jonathan Gadzik, a graduate of the Columbia University School of Engineering and Applied Science with a major in Computer Science, co-authored Chapter 17 and contributed to Chapters 9, 22 and Appendices D and E.
- Kyle Lomelí, a graduate of Oberlin College with a degree in Computer Science and a minor in East Asian Studies, contributed to Chapters 11, 14–15, 19 and 24.
- Lauren Trees, a graduate if Brown University in English, edited the entire manuscript for smoothness, clarity and effectiveness of presentation; she also co-authored the Preface, Chapter 1 and Appendix P.
- Rashmi Jayaprakash, a graduate of Boston University with a major in Computer Science, co-authored Chapter 24 and Appendix G.
- Laura Treibick, a graduate of the University of Colorado at Boulder with a degree in Photography and Multimedia, is Director of Multimedia at Deitel & Associates, Inc. She contributed to Chapter 16 and enhanced many of the graphics throughout the text.
- Betsy DuWaldt, a graduate of Metropolitan State College of Denver with a major in Technical Communications (Writing and Editing emphasis) and a minor in Computer Information Systems, is Editorial Director at Deitel & Associates, Inc. She co-authored the Preface, Chapter 1 and Appendix P and managed the permissions process for the book.
- Barbara Deitel applied the copy edits to the manuscript. She did this in parallel with handling her extensive financial and administrative responsibilities at Deitel

& Associates, Inc., which include serving as Chief Financial Officer. [Everyone at the company works on book content.]

- Abbey Deitel, a graduate of Carnegie Mellon University's Industrial Management Program and President of Deitel & Associates, Inc., recruited 40 additional full-time employees and interns during 2001. She also leased, equipped and furnished our second building to create the work environment from which *C# How to Program* and our other year 2001 publications were produced. She suggested the title for the *How to Program* series, and edited this preface and several of the book's chapters.

We would also like to thank the participants in the Deitel & Associates, Inc., College Internship Program.[2]

- Jeffrey Hamm, a sophomore at Northeastern University in Computer Science, co-authored Chapters 16, 18, 20–21 and Appendices D and H.
- Kalid Azad, a sophomore at Princeton University in Computer Science, contributed to Chapters 1, 2, 12–13, 16 and Appendix D. He created PowerPoint-slide ancillaries for Chapters 1–7 and researched Visual Studio .NET and Microsoft's .NET initiative.
- Christopher Cassa, a junior at MIT in Computer Science, contributed to Chapters 3–7 and 18.
- David Tuttle, a senior at Harvard in Computer Science, contributed to Chapters 8, 18–19 and 24 and coded examples for Chapters 3–6, 7, 11,16–17,19, 23 and 26.
- Ori Schwartz, a sophomore at Boston University in Computer Science, produced solutions for all the chapters and contributed to Chapter 16.
- Thiago Lucas da Silva, a sophomore at Northeastern University in Computer Science, tested all the programming examples through the various beta releases and release candidates of Visual Studio .NET.
- Matthew Rubino, a sophomore at Northeastern University in Computer Science, created ancillary materials for the entire book.
- Elizabeth Rockett, a senior in English at Princeton University, edited 1-3, 7–8, 14, 17 and 19-24.
- Barbara Strauss, a senior in English at Brandeis University, edited Chapters 1–6, 9–13 and 18–24.
- Christina Carney, a senior in Psychology and Business at Framingham State College, helped with the Preface.

2. The *Deitel & Associates, Inc. College Internship Program* 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 Sudbury, 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 to college graduates. For more information about this competitive program, please contact Abbey Deitel at **deitel@deitel.com** and visit our Web site, **www.deitel.com**.

- Reshma Khilnani, a junior in Computer Science and Mathematics at Massachusetts Institute of Technology, contributed to Chapter 18 and Appendix E.
- Brian Foster, a sophomore at Northeastern University in Computer Science, helped with the Preface and Bibliography.
- Mike Preshman, a sophomore at Northeastern University with a major in Computer Science and minors in Electrical Engineering and Math, helped with the Bibliography.

We are fortunate to have been able to work 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, Petra Recter and her boss—our mentor in publishing—Marcia Horton, Editorial Director of Prentice-Hall's Engineering and Computer Science Division. Camille Trentacoste and her boss Vince O'Brien did a marvelous job managing the production of the book. Sarah Burrows handled editorial responsibilities on the book's extensive ancillary package.

The *C# Multimedia Cyber Classroom* was developed in parallel with *C# How to Program*. We sincerely appreciate the "new media" insight, savvy and technical expertise of our electronic-media editors, Mark Taub and Karen McLean. They and project manager Mike Ruel did a wonderful job bringing the *C# Multimedia Cyber Classroom* and *The Complete C# Training Course* to publication.

