

PREFACE

The *Visual Basic .NET Coach* builds on an approach that I introduced in *The C++ Coach* and *The Visual Basic Coach*, and applies it to the exciting new world of Visual Basic .NET. The book's premise is the same one that successful coaches have long used: The more you practice a skill, the better your mastery of that skill becomes.

Given the huge installed base of Visual Basic 6.0 developers, Visual Basic .NET is sure to have a large following. This new version of Visual Basic .NET is much more than a minor update; it is a huge departure from the development environment of version 6.0. Object-oriented development inundates every aspect of development with Visual Basic .NET. More than ever, a textbook is needed that targets the student who is learning programming for the first time or wants to learn the Visual Basic .NET language in a friendlier manner.

Target Audience

Although I was trained as a computer science major, I have done most of my consulting in the Information Systems field. While some computer scientists are striving to become hard-core applications developers who will write the latest operating system from Microsoft, many more programmers are studying to become the information systems professionals of tomorrow. A computer science professional most likely will go on to developing operating systems and application programs like Microsoft Word or Excel.

However, information systems professionals may move on to develop database applications or become network administrators or even database administrators.

The “Coach” series of textbooks are designed specifically for Information Systems professionals. Their needs are unique in that they require non-mathematical, non-scientific examples that relate to topics they already know. Therefore, *The Visual Basic .NET Coach* is comprised of business, humorous, and sports examples so that these professionals can easily relate to the problems they are trying to solve.

Whether you intend to program for the rest of your life, are required to take a course in school, or wish to merely gain an appreciation for programming, this book can help you. It is specifically designed to be a standalone volume for students who are not majoring in computer science and learning Visual Basic .NET. It also may function as a bridge between your current level of understanding and more advanced Visual Basic .NET texts.

Author’s Approach

Because the audience for *The Visual Basic .NET Coach* is different from that of most programming texts, I have taken a unique approach by applying many techniques learned from coaching athletes to the teaching of computer programming. I have competed as an international athlete representing the United States and currently am coaching Olympic hopefuls in the sport of race walking. In many ways, the teaching of race walking is similar to that of computer programming.

Race walking is a complex sport that requires combining Olympian endurance with a very complicated technique. In order to master it, one must expend a great deal of practice and attention to the details of proper technique. Coaches do not start athletes by sending them into competition on the first day of practice. Instead, they usually require them to repeat drills over and over again until they have mastered the techniques of the sport.

Similarly, instead of beginning with problem solving, this text teaches sound Visual Basic .NET syntactical fundamentals first. While learning the basic building blocks of Visual Basic .NET, you will perform programming drills repeatedly until you understand the fundamentals and subtleties of Visual Basic .NET. These skills will be reinforced with clearly defined problems and solutions that focus on problem solving. First a problem is defined, then the issues of solving the problem are discussed, and finally the solution is provided.

The many object-oriented aspects of Visual Basic .NET are introduced in a carefully built-up presentation. You will first study simple objects like a form, button, or text box, and then become comfortable with their operation before defining your own. As your programming skills increase, you will grapple with more difficult topics, such as advanced object-oriented techniques, thereby allowing you to master the earlier concepts before trying to tackle the idiosyncrasies of inheritance and overloading methods.

This book is not intended to be a complete reference for Visual Basic .NET. The language and object model are far too large to do everything justice. Instead, throughout the text we will present the most important features of Visual Basic .NET, explain their syntax, and present drills that explore the subtleties of the syntax. We supplement these drills with real-world examples of programming problems and build upon the knowledge gained from these drills. Because there are so many options in Visual Basic, topics deemed optional are added at the end of each chapter in a Coach’s Corner so that more advanced students can learn them, while students looking for a basic understanding of the language can skip over them. This approach contrasts with that of many other texts, which try to be complete reference manuals instead of instructional textbooks. They introduce too many constructs of Visual Basic .NET before solidifying a mastery of the most required Visual Basic .NET syntax.

While writing the proper programs is important, so is motivating the group and people around you to get the job done. Whether it is breaking up the monotony of a long

race walk or the dryness of a computer text with some light humor, the idea is the same. *The Visual Basic .NET Coach* finds different ways to motivate and amuse you as you read through the text.

Scope of Coverage

The Visual Basic .NET Coach covers Visual Basic .NET as if you have never learned a programming language before. Starting with a brief introduction to computer languages and where Visual Basic .NET fits in, I get students rolling immediately with an example that shows why Visual Basic .NET has the word Visual in its title. Students are introduced to the development environment and how to create an application with images and text.

Once a basic understanding of the environment is mastered, a discussion of fundamental programming constructs—variables, operators, conditional statements, functions, objects, and loops—follows. These constructs are used to develop programs that solve relatively simple, but useful problems. Interwoven between the presentation of these constructs is the introduction of many of the basic controls used regularly by Visual Basic .NET developers. Instead of just listing these controls, examples are given that motivate the benefit of one control over the other, based on the goal of the application.

