Instructor's Summary

This summary provides information about the course that you can teach with *Murach's Beginning Java 2, JDK 5*, about the book itself, about the student materials that can be downloaded from our web site, about the Instructor's CD, and about upgrading from the previous edition of *Murach's Beginning Java 2*.

We think that this summary provides some useful information, so we hope that you'll take a few minutes to read it. If you prefer, though, you can check the contents that follow and read just the topics that are of interest to you.

Mike Murach & Associates, Inc.
murachbooks@murach.com • www.murach.com

About the course

The topics that follow give you information about the course components and the way you can use these components in an effective course.

What the course components are

The primary component of this course is *Murach's Beginning Java 2, JDK 5*. This is the second edition of the book that has been used by more than 100 universities, colleges, and community colleges for courses ranging from first-programming courses to upper level programming courses. To make it easy for you to use this book for your course, we also provide these components:

- A downloadable Student Workbook that includes behavioral objectives, chapter summaries, terms lists, self-study tests, the exercises from the book, and student projects that force the students to apply what they've learned by developing complete applications.
- Downloadable Java classes and data for the book applications, the exercises in the book and Student Workbook, and the student projects in the Student Workbook.
- An Instructor's CD that contains everything else you need for running a course, including PowerPoint slides, question banks for tests, solutions to the exercises, and solutions to the projects.

The students can download the first two components from our web site, but they are also included on the Instructor's CD so you don't have to download them. As you read through the rest of this document, you'll learn more about these components.

What types of courses the book can be used for

The first edition of this book was used for first programming courses, first Java courses, and advanced Java courses. It was used in high school, community college, college, and university courses. And it was used by several large corporations for inhouse training.

One of the features that makes our book unique is its continual focus on practical development skills. For instance, ours is one of the few Java books that presents data validation techniques for both console and GUI applications, even though they're required for every real-world application. Ours is one of the few books that presents BigDecimal arithmetic, even though you need to know how to use it for common business calculations. And ours is one of the few books that does an adequate job of showing how to populate business objects with data from files or a database, even though that's essential no matter what type of application you're developing.

For that reason, our book is especially well suited for technical schools and for Information Science, CIS, and IT departments. But the first edition of the book was also used by Computer Science departments for their Java courses. Either way you look at it, the new book teaches all of the essential subjects for a solid Java course, and it does that in a way that will help the students learn faster and better than ever.

What Java versions our book supports

Since Java 1.0 was first released in 1996, five other versions have been released: from 1.1 through 1.5. With version 1.2, Java became known as Java 2, and that name is still in use today because it is still the Java 2 Platform that you're working on, even though the version numbers have gone beyond that.

As our book went to press, the current release of Java was version 1.5.0, which is also referred to as Java 5.0. That's the version that we used to develop all of the applications, exercises, and projects for this book.

Because all Java versions are upwards-compatible, everything in the previous versions will work with the new versions. In general, a new version provides new classes and methods. Some of these provide new capabilities; some improve upon the old ones. When a new class improves upon an old one, the old one is marked as "deprecated." That means it will still work, but you shouldn't use it any more.

Because our book is based on Java 5.0, your students need to use that version or a higher one with our book. Otherwise, they won't be able to use the new features of Java 5.0. In chapter 1, your students can learn how to download the latest version of Java from the Sun web site.

Since Java is platform-independent, a Java application can be run on any platform that supports Java, like any computer that runs Windows, Solaris, or Linux as the operating system. However, since the platform for most computer users today is Windows, this book uses Windows to illustrate any platform-dependent procedures. If your students are using some other platform, they may need to download information from the Sun web site to learn how to do some of those procedures on their systems.

What IDEs our book supports

In chapter 1, the book shows how to download and use an inexpensive product called TextPad. It can be used to enter, edit, compile, and run Java classes, but doesn't provide any debugging tools. We think this is an excellent product for students who are learning Java, and most instructors who adopted the first edition of our Java book agreed.

Chapter 1 also introduces Java IDEs, but the book doesn't show how to use any of them. However, we do offer a downloadable PDF file that shows how to install and use a free Java IDE called BlueJ (see http://www.murach.com/bluej). This is a relatively simple IDE that offers some class diagramming and debugging features. That's why some instructors prefer to have their students use this IDE instead of TextPad.

