

Preface

This book offers instruction in Visual Basic .NET programming to college students who have completed a semester course or equivalent in the same topic. After having studied the book and completed the programming exercises, students should be able to create small to medium-sized business applications involving databases that run on desktops and the Web.

We believe effective programmers must combine theory with practice, so they can adapt to ever-changing computing environments. This book does not cover the breadth of topics found in some professional reference books, but it has a number of features that make it useful in the classroom:

- A step-by-step learning approach in which new ideas and concepts build on existing ones.
- Check-up exercises at the end of each section.
- Review questions and programming exercises at the end of each chapter.
- A student CD-ROM containing all sample programs and databases.
- A Web site actively supported by the authors.

Additional Materials for Professors

We think that one of the primary selling points of a textbook lies in the quality of support given by the authors to adopting professors. If you look at the textbook Web site currently maintained by Tony Gaddis (<http://gaddisbooks.com>) and Kip Irvine (<http://kipirvine.com/vbnet>), you will see that we have invested a great deal of time in customer support. The following materials are available at the book's Instructor Web site:

- A PowerPoint™ slide presentation for each chapter.
- A test bank for each chapter.
- Solutions to all programming exercises.
- Timely online support from the authors.

Skills We Emphasize

We believe students who plan to use Visual Basic .NET professionally should have at least the following basic areas of expertise:

- Object-oriented design and programming
- Desktop applications using relational databases
- Web applications using relational databases

In keeping with these basic goals, we have identified a number of learning objectives addressed by our book. The objectives are divided into broad areas: Object-oriented design, object-oriented programming, user interfaces, databases, and Web programming.

Object-Oriented Design

After reading this book, students should be able to do each of the following:

- Recognize and understand basic UML notation.
- Understand design differences between inheritance, composition, and interface implementation.
- Understand how polymorphism contributes to effective OO design.
- Design applications involving classes with composition and inheritance relationships.
- Create use-case scenarios to describe detailed execution steps.
- Design applications using multi-tier design models.

Object-Oriented Programming

After reading this book, students should be able to do each of the following:

- Understand basic reflection and run-time type identification.
- Understand early and late binding, as well as upward and downward type casting.
- Understand how delegates (function objects) are used.
- Implement exception handling with multiple catch blocks, throwing and re-throwing exceptions, and custom exception classes.
- Implement derived class constructors and constructors with optional parameters.
- Define shared methods, fields, and properties.
- Define structures, enums, classes, and nested classes.
- Overload and override class methods.
- Control access to members via Public, Private, Protected, and Friend modifiers.
- Define and implement interfaces in derived classes.
- Implement common .NET interfaces such as IComparable and IComparer.
- Use .NET collections in programs.

User Interfaces

After reading this book, students should be able to do each of the following:

- Be familiar with common user-interface design issues.
- Be able to use .NET error handling controls and event handling to trap errors at the user interface level.
- Customize advanced Visual Basic .NET controls such as the DataGrid, DataView, TreeView, Repeater, and DataList.
- Write program code that manipulates advanced Visual Basic .NET controls and responds to their events.

Databases

After reading this book, students should be able to do each of the following:

- Design relational databases containing multiple table relationships.
- Display data from related database tables.
- Update database tables using datasets.
- Implement the data tier in a multi-tier design using a database connection.
- Handle common database errors using exception handling.
- Create SQL action queries that update databases and alter the structure of databases.
- Create ADO.NET Command objects that execute database queries.
- Use DataReaders to read database data.
- Code applications that make extensive use of ADO.NET database objects.
- Create advanced SQL queries for viewing and updating databases.
- Create stored procedures and execute them from programs.
- Be familiar with database constraints and database security.

Web Programming

After reading this book, students should be able to do each of the following:

- Create ASP.NET applications using both flow layout and grid layout modes.
- Use the following basic HTML and ASP.NET controls in programs: Table, Label, TextBox, Button, LinkButton, HyperLink, and Calendar.
- Use all types of Web Forms validator controls.
- Use Web Forms list-type controls, including ListBox, DropDownList, RadioButtonList, and CheckBoxList. Create event handlers for the same controls.
- Upload files from Web browsers.
- Send mail with attachments from Web applications.
- Manage page-level state, session state, and application state in ASP.NET programs.
- Implement custom HTTP error handling at both the application level and IIS level.
- Create, delete, and modify browser cookies.
- Fill all list-type controls using a DataReader.
- Use data binding with Web Forms controls.
- Use template-based controls such as DataList and Repeater.
- Configure advanced DataGrid options and respond to DataGrid events.
- Create IIS application directories and deploy Web applications.
- Use ADO.Command objects in Web Forms to update databases.

- Be familiar with the protocols, languages, and processes employed by ASP.NET Web services.
- Create Web services that return value types and object types, including datasets.
- Create Visual Basic .NET applications to consume Web services.

Chapter Descriptions

Chapter 1: Classes. Object-oriented design; creating classes, properties, constructors, and destructors; multi-tier applications; composition relationships between classes, and nested classes.

Chapter 2: Exceptions and User Interfaces. User-interface design; input validation; ImageList, Toolbar, ListView and TreeView controls; structured exception handling.

Chapter 3 ADO.NET Databases: Introduction to ADO.NET; using data-bound controls; brief look at SQL; navigating, adding, and removing rows; filling list and combo boxes; selecting DataTable rows; parameterized queries.

Chapter 4: DataGrid, DataView, and ListView. DataGrid table styles and column styles; updating a datagrid; the DataView and unbound ListView controls; Three-tier Sports Rental Income program; Sports Rental Checkout program; Command objects; inserting, updating, and deleting table rows.

Chapter 5: Databases with Related Tables. Connecting to SQL Server databases; relational database design; SQL queries that join tables; database constraints; DataGrid control with related tables; related tables with unbound controls.