Once students have gained a firm understanding of the basics, we concentrate on additional features of Visual Basic .NET such as arrays, structures, files, and the advanced concepts of object-oriented development like inheritance and polymorphism. While many introductory classes in Visual Basic .NET do not cover extensive advanced topics, we have made it easy to select the ones that you wish to add to your course. Databases are covered in a single chapter explaining how to access data stored in a database using the built-in wizards and given an introduction to SQL. Also included are sections on graphic routines, menus, and multiform applications.

Pedagogy Drills

We all know that you cannot learn to program by merely reading a textbook in a narrative format. *The Visual Basic .NET Coach* provides about 100 drills that provide students with immediate feedback on their understanding of what they just learned. The drills are presented as questions, and their complete solutions with detailed explanations are included at the end of each chapter.

The following is a sample of a few drills from Chapter 4 on conditional statements. It shows how students can get immediate feedback with slight variations of the same problem, so they can master all aspects of the concept they are learning.

DRILL 4.8

Assume that the code for the previous example was instead coded as follows:

```
Private Sub btnIfElse_Click(...  
    Dim sngPurchasePrice As Single  
  
    sngPurchasePrice = Val(txtInput.Text)  
  
    If (sngPurchasePrice > 100) Then  
        lblOutput.Text = (sngPurchasePrice * 0.05).ToString()  
    End If  
End Sub
```

(continues)

DRILL 4.8 (continued)

```

ElseIf (sngPurchasePrice > 500) Then
    lblOutput.Text = (sngPurchasePrice * 0.1).ToString()
Else
    lblOutput.Text = "NO DISCOUNT"
End If
End Sub

```

What do you think would be contained in `lblOutput`:

- 1 If the user enters 600.00 in the `txtInput` text box?
- 2 If the user enters 250.00 in the `txtInput` text box?
- 3 If the user enters 50.00 in the `txtInput` text box?

The answers are available at the end of each chapter:

Drill 4.8**With 600.00 Entered by the User**

When the program is executed and the button is clicked, the `If` statement is evaluated. If the user enters 600.00, then the `If` statement evaluates the expression by comparing the value entered to 100. Therefore, the conditional expression will evaluate to `True`. This causes all the statements until the `ElseIf` to be executed. Therefore, the purchase price is multiplied by .05 and the result is converted to a `String`. So the value "30" is placed in the label `lblOutput`. Since the conditional expression in the `If` statement evaluated to `True`, none of the statements after the `ElseIf` or `Else` statements and before the `End If` statement are executed. Note this is not what you wanted to have happen! Because you evaluated the condition comparing the value to 100 first, it will evaluate to `True` when you really want the expression `> 500` to evaluate to `True`. The order you evaluate your conditions can make a difference.

With 250.00 Entered by the User

When the program is executed and the button is clicked, the `If` statement is evaluated. If the user enters 250.00, then the `If` statement evaluates the expression by comparing the value entered to 100. Therefore, the conditional expression will evaluate to `True`. This causes all the statements until the `ElseIf` to be executed. Therefore, the purchase price is multiplied by .05 and the result is converted to a `String`. So the value "12.5" is placed in the label `lblOutput`. Since the conditional expression in the `If` statement evaluated to `True`, none of the statements after the `ElseIf` or `Else` statements and before the `End If` statement are executed.

With 50.00 Entered by the User

Finally, if the user enters a value less than or equal to 100, neither the `If` or `ElseIf` conditional expressions evaluate to `True`. Therefore, the statements after the `Else` statement and before the `End If` statement are executed.

So the text "NO DISCOUNT" is placed in the label `lblOutput`.

Tips

Throughout the chapters, tips highlight key programming issues.

COACH'S TIP

A Long variable can hold any value that an `Integer` variable can, but not vice versa.

Warnings

Commonly made mistakes are highlighted with a warning box.

COACH'S WARNING

When you list a range of numbers for a Case statement, you must list the smaller number first.

End-of-chapter Material

Key Terms

At the end of each chapter all new terms introduced within the chapter are listed with a condensed definition of the term.

Case Studies

A case study in each chapter focuses on the skills developed in the chapter in a practical real-world example. Throughout most of the text, our case study focuses on a business owner creating a system that will process payroll for his company. It is simple to follow and clear. We provide the problem statement, discuss its solution, and then present the coded solution. Such immediate reinforcement of the skills learned in the chapter greatly improves retention.

Coach's Corner

At the end of each chapter, optional topics are included to round out the student's knowledge of the topics introduced in each chapter. These topics may be important to the mastery of Visual Basic .NET, but may not be timely in an academic discussion of a topic.