Of course, you can also use any other Java IDE with our book. The trouble is that the more time the students take to learn how to use the IDE, the less time they have for learning Java. So we think TextPad and BlueJ are two of the best alternatives for most Java courses.

About the book

When you review *Murach's Beginning Java 2, JDK 5*, it's pretty obvious what we tried to do. That is, we tried to present a complete set of real-world Java skills with as much structure, logic, and clarity as possible. The topics that follow, though, present some of the thinking behind the book that may not be that obvious.

What each of the 5 sections is designed to do

Unlike many of the competing books, our book is divided into five sections, which give some much-needed structure to a comprehensive language like Java. The paragraphs that follow describe the contents and logic of each of these sections.

- In *section 1*, your students will quickly master the basics of the Java language. That includes how to install Java; how to develop a console application using the Scanner class for user input; how to work with data and arithmetic; how to use the selection and iteration structures; how to use some of the Java classes and methods; and how to write and use static methods. By the time your students complete this five-chapter section, they will know how to write bulletproof applications with simple exception handling and complete data validation. And that will give them a realistic view of how an application should work.
- With section 1 as background, the students are ready to learn how to create and use their own classes, and that's what they learn in *section 2*. To start, chapter 6 shows them how to create and use their own business and data access classes, which is the essence of object-oriented programming. Then, in chapters 7 and 8, they learn how to use inheritance and interfaces, not only as they develop their own classes, but also as they use Java classes. By the time your students finish this section, they'll have all the skills they'll need for developing three-tiered applications and all the skills they'll need for getting the most from the Java API.
- In *section 3*, your students will learn more of the core Java features. That includes how to work with arrays, collections, dates, strings, exceptions, and threads. Along the way, they'll learn new 5.0 features like enhanced for loops, typed collections, generics, autoboxing, assertions, and the StringBuilder class. However, due to the modular design of these chapters, you don't have to teach all of these chapters before you go on to the next sections. In fact, as you will see in a moment, you can skip to section 4 or 5 once you've complete the chapters on arrays and collections.
- In *section 4*, your students will learn how to develop graphical user interfaces (GUIs). First, they'll learn how to use Swing components to develop GUI applications that handle events, validate data, and populate objects. Then, they'll learn how to develop applets for the same purposes. We think this is the proper place and treatment for applets now that servlets have become the standard for web development.
- In *section 5*, your students will learn how to populate business objects with data from text files, binary files, XML files, or a database. This shows them how to implement the data access classes that they learned about in section 2, which up to this point they've used as encapsulated methods. Remember, though, that you can present this section or any of its chapters much earlier in the course if you want to.

7 ways our book improves Java instruction

Because Java has always been a difficult subject to teach, we've tried to improve upon the features of the first edition so the new book works better than ever. Here, then, are some of the features of the new edition that we think will help your students learn faster and better than ever before.

- Since the new Scanner class makes it easier to develop console applications, we've used it from the start. In fact, your students will learn how to develop a simple application that gets user input, performs a calculation on it, and displays the results in chapter 2. That gets your students off to a fast start whether this is their first programming course or whether they already have experience with another language.
- Like the first edition, the new edition doesn't try to show the students how to develop their own classes until they know the Java essentials, which include how to use Java classes and objects. Instead, the Java essentials are presented in section 1, and object-oriented programming is presented in section 2. In other words, we don't try to present OO theory until the students know enough to apply it. In our experience, any other approach is both confusing and counterproductive.
- If you look at the table of contents for our book, you'll see that the first two chapters introduce the student to Java programming. After that, each chapter is built around a related set of Java skills or functions. That highly-structured approach helps the student learn more thoroughly because (1) each skill set is presented completely in a single chapter, not spread over several chapters, and (2) Java isn't seen as an overwhelming collection of classes and methods. This also makes it easier for the students to look up the coding details that they can't remember later on.
- In all of the Java chapters in our book, you'll see summaries of the classes, constructors, and methods that are most useful for the tasks at hand. This helps the students learn more efficiently. In contrast, most competing books make little effort to summarize the classes and methods that are used the most. As a result, the students are forced to dig through the text and examples to learn about the classes and methods that they need.
- Unlike competing books, our book focuses on the skills for developing object-oriented Java applications in the real world. For instance, as I've already mentioned, ours is one of the few books that presents data validation, the use of the BigDecimal class, and the right way to populate business objects via data access classes. Beyond that, though, we try to maintain this real-world focus in every chapter of the book.
- Our "paired pages" method of presentation helps the students learn faster by reading less. Even more important, though, this method makes it easier for students to refresh their memories when they need to look up information because it's all in the figures. If, for example, the students need to refresh their memories about how to use the LinkedList class, they'll find a summary of its constructors and methods in figure 11-9, examples of its use in figures 11-10 and 11-11, and a complete application that uses a linked list in figures 11-12 and 11-13. Is it any wonder why instructors tell us that their students "love" our paired pages?
- Unlike other books, the exercises at the ends of our chapters never present new information because we think that's a poor way to present information. Instead, our exercises guide the students through the application of what they've already learned and challenge the students to apply what they've learned in new ways. Also, because