Chapter 6: Using SQL Server. SQL queries; creating databases and tables; using Server Explorer; using Enterprise manager; data definition language (DDL); data manipulation language (DML); executing queries in program code; creating and executing stored procedures.

Chapter 7: Web Forms (ASP.NET). ASP.NET basics; creating ASP.NET applications; Request and Response objects; ASP.NET objects and namespaces; standard Web Forms controls; CheckBoxList and RadioButtonList controls; formatting tips.

Chapter 8: Web Forms II. Custom error handling; Calendar control; data binding with arrays; uploading files; sending mail; data validation controls; page-level state; application state; session state; browser cookies; deploying a Web application.

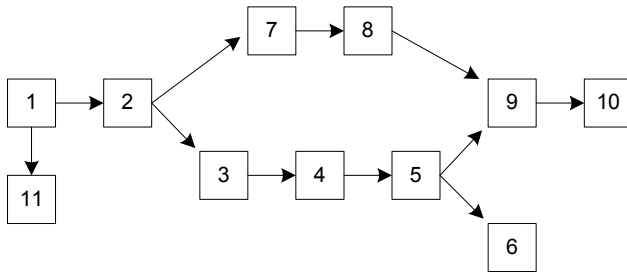
Chapter 9: ASP.NET Databases. Using a DataReader; CheckBoxList and RadioButtonList controls; Repeater control; DataList control; DataGrid control; adding Buttons to DataGrids.

Chapter 10: Web Services. Bookinfo Web service; creating and testing Web services; consuming Web services; connecting to a database; application state; UserList Web service.

Chapter 11: Advanced Classes. Enums, structures, objects and reflection; System.Object class; interfaces; IComparable and IComparer interfaces; inheritance; inheritance with constructors; overriding and overloading; abstract classes and methods; polymorphism; HashTable, SortedList; visual inheritance; delegates.

Sequencing the Book's Chapters

We recommend a sequential path through the book's chapters as the easiest approach for students. It is possible to delay covering Chapter 6 (*Using SQL Server*), or to omit it altogether. Chapter 6 emphasizes SQL action queries and stored procedures, which are recommended but not required when completing programming exercises in Chapter 9 (*ASP.NET Databases*). Chapter 11 (*Advanced Classes*) can immediately follow Chapter 1 (*Classes*). Chapters 7 and 8 (ASP.NET) can be covered immediately after Chapter 1 because they do not use databases. Chapters 9 and 10 must be covered after Chapter 5 (*Databases with Related Tables*). In summary, the following graph shows the chapter dependencies:



About the Authors

Kip Irvine has a M.S. degree in Computer Science and taught computer programming at Miami-Dade College for 17 years. He now teaches in the School of Computer Science at Florida International University. He has written four college textbooks: *COBOL for the IBM-PC*, *C++ and Object-Oriented Programming*, *Assembly Language for Intel-Based Computers*, and *Advanced Visual Basic 6*. He was a founding programmer at Omega Research. His books have been translated to Russian, Korean, Chinese, Polish, and French. He also has a doctorate in Music Composition.

Kaiyang Liang has a Doctorate in Mathematics from the University of Miami. He has been a professor at Miami-Dade College for twelve years, currently teaches in the Computer and Information Systems department. He is a Microsoft Certified Trainer (MCT) and has created many professional database applications. He was the co-author of the previous edition of this book, entitled *Advanced Visual Basic 6*.

Tony Gaddis teaches courses on computer programming languages, operating systems, and physics at Haywood Community College in North Carolina. He was selected as the North Carolina Community College Teacher of the Year in 1994, and received the Teaching Excellence award from the National Institute for Staff and Organizational Development in 1997. Tony has also provided training to companies and agencies, including NASA's Kennedy Space Center. He is the author of the *Starting Out with C++* series, and coauthor of *Starting Out with Visual Basic 6.0*, also published by Scott Jones.

Acknowledgements

We wish to thank the following persons for their contributions to this book:

- **Richard Jones**, editor and publisher, was the driving force behind this book. Thank you Richard, for never giving up on us.
- **Mario Rodriguez** was the editor who kept all production details on track. We had a lot of fun working together.

-

- **Eileen Troy and GEX Publishing Services** did a great job of keeping the production moving, with quality work all the way.

The following people were important reviewers and contributors:

- **Jeff Kent**, Los Angeles Valley College
- **Robert Phillips**, Compass Computing Group
- **Anita Philipp**, Oklahoma City Community College
- **Andre Poole**, Florida Community College at Jacksonville
- **Michael Olivero**, Visual Basic .NET consultant and Microsoft Student Ambassador at Florida International University, 2003.

The authors would like to thank the following professors who reviewed chapters during the book's development:

- Anthony Basilico, Community College of Rhode Island
- William Dorin, Indiana University Northwest
- Dana Johnson, North Dakota State University
- Astrid Lipp, Georgia State University
- Sally Field Mullan, College of DuPage
- Theresa Nagy, Northern Virginia Community College
- Andre Poole, Florida Community College at Jacksonville
- Ed Schott, Walsh University
- Craig Van Lengen, Northern Arizona University
- Sandy Wells, Gadsden State Community College **Recommended Reading**

We highly recommend the following books for further study:

- Dino Esposito. *Programming Microsoft ASP.NET*. Microsoft Press, 2003.
- Balena, Francisco. *Programming Microsoft Visual Basic .NET (Core Reference)*. Microsoft Press, 2003.
- Walther, Stephen. *ASP.NET Unleashed, 2nd Edition*. Sams, 2003.
- David Sceppa. *Microsoft ADO.NET (Core Reference)*. Microsoft Press, 2002.