Additional Exercises

Finally, each chapter is followed with a series of short-answer-style questions and programming assignments for students to practice what they have learned in the chapter.

Different Course Options

This book is designed for a one- or two-semester introductory programming sequence for non-computer science majors. In general, each chapter builds on your programming arsenal and is ideally designed to follow sequentially. However, the text was designed with the idea that many topics are optional. Many of these optional topics are presented at the end of each chapter in the Coach's Corners. These topics can be added or removed without adding dependency issues.

The biggest issue with determining the order in which to teach Visual Basic .NET is when to teach objects. The chapters have been carefully designed so that objects can be moved later or the advanced object topics earlier.

- ◆ Chapters 1–4 are fairly mandatory for any programming sequence. You then have the option of continuing in order or moving chapters around.
- ◆ Chapter 5 on subroutines and functions can be moved after objects are introduced in Chapter 6.
- ◆ Chapter 6 on the basic declaration of objects can be moved before Chapter 5 or after Chapter 7.
- ◆ Chapter 7 on repetition can be moved before Chapter 6 and, with a little tweaking, before Chapter 5 as well.

x PREFACE

- ◆ Chapter 8 on arrays, collections, and structures can be moved as late as desired.
- ◆ Chapter 9 on files can also be moved as late as desired.
- ◆ Chapter 10 on advanced object-oriented techniques can be moved up right after the basic objects chapter.
- ◆ Chapter 11 on database concepts can be taught as soon as Chapters 5, 6, and 7 have been covered.
- ◆ Chapter 12 on advanced Visual Basic .NET topics can be taught as desired. Many of these topics may or may not be covered and can be added to enhance other topics.

Many of the additional chapters can be skipped, based on the preferences of the instructor.

To the Student

So you're taking a computer course. Odds are you'll either love it or hate it. It's the job of this text to help you through it. Regardless of why you are taking the course, you need to learn the material presented by your teacher. If you follow my advice, you will make the process a lot easier. This book is designed to be interactive. **DO NOT** just read the drills. Try them. You will learn much more that way. While some of the examples are dry and simply there to help with your mastery of the Visual Basic .NET syntax, many more are colorful statements from things I enjoy. The sports theme shows up in many examples, but I have also included many names and quotes from movies. Try to see if you can figure them out. (Hopefully, I am not dating myself too badly.)

Instructor Supplements

The following supplements are available on-line for qualified instructors only. Please contact your Addison-Wesley representative for information.

- ◆ **Instructor's Manual**—includes complete answers to the exercises.
- ◆ **Test Bank** in powerful test generator software—includes a wealth of free response, multiple choice, and true/false type questions.
- ◆ **Lab Manual**—more than 20 labs, each with a complete step-by-step solution including many screen captures, to accompany the topic progression in the text.
- ◆ **CourseCompass**—A dynamic, interactive online course management tool powered by Blackboard, CourseCompass provides flexible tools and rich content resources that enable instructors to easily and effectively customize online course materials to suit their needs. Now instructors can track and analyze student performance on an array of Internet activities. Please contact your Addison-Wesley representative for more details.

Student Supplements

Please visit www.aw.com/cssupport for the following resources.

- ◆ **Source Code** to all programs.
- ◆ **PowerPoint** slides which include programs and art from the text.

Acknowledgments

There are many more people's efforts that go into the creation of a textbook other than the author's. Obvious thanks go to my editors Susan Hartman Sullivan and Mary Clare

McEwing, without whose efforts *The Visual Basic .NET Coach* would not have reached publication; Patty Mahtani, Emily Genaway, Daniel Rausch, and Adriana Lavergne for their assistance in the production cycle of the text; and Michael Hirsch for making sure the world knows about this text. Additional thanks go to Tim Burke, who helped me proof the text, as well as acting as a sounding board for many examples and to Steve Davis, my technical reviewer.

Along the way many people helped with the many stages my manuscript went through. Additional thanks go to my reviewers:

Tammy L. Ashley, *New Hampshire Community Technical College at Manchester*

Paula Baxter, *San Juan College*

Chris Beaumont, *Queens College*

Dana Johnson, *North Dakota State University*

Bruce W. Mielke, *University of Wisconsin, Green Bay*

Larry Press, *California State University, Dominguez Hills*

Steve Robischon

Alfred J. Seita, *Kapiolani Community College*

Richard Southern, *El Paso Community College*

Preston Vander Weyst, *Edmonds Community College*

Edward J. Williams, *University of Michigan*

Jeff Yeley, *Houston Community College*

David Zolzer, *Northwestern State University*

Special thanks go to Jim McKeown, Dakota State University, who helped improve the object chapters.

Special thanks also go to Dr. Nira Herrmann and Pat Henry, who continue to make teaching at Drexel a pleasure.