we provide the starting code and data for most of the exercises, your students get a maximum amount of practice in a minimum amount of time.

Frankly, there are many other reasons why our book will help your students learn faster and better than they will with competing books, but these are seven that differentiate our book from the others. The result is that your students will learn more from our book, ask better questions when they come to class, and need less help in lab. And that in turn will free you to do a better job of teaching.

How the modular design of our book helps you customize your course

Sections 1 and 2 of our book present a complete subset of Java that includes all the skills your students need for designing, coding, and testing object-oriented, console applications. When your students finish the nine chapters in these sections, they will have a clear view of what Java programming is and what they need to learn to become proficient at it.

Then, as much as possible, we implemented the chapters in section 3 as independent instructional modules. And we implemented all of section 4 and each chapter in section 5 as independent modules. We refer to this type of book organization as "modular organization," and this is summarized in the table that follows.

Section	Chapters	Title	Prerequisites	Teaching order
1	1-5	Essential Java skills	None	Sequential (the chapters must be read in sequence)
2	6-9	Object-oriented programming	Section 1	Sequential, but you can skip to section 3 after chapter 6
3	10-14	More Java essentials	Section 1, plus chapter 6	Best read sequentially, but you can skip to section 4 or 5 after chapters 10 & 11
4	15-18	GUI programming	Sections 1 & 2, plus chapters 10 & 11	Best read sequentially, but you can skip to chapter 18 after chapter 15
5	19-21	Data access programming	Sections 1 & 2, plus chapters 10 & 11	Random (whichever sequence you prefer)

This table shows that the prerequisites for sections 4 and 5 are sections 1 and 2 plus chapters 10 and 11. As a result, you can skip to either section 4 or 5 after you teach the first 11 chapters of the book. If, for example, you want to show your students how to implement the data access classes of an application right after you complete chapter 11, you can do that by skipping to section 5. Or, if you want your students to learn GUI programming next, you can skip to section 4 after chapter 11.

This table also shows that you don't have to teach all of the chapters in section 2 before you go on to section 3. In fact, you only need to teach chapter 6, which shows your students how to create and use their own business classes and objects. This lets you teach some of the skills in section 3, like how to work with arrays, collections, dates, and

strings, before returning to section 2 and teaching how to work with inheritance, polymorphism, and interfaces. This approach is often useful when you're teaching students with little or no programming experience.

In practice, you may decide that you want to teach the chapters in numeric order, and that works too. However, we encourage you to adjust the sequence of your course to the interests of the students and the requirements of the course. That's why we built as much modularity as possible into the book.

How the top-down design of our chapters helps you customize your course

As part of our instructional theory, we have also implemented each chapter from the top down. This means that the most important skills are presented first and the least important skills are presented last. That in turn means that you don't have to teach every chapter in its entirety. Instead, your assignments can stop at any of the logical stopping places within a chapter.

If, for example, you look at the contents for chapter 3 of the book, you'll see four major topics: (1) the basic skills for working with data, (2) the use of Java classes for working with data types, (3) an Invoice application that illustrates the use of these skills, and (4) the use of the BigDecimal class for working with decimal data. Then, if you're teaching a first-programming class and decide that it's too early to present the BigDecimal class, you can skip the last topic and come back to it later. But if you're teaching a class for students who already have programming experience, BigDecimal is a topic that you'll want to include.

The same logic applies to many of the other chapters in the book, especially the more advanced chapters. When you assign chapter 11, for example, you may want to leave out the topic on legacy collections. When you assign chapter 13, you may want to leave out the topic on assertions. And when you assign chapter 21, you may want to leave out the topic on metadata. This lets you adjust the complexity of your course to the aptitude and progress of your students.

