Preface

Advanced Visual Basic® 2010, Fifth Edition, offers instruction in Visual Basic .NET programming to those who have completed a semester course or equivalent in the same topic. After studying the book and completing the programming exercises (called Programming Challenges), students should be able to create small- to medium-size Windows and Web applications that use databases. They will also gain essential concepts in object-oriented programming, event-driven programming, and test-driven development.

Effective programmers must combine theory with practice in order to adapt to changing computing environments. This book does not cover the breadth of topics found in some professional reference books, but it provides a practical approach to programming and problem solving. The following features make it helpful in the classroom:

- A step-by-step learning method in which new ideas and concepts build on existing ones
- Tutorials in which students gain hands-on experience by working with the chapter topics
- Review questions (called Checkpoints) at the end of each chapter section
- Tips that provide advice for solving programming problems, sprinkled throughout the chapters
- A list of key terms at the end of each chapter
- Review questions and exercises at the end of each chapter
- Programming projects at the end of each chapter that reinforce the chapter material
- A companion website that contains sample programs and other support materials

Changes in the Fifth Edition

This edition of Advanced Visual Basic 2010 offers many improvements. We place much more emphasis on object-oriented programming principles and software design than we did in the previous edition. The multi-tier application model (data, business objects, user interface) is a strong influence, as is programming with collections and components. Overall, the fifth edition provides an enhanced approach to designing, implementing, and testing well-constructed, maintainable, and extensible applications. Of particular note are the new sections on strongly typed collections, LINQ to Objects, LINQ to SQL, Unit Testing, Windows Presentation Foundation (WPF), and Windows Communication Foundation (WCF). The example programs were written in Visual Studio 2010.

Additional Materials

A primary selling point of a textbook lies in the quality of support given by the authors to adopting professors. The publisher makes excellent resources for this book available at www.pearsonhighered.com/irvine. The following materials are available to students and instructors:

- Online VideoNotes, narrated by Kip Irvine, which explain concepts and examples from the chapters
- A PowerPoint slide presentation for each chapter
- Example programs
- Online list of corrections to errors in the book

In addition, the following are available for professors:

- Answers to Review Questions and Exercises
- Solutions to Programming Challenges

Learning Objectives
Following are the learning objectives for this book, indicating the skills and knowledge that students may expect to attain:

- Master the use of .NET controls to create rich user interfaces
- Master the design and implementation of object-oriented multi-tier applications
- Master the design of manual and automated tests for desktop applications
- Master the use of .NET controls and exception handling to trap errors at the user interface level
- Master the displaying and updating of data in related database tables
- Master the creation of ASP.NET applications that contain multiple Web pages and databases
- Master the use of page-level state and session state in ASP.NET programs
- Be familiar with database constraints and database security
- Be familiar with creating and consuming Web services
- Be familiar with creating database-driven reports

Sequencing the Chapters
If your Advanced Visual basic course emphasizes Windows applications, we recommend that you complete Chapters 1 through 7 in sequence. For courses that cover Web programming, continue to Chapters 8 through 11. Chapter 12 (Reports, MDI, Interfaces, and Polymorphism) can be introduced any time after Chapter 4.

Chapter Descriptions
Chapter 1: Classes. Chapter 1 begins with basic concepts of classes and objects. Next, we show how to define classes, and enumerated types and structures, and how they are used in applications. We build a two-tier Bank Teller application that uses classes to simulate the basic operations of a software teller machine. The concept of multi-tier application design will continue to be a central theme throughout the book. We also introduce manual software testing as an important tool for validating program output.

Chapter 2: Input Validation and User Interfaces. Chapter 2 provides in-depth coverage of input validation and error handling. The ErrorProvider control is introduced as an ideal tool for input validation. Next, we review the ListBox, ComboBox, and CheckedListBox controls, and we show how to write code that deals with multiple selections. Following that, we show how to use the FlowLayoutPanel, WebBrowser, SplitContainer, and TabControl controls to create more creative user interfaces. The chapter finishes with the design and development of a software wizard and shows some of the power and flexibility of this type of application.

Chapter 3: Collections. Chapter 3 introduces some of the most useful and powerful collection classes in the .NET library, with the idea that collections of objects help to build concepts that can later be applied to databases. We show how to create and use ArrayLists, as well as strongly typed Lists and Dictionaries. The chapter finishes with examples that show how to search lists of objects, using Microsoft’s new Language Integrated Query (LINQ) technology.

Chapter 4: Using SQL Server Databases. Chapter 4 focuses on the basics of displaying and updating databases using .NET controls. It shows how Visual Studio enables data binding, which is the connecting of the user interface directly to database components. The chapter also stresses basic database concepts; how to execute SQL queries; how to sort and filter database data; how to display data in a grid; and how to bind individual controls to database
After reading this chapter, students should be able to display and update database tables rapidly with almost no programming.

**Chapter 5: Database Applications.** Chapter 5 focuses on database programming, using the ADO.NET library. Students can think of it as an extension of the database concepts and database binding from Chapter 4. In Chapter 5, students can integrate their knowledge of multi-tier application design with objects and databases. The chapter concludes with an extended example application that schedules appointments for a home repair services company.

**Chapter 6: Advanced Classes.** Chapter 6 introduces structures, which are simple containers for variables, properties, and methods. Then the chapter demonstrates the building of components, also known as class libraries. Then the chapter introduces unit testing, the industry standard for automated testing of individual units of code. This is followed by a brief introduction to defining and using custom event types in classes. The chapter ends with inheritance, a fairly large topic that is a core topic in object-oriented programming.