We owe special thanks to the creativity of Tamara Newnam (**`smart_art@earthlink.net`**), who produced the art work for our programming-tip icons and for the cover. She created the delightful creature who shares with you the book's programming tips. Barbara Deitel and Abbey Deitel contributed the bugs' names for the front cover.

During the development of this manuscript, we were fortunate to have had two universities—the Massachusetts Institute of Technology and Yale University—beta-test the book in the Fall 2001 semester. MIT Professor John Williams used the text to teach the graduate-level class, *Web System Architecting—Part I: Programming Clients and Web Services Using C# and .NET*, for the Off-Campus Advanced Study Program. Chris Cassa, a summer 2001 intern at Deitel & Associates, Inc., was the teaching fellow for the class. Yale Professor Paul Hudak used the manuscript for an *Introduction to Programming* class, which taught object-oriented programming languages. We would like to thank Professor Williams, Professor Hudak and Chris for their contributions. The feedback we received was crucial to fine-tuning this text.

We wish to acknowledge the efforts of our first- and second-round reviewers and to thank Crissy Statuto and Jennifer Cappello of Prentice Hall, who recruited the reviewers and managed the review process. Adhering to a tight time schedule, these reviewers scrutinized the text and the programs, providing countless suggestions for improving the accuracy and completeness of the presentation. It is a privilege to have the guidance of such talented and busy professionals.

***C# How to Program reviewers*:**
Hussein Abuthuraya (Microsoft)
Lars Bergstrom (Microsoft)
Indira Dhingra (Microsoft)
Eric Gunnerson (Microsoft)
Peter Hallam (Microsoft)

Habib Hegdarian (Microsoft)
Anson Horton (Microsoft)
Latha Lakshminaray (Microsoft)
Kerry Loynd (Microsoft)
Tom McDade (Microsoft)
Syed Mehdi (Microsoft)
Cosmin Radu (Microsoft)
Ratta Rakshminarayana (Microsoft)
Imtiaz Syed (Microsoft)
Ed Thornburg (Microsoft)
Richard Van Fossen (Microsoft)
Rishabh Agarwal (Delteq Systems Pte. Ltd.)
José Antonio González Seco (Sadiel S.A.)
Paul Bohman (WebAIM)
Alex Bondarev (SureFire Commerce, Inc.)
Ron Braithwaite (Nutriware)
Filip Bulovic (Objectronics PTY Ltd.)
Mark Burhop (University of Cincinnati)
Carl Burnham (Southpoint)
Matt Butler (Oakscape Inc.)
Andrew Chau (Rich Solutions, Inc.)
Dharmesh Chauhan (Microsoft Consultant, Singapore)
Shyam Chebrolu (SAIC Broadway & Seymour Group)
Kunal Cheda (DotNetExtreme.com)
Edmund Chou (MIT Student, www.devhood.com project, Microsoft Intern)
James Chegwidden (Tarrant County College)
Vijay Cinnakonda (University of Toledo)
Michael Colynuck (Sierra Systems)
Jay Cook (Canon Information Systems)
Jeff Cowan (Magenic Technologies)
Robert Dombroski (AccessOnTime)
Shaun Eagan ((Eagan Consulting)
Brian Erwin (Extreme Logic)
Hamilton Fong (Montag & Caldwell, Inc.)
Gnanavel Gnana Arun Ganesh (Arun Microsystems)
Sam Gentile (Consultant)
Sam Gill (San Francisco State University)
John Godel (TJX)
Dave Haglin (Minnesota State University in Mankato)
Jeff Isom (WebAIM)
Rex Jaeschke (Consultant)
Amit Kalani (MobiCast)
Priti Kalani (Consultant)
Bryan Keller (csharphelp.com)
Patrick Lam (EdgeNet Communications)
Yi-Fung Lin (MIT Student, www.devhood.com project, Microsoft Intern)

Maxim Loukianov (SoloMio Corporation)
Guarav Mantro (EDS PLM Solutions)
Jaimon Mathew (Osprey Software Technology)
Robert Meagher (Compuware NuMega Lab)
Arun Nair (iSpan Technologies)
Saurabh Nandu (Mastercsharp.com)
Simon North (Synopsys)
Jibin Pan (csharpcorner.com)
Graham Parker (VBUG)
Bryan Plaster (Valtech)
Chris Rausch (Sheridan Press)
Debbie Reid (Santa Fe Community College)
Bryn Rhodes (Softwise, Inc.)
Craig Schofding (C.A.S. Training)
Rahul Sharma (Maxutil Software)
Devan Shepherd (XMaLpha Technologies)
David Talbot (Reallinx, Inc.)
Satish Talim (Pune-Csharp)
Pavel Tsekov (Consultant)
John Varghese (UBS Warburg)
Peter Weng (MIT Student, www.devhood.com project, Microsoft Intern)
Jesse Wilkins (Metalinear Media)
Warren Wiltsie (Fairleigh Dickinson University/Seton Hall University)
Phil Wright (Crownwood Consulting Ltd.)
Norimasa Yoshida (MIT Graduate Student)

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.