Why our book works so well for students who are taking Java as a first language

In chapter 1, the students learn how to install Java and compile and test Java classes. Then, in chapter 2, the students learn how to develop a complete Java application that gets data from the user via the Scanner class, performs some calculations on that data, displays the results of those calculations at the console, and repeats this process until the user ends the application. Along the way, the student learns how to use the double and integer data types, simple arithmetic statements, simple if-else statements, a simple while statement, and a few Java classes and methods.

We call that the "Aha!" experience because it gives the students a useful perspective on what programming is. From that point on, the students learn more rapidly because they have that perspective. In chapter 3, they learn all the skills they need for working with data, which they've already been introduced to. In chapter 4, they learn how to use all of the control structures, which they've already been introduced to. And in chapter 5, they learn how to make bulletproof applications by adding exception handling and data validation.

In contrast, many competing books use what we call the "small step" approach. With that approach, the first application is often as simple as the "Hello World!" application.

Then, each subsequent application or chapter adds a small piece of the puzzle at a time: data types, arithmetic, selection structures, iteration structures, and so on. The problem with this approach is that the students don't get the perspective they need for rapid progress until several chapters into the book. Until they do, progress is painstakingly slow for the high-aptitude students and puzzling for the low-aptitude students.

Why our book works so well for students who are taking Java as a second language

The approach that I've just described works even better for students who are learning Java after they've learned a language like C++, Visual Basic, or C#. These students can move rapidly through the first five chapters because they're just learning how to do the skills that they already know with a new language. Once they've done that, your course can focus on the more interesting aspects of Java: object-oriented programming, GUI programming, data access programming, and more. In short, you'll be on your way to a great course because you won't waste time on the basics.

About the student materials

To help your students get the most from our book, we provide two components that can be downloaded from our web site and one CD that contains the software required by the course.

The downloadable Student Workbook

At the end of each chapter in the text, you'll find a chapter summary and the exercises for the chapter. For those who find that too limited, though, we provide a Student Workbook in PDF format that can be downloaded from our web site.

This Workbook contains study aids like behavioral objectives, chapter summaries, terms lists, and self-study questions. It also includes the exercises from the book. And it includes student projects that require the students to develop complete Java applications from scratch. In contrast to the exercises, the student projects provide the specifications for the application without any guidance. That way, they are the surest test of whether a student has mastered the skills that are presented in the book.

The download for the Student Workbook also includes the starting classes and data that are required for the student projects. That way, the students have everything they need for doing the projects.

The downloadable book applications, exercise starts, and data

The other student download provides (1) the Java source code, class files, and data for the applications that are presented in the book; and (2) the starting source code and data for the exercises that are presented in the book and in the Student Workbook. This lets the student compile and run the applications that are presented in the book. It also prepares the student for doing the exercises.

To install and use these files, the student can refer to appendix A in the book. In general, though, this download consists of one executable file that creates the required folders and stores the right files in each folder.

The CD that contains Java and TextPad

In the first edition of our book, we included all of the required software on a bound-in CD. That way, the students were able to install Java 1.4, the Java documentation, and TextPad from the CD and didn't have to take the time to download it from the Sun and TextPad web sites. This also insured that everyone had the same releases of the software.

With the JDK 5 edition of the book, though, we didn't include a CD. However, for those students who don't want to have to download Java, TextPad, and the other required files, we provide a CD that includes the executables for installing the required software as well as all the applications, source files, and data that the students require. That way, they don't have to download anything.

This CD is available two ways. First, we package the CD with the book:

ISBN 1-890774-34-0 Book plus software CD $56.50

In this case, we've charged a nominal fee of $7.00 for creating the CDs and packaging them with our books. Second, we offer the CD with the printed version of the Student Workbook that's described next.

The printed Student Workbook and bound-in CD

To make it easier for the student to use the Student Workbook and student CD, we also offer a printed version of the Student Workbook that has the CD bound in. Here again, we've tried to offer this component at a nominal price:

ISBN 1-890774-33-2 Workbook with bound-in CD $20.00

Since this saves the students the cost of printing the 200-page Workbook on their own PCs, we think it's worth buying the print edition. This also encourages the students to do their studying in print, which we think is more efficient than studying on screen.

What the Instructor's CD contains

Simply stated, we have tried to include everything you need for running an effective Java course on the Instructor's CD for *Murach's Beginning Java 2, JDK 5*. A complete summary follows.