**Chapter 7: LINQ to SQL.** Chapter 7 introduces LINQ to SQL, a powerful tool for querying and updating database data. LINQ to SQL offers the opportunity to use object-oriented programming techniques to view and update databases. Essentially, students work with databases in the same way that they did with in-memory collections in Chapter 3. They learn how to create entity classes that model database tables. They learn how to create selection queries that join multiple entity classes, using common linking properties. Students learn how to insert, update, and delete table entries.

**Chapter 8: Creating Web Applications.** Chapter 8 introduces the ASP.NET runtime environment and shows how to use Visual Studio to create Web sites. Students learn what happens when an ASP.NET page is processed by a Web server. Students learn about runtime events and about the different categories of controls available in ASP.NET applications, and they learn differences between HTML controls and ASP.NET controls. The chapter describes application and configuration files required by ASP.NET applications. Finally, the chapter shows how to create a simple Web application containing various types of buttons, labels, headings, and text boxes.

**Chapter 9: Programming Web Forms.** Chapter 9 introduces students to programming techniques in ASP.NET applications. They also learn about XHTML, cascading style sheets, menus, and validation controls. Students learn how to upload files to a Web site and send email from a Web site. They learn how to save page state information and how to save information when users switch between pages. Finally, the chapter shows how application cookies are created.

**Chapter 10: Web Applications with Databases.** Chapter 10 introduces master-detail pages, which let students create a consistent look across a Web site. Following that, the chapter shows how to use data-bound controls with databases. It examines some advanced use of the GridView control. The chapter shows a few basic techniques available in JavaScript, and finishes with a brief introduction to the Microsoft Ajax extension controls.

**Chapter 11: Web Services and Windows Presentation Foundation.** Chapter 11 helps students understand the basic technologies behind Web services and the types of applications that use them. Students learn how to create and consume Web services. Next, the chapter introduces Microsoft’s exciting new Windows Presentation Foundation (WPF). WPF programs can be run from both the desktop and the Web. We show how to use ClickOnce technology, which greatly simplifies application deployment and installation.

**Chapter 12: Reports, MDI, Interfaces, and Polymorphism.** Chapter 12 introduces several important topics. First, its shows how to create reports for the desktop and Web,
using Microsoft Report templates and the ReportViewer control. Next, it shows how to create Multiple Document Interface (MDI) applications, which manage multiple client windows under a single parent window. Then we introduce advanced topics in object-oriented programming: interface types, abstract classes, and polymorphism. Although these topics are not heavily emphasized in Visual Basic applications, they can be important as programs grow in size and complexity.

**Appendix A: Answers to Checkpoints.** Students may test their progress by comparing their answers to the review questions at the end of each chapter section. These lists of review questions are called Checkpoints. Appendix A provides all theCheckpoint answers.

**Appendix B: Optional Reference Topics.** Appendix B contains a collection of optional reference topics. It shows how to calculate TimeSpan objects and how to format dates and times. It shows how to use the ListView control. Next is a guide to SQL Queries (SELECT, INSERT, DELETE, and UPDATE). Finally, it shows students how to write messages to the application log file. This can be a powerful tool for diagnostic and error reporting.

**Acknowledgments**

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

- Matt Goldstein, Acquisitions Editor at Addison-Wesley, who was the driving force behind this book
- Chelsea Bell, Editorial Assistant at Addison-Wesley
- Jogender Taneja, my project manager at Aptara, did a great job of keeping the production moving, with quality work all the way and Marianne L’Abbate, the excellent copy editor

We wish to thank the following individuals who reviewed the current edition:

Evans Adams, Fort Lewis College
Patricia McDermott-Wells, Florida International University
David S. McDonald, Georgia State University
Rudy Lee Martinez, Austin Community College

We also wish to thank the following professors who reviewed earlier editions of this book:

Jeffery Allen, Indiana University Purdue University Indianapolis
Chuck Bailey, Kenai Peninsula College
Anthony Basilico, Community College of Rhode Island
Joni Catanzaro, Louisiana State University
Ronald Del Porto, Penn State University, Behrend
William Dorin, Indiana University Northwest
Dana Johnson, North Dakota State University
Melody Kiang, California State University, Long Beach
Bruce LaPlante, University of Wisconsin, Green Bay
Astrid Lipp, Georgia State University
Thomas McCullough, Hillsborough Community College
David McDonald, Georgia State University
Sally Field Mullan, College of DuPage
Teresa Nagy, Northern Virginia Community College
Adam Peck, Ohlone College
Anita Philipp, Oklahoma City Community College
Andre Poole, Florida Community College at Jacksonville
Ed Schott, Walsh University
Craig Van Lengen, Northern Arizona University
About the Authors

Kip Irvine holds a Master of Science degree in Computer Science from University of Miami. He taught computer programming at Miami-Dade College for seventeen years, and he has taught at Florida International University since 2000. He has written programming textbooks for Addison-Wesley and Prentice-Hall, covering subjects such as Assembly Language, C++, Visual Basic, and COBOL. His books have been translated into Russian, Korean, Chinese, Polish, Spanish, and French. He briefly worked as a software developer in the industry.

Tony Gaddis taught 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 a best-selling author of numerous computer programming textbooks for Addison-Wesley, covering topics such as Alice, Java, C++, C#, Visual Basic, and algorithms.