Well, that's it for now. Welcome to the exciting world of C# programming. We hope you enjoy this look at leading-edge computer applications. Good luck!

Dr. Harvey M. Deitel
Paul J. Deitel
Tem R. Nieto
Cheryl H. Yaeger
Marina Zlatkina
Jeff Listfield

About the Authors

Dr. Harvey M. Deitel, CEO and Chairman of Deitel & Associates, Inc., has 40 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 NT. 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 is the author or co-author of several dozen books and multimedia packages and is writing many more. With translations published in Japanese, Russian, Spanish, Traditional Chinese, Simplified Chinese, Korean, French, Polish, Italian and Portuguese, Dr. Deitel's texts have earned international recognition. Dr. Deitel has delivered professional seminars to major corporations and to government organizations and various branches of the military.

Paul J. Deitel, Executive Vice President and Chief Technical Officer of Deitel & Associates, Inc., is a graduate of the Massachusetts Institute of Technology's Sloan School of Management, where he studied Information Technology. Through Deitel & Associates, Inc., he has delivered Java, C, C++, Internet and World Wide Web courses to industry clients including Compaq, Sun Microsystems, White Sands Missile Range, Rogue Wave Software, Boeing, Dell, Stratus, Fidelity, Cambridge Technology Partners, Open Environment Corporation, One Wave, Hyperion Software, Lucent Technologies, Adra Systems, Entergy, CableData Systems, NASA at the Kennedy Space Center, the National Severe Storm Laboratory, IBM 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, Inc., 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.

Tem R. Nieto, Director of Product Development of Deitel & Associates, Inc., is a graduate of the Massachusetts Institute of Technology, where he studied engineering and computing. Through Deitel & Associates, Inc., he has delivered courses for industry clients including Sun Microsystems, Compaq, EMC, Stratus, Fidelity, NASDAQ, Art Technology, Progress Software, Toys "R" Us, Operational Support Facility of the National Oceanographic and Atmospheric Administration, Jet Propulsion Laboratory, Nynex, Motorola, Federal Reserve Bank of Chicago, Banyan, Schlumberger, University of Notre Dame, NASA, various military installations and many others. He has co-authored numerous books and multimedia packages with the Deitels and has contributed to virtually every Deitel & Associates, Inc., publication.

Cheryl H. Yaeger, Director of Microsoft Software Publications with Deitel & Associates, Inc., graduated from Boston University in 3 years with a bachelor's degree in Computer Science. Other Deitel publications she has contributed to include *Perl How to Program, Wireless Internet & Mobile Business How to Program* and *Internet and World Wide Web How to Program, Second Edition*. Cheryl is increasingly interested in Microsoft's .NET strategy and in learning how Microsoft's .NET initiative will develop in the coming year.

Marina Zlatkina graduated from Brandeis University in three years with degrees in Computer Science and Mathematics and is pursuing a Master's degree in Computer Science at Brandeis. During her Brandeis career, she has conducted research in databases and has been a teaching assistant. She has also contributed to the Deitel & Associates, Inc. publication, *e-Business & e-Commerce for Managers*.

Jeff Listfield is a senior at Harvard College in Computer Science. His coursework includes classes in computer graphics, networks and computational theory and he has programming experience in C, C++, Java, Perl and Lisp. Jeff also contributed to the Deitel & Associates, Inc., publication *Perl How to Program*.

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 courses on Internet and World Wide Web/ programming, wireless Internet programming, object technology, and major programming languages and platforms, such as Visual Basic .NET, C#, Java, advanced Java, C, C++, 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 25-year publishing partnership with Prentice Hall, Deitel & Associates, Inc., publishes leading-edge programming textbooks, professional books, interactive CD-ROM-based multimedia *Cyber Classrooms*, *Complete Training Courses*, e-books, e-matter, Web-based training courses and course management systems e-content. 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 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 details.

The World Wide Web Consortium (W3C)

Deitel & Associates, Inc., is a member of the *World Wide Web Consortium (W3C)*. The W3C was founded in 1994 "to develop common protocols for the evolution of the World Wide Web." As a W3C member, Deitel & Associates, Inc., holds a seat on the W3C Advisory Committee (the company's representative is our Chief Technology Officer, Paul Deitel). Advisory Committee members help provide "strategic direction" to the W3C through meetings held around the world. Member organizations also help develop standards recommendations for Web technologies (such as XHTML, XML and many others) through participation in W3C activities and groups. Membership in the W3C is intended for companies and large organizations. To obtain information on becoming a member of the W3C visit **www.w3.org/Consortium/Prospectus/Joining**.