Book applications, exercise starts, project starts, and data

To make it easy for you to review or demonstrate the applications that are presented in the book, they are included on the Instructor's CD along with the required data files. Similarly, the source files and data that are needed for the exercises and projects are also included on the Instructor's CD. Although these files are also available as student downloads, this lets you access all of the instructional materials from a single CD.

The Student Workbook

We've also put the downloadable Student Workbook on the Instructor's CD. That way, you don't have to download anything.

Behavioral objectives

In case you want to modify the behavioral objectives that are in the Student Workbook, the Instructor's CD consolidates them all in a single Word file. That way, you can add, delete, or modify our objectives to suit your purposes. You can then distribute the modified objectives to your students.

We prepared these objectives based on the principles presented by Robert F. Mager in his classic book, *Preparing Instructional Objectives.* That means that our objectives clearly describe what the students should be able to *do* when they complete each chapter. That also means that you should be able to test your students to make sure that they are able to meet the objectives.

If you study our objectives, you can see that the first objectives for each chapter are what we refer to as *applied objectives*. These ask the students to apply what they've learned as they develop Java applications. These are the critical objectives of a programming course, and we've done our best to provide questions in the question banks that test these objectives. Nevertheless, these objectives are best tested by the exercises and student projects.

After the applied objectives for each chapter, you'll find what we refer to as *knowledge objectives*. These objectives define skills like identifying, describing, and explaining the required concepts, terms, and procedures. In general, all of the students should be able to do the knowledge objectives, even if they have trouble with the applied objectives. And the question banks that we provide do a thorough job of testing these objectives.

To help you get the most from the behavioral objectives, we have included them at the start of the PowerPoint slides for each chapter as well as in the Student Workbook. Then, the self-study questions in the Student Workbook test whether a student can meet the objectives. And the questions in our question banks also test whether a student can meet the objectives (and only those objectives). As we see it, if you can convince your students that they only need to be able to do the skills that are described by the objectives, their study becomes far more focused and efficient.

Question banks for tests

To test comprehension, the Instructor's CD provides question banks that contain three types of questions: multiple-choice, completion, and problem. True to the theory behind the use of behavioral objectives, all of the questions in these banks are designed to test the skills that are described in the behavioral objectives. As a result, if your students can do all of the skills that the objectives describe, they should be able to pass the tests.

For this book, we've put extra effort into the question banks because that was the most criticized component of the Instructor's CD for the first edition. As a result, there are more than 45 questions for each chapter, and the problem questions do a much better job of testing the applied objectives. So once you have our question banks in the format of your choice, you should be able to generate some excellent tests from them.

Although we developed these question banks in ExamView (which is a terrific product), we provide them on the CD in both Rich Text (RTF) and ExamView format. If you prefer the tests in some other format like WebCT or Blackboard, though, please let us know. To do that, just email mike@murach.com or kelly@murach.com, and we'll do our best to accommodate you.

Exercise solutions

For each exercise in the book, the Instructor's CD provides a satisfactory solution. That way, you can present the solutions in class and compare our solutions with the solutions that the students come up with. You can also run the applications with the data that's provided to demonstrate the applications to your students.

Student projects

In case you want to modify the student projects that are in the Student Workbook, the Instructor's CD consolidates them all in a single Word file. This version of the student projects also includes a summary of all the projects along with difficulty levels and instructor notes.

If you want to add, delete, or modify our projects to suit your purposes, you can do that by using this Word document. Then, you can distribute the modified projects to your students.

Project solutions

All of the solutions to the student projects in the Student Workbook are also included on the Instructor's CD. Then, you can present them in class or compare them with the student solutions. And you can run them to demonstrate how they work.

PowerPoint slides

Because our book uses the paired-pages method of presentation, all of the critical information is presented in the figures. Then, in the PowerPoint slides, we present abridged versions of that information. That includes all of the charts, screens, diagrams, tables, and code that you may want to review in class, but it omits the details that are best left to the textbook. As a result, you can use the slides to review all of the skills that are presented in the book, especially those that your students have difficulty with. In addition, the slides for each chapter start with the objectives that are presented in the Student Workbook.

If you want to modify any of the PowerPoint slides, you should know that we prepared the slides by copying the Word text from our figures into PowerPoint. As a result, you can't use PowerPoint to modify the text in the normal way. Instead, you need to double-click on the text for a slide to open it up in Word, make modifications to the text in Word, and click outside the text to return to PowerPoint. If you try this, though, you'll see that it's an easy process. You can also use PowerPoint in the normal way to add slides, delete slides, or add your own presentation notes to the slides.

PDF documents that expand the contents of the text

When we selected the contents for *Murach's Beginning Java 2, JDK 5*, we tried to include all of the skills that every Java developer needs to have without getting into Servlets and JSPs and without going over 800 pages. That meant that we had to omit some of the marginal skills that aren't needed by every developer. Unfortunately, this is a subjective process, and not everyone will agree with the choices that we made.

For that reason, we intend to provide some PDF documents that can be downloaded from our web site. In general, these documents will use our "paired pages" format just like the chapters in our book. Our goal in offering these documents is to provide all of the content that you want to offer in your courses.

The BlueJ tutorial that I've already mentioned is one of the documents that will be available from our web site. Another is a tutorial on the printf method. But we're open to your ideas for other documents. So, if you have a suggestion, please email mike@murach.com or call Mike at 1-800-221-5528 Extension 15.

BlueJ tutorial

This is a 16-figure document that shows how to use BlueJ as the IDE in this course. In particular, it shows how to use BlueJ to design, code, and test the applications, exercises, and projects that are part of the course materials for our book.

Printf tutorial

This is a short tutorial that shows how to use the new printf method to display formatted data on the console. We omitted this from our book because we thought it was a marginal new feature and because we didn't want to put too much emphasis on console applications. We realize, though, that some competing books present it. So if you want to present it, this tutorial is available.

If you're upgrading from the first edition…

If you used the first edition of *Murach's Beginning Java 2* for one or more of your Java courses, you may be interested in how the JDK 5 edition and its instructional materials have been improved. The topics that follow summarize the changes. As we see it, if you liked the first edition, you're going to be delighted by the second edition because it's a major improvement.

How the book has been improved

Of course, the new edition includes all of the major new features that come with JDK 5.0. But even more important, the book has been re-engineered from start to finish based on the comments and criticism that we've received. The result is a book that is not only easier for the students, but also far better technically. For instance:

- The introductory section of the book has been reorganized and rewritten so it now uses the new Scanner class to get user input. This makes the book easier for first-time programming students as well as students who have already taken a programming course.
- The object-oriented programming section has been reorganized and rewritten so it takes OOP to another level. In particular, the book has expanded and enhanced the treatment of inheritance, polymorphism, and interfaces. And it shows how to use a three-tiered architecture and the factory pattern to develop Java applications.
- The core Java features in section 3 are treated more carefully than in the first edition so they're easier to learn. They are treated more thoroughly so they're stronger technically. And they're integrated with the concepts of object-oriented programming that were established in section 2 so the OO perspective is maintained.
- The GUI skills in section 4 are treated more carefully so they're easier to master. In particular, this section does a much better job of presenting the skills for handling events and validating data, which are critical features of GUI applications. On the other hand, some of the content in the first edition like menus, fonts, and colors has been dropped because we thought they were marginal skills and because we wanted to reduce the emphasis that the first edition put on GUI applications.

Although there's a lot more to what we did as we re-engineered the JDK 5 edition of our Java book, this should give you a better idea of what the major changes are.

How the instructor's materials have been improved

Two of the most criticized components in the instructor's materials for the first edition were the student projects and the question banks. So this time, we're providing a wider range of projects. We're offering questions banks with more questions, better questions, and three types of questions. We're providing the question banks in both RTF and ExamView format. And we can also provide them in WebCT or Blackboard format.

Besides that, we're offering a Student Workbook and downloadable PDF documents that enhance the contents of the text. And we'll be glad to add any other components if that what it takes to get your adoption. So, please let us know what you need!

Your comments, please!

If you have any comments about or suggestions for *Murach's Beginning Java 2, JDK 5* or any of its student or instructor's materials, we would appreciate hearing from you. We'll also be glad to answer any questions that you have. And if you're going to adopt, please let us know so we can keep you up-to-date on any enhancements to the instructional materials.

The easiest way to reach us is to e-mail or call us. And thanks for reviewing our book and course materials.

Mike Murach
Publisher
mike@murach.com
800-221-5528 Ext. 15

Doug Lowe
Author
doug@murach.com
800-221-5528 Ext. 25

Joel Murach
Author
joelmurach@yahoo.com
510-410-1560