

Instructor's Summary for *Murach's Beginning Java with NetBeans*

This summary introduces you to the instructor's materials we've developed for this book and helps you get started using them. At the least, we recommend that you read the topics under "What's included in the instructor's materials," because they not only describe the instructor resources but also our underlying instructional philosophy.

What's included in the student download	2
Book applications	2
Exercise starts.....	2
Exercise solutions	2
What's included in the instructor's materials	2
Book applications, exercise starts, and exercise solutions.....	3
Objectives	3
PowerPoint slides.....	3
Tests	4
Extra exercises, starts, and solutions	4
Projects and solutions	4
How to get started.....	5
The directories and files that are installed	5
How to install on a PC	6
How to install on a Mac	6
Need anything else?	6



Mike Murach & Associates, Inc.

1-800-221-5528 • (559) 440-9071 • Fax: (559) 440-0963

murachbooks@murach.com • www.murach.com

Copyright © 2015 Mike Murach & Assoc. Inc. All rights reserved.

What's included in the student download

To help your students get the most from our book, our website lets them download a file that includes (1) the book applications, (2) starting code for the chapter exercises in the book, and (3) the *solutions* to the exercises. The appendixes in the book show them how to download and set up these materials on their own Windows or Mac system.

Book applications

The book applications are the full sample applications that are presented in the chapters of the book, and they're stored in a top-level folder named *book_apps*. Once the students download these applications, they can run them to see how they work. They can review the code in all of the files of the applications. And they can copy code from the book applications for use in their own applications.

Exercise starts

To help your students master the skills that are taught in each chapter, the book provides exercises at the end of each chapter. For each exercise, the students start from folders and files that contain some of the code that the exercise requires. Since the students don't have to start these exercises from scratch, they are able to practice the critical skills that are presented in the chapters without redoing trivial code. This means that they get more practice in less time. When downloaded, these files are stored in a top-level folder named *ex_starts*.

Exercise solutions

The solutions to the exercises in the book are also included in the download in a top-level folder named *ex_solutions*. As a result, your students can get past the learning obstacles that can occur when they're working on their own, avoid getting hung up on trivial coding errors, and check their solutions against the author's solution to see what, if anything, they could have done better. We think this helps students learn faster and better.

We realize, however, that this makes it difficult for you to use the book exercises to test your students. That's why the instructor's materials include a second set of exercises as well as a set of projects that can be used for evaluation purposes, along with their solutions. We don't make these available to anyone other than instructors and trainers.

What's included in the instructor's materials

The instructor's materials for *Murach's Beginning Java with NetBeans* will help any college instructor or corporate trainer run an effective course based on the book. Besides the materials in the student download, these resources include instructional objectives, PowerPoint slides, tests, an extra set of exercises, solutions to those exercises, projects, and solutions to those projects. A summary of these materials follows.

Book applications, exercise starts, and exercise solutions

These are the same materials that your students can download from our website. We've included them in the instructor's materials so that you can easily demonstrate and review the book applications and exercise solutions in class, without having to download them yourself.

Objectives

Since we believe that instructional objectives should be the start of any educational methodology, we provide a set of objectives for each chapter in the book. We prepared these objectives based on the principles presented by Robert F. Mager in his classic book, *Preparing Instructional Objectives*. As a result, our objectives describe the skills that your students should have when they complete a chapter, and you should be able to test whether they have those skills.

If you review the objectives, you'll see that the first objectives for each chapter are what we refer to as *applied objectives*. These objectives ask the students to apply what they've learned as they develop Java applications. They are the critical objectives of a programming course, and they are best tested by having the students do projects like the ones that we provide.

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. These objectives determine whether your students are able to talk intelligently about the topics that are presented. And these objectives can be tested by the questions in our test banks.

To help you get the most from the instructional objectives, we have included them at the start of the PowerPoint slides for each chapter. As we see it, if you can convince your students that they only need to be able to learn the skills that are described by the objectives, their study becomes far more focused and efficient.

Note, however, that we also provide our objectives as a Word document so you can easily modify them to make them more appropriate for your class. For instance, you may want to delete some of our objectives or add some of your own. That's a good way to focus your students on the critical learning objectives for your particular course.

PowerPoint slides

The PowerPoint slides present abridged versions of all of the information that's presented in the figures of the book. That includes all of the diagrams, screens, tables, and code that you may want to review in class. As a result, these slides make it easy to review any of the skills that your students have difficulty with. In addition, the slides for each chapter start with the instructional objectives so you can review them in class.

If you want to modify any of the PowerPoint slides, you should know that we prepared them 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. You can also use PowerPoint in the normal way to add slides, delete slides, or add your own presentation notes to the slides.

Tests

To test comprehension, you can use the test banks that we've created; there's one for each chapter in the book. We developed these test banks in ExamView, and we provide them in multiple formats: ExamView format, the current Blackboard formats, Respondus format, and Rich Text Format (RTF).

Each test bank provides questions that are designed to test the skills that are described by the objectives for that chapter, and each test question is designed to test the skill described by one objective. This keeps the promise to the students that they are only expected to have the skills that are described by the objectives.

In our test banks, we use only completion and multiple-choice test questions because they have the highest validity. To us, that means that the students who get the best scores are also the ones with the best knowledge and skills. In contrast, matching and true/false questions have low validity, so we don't use them.

Extra exercises, starts, and solutions

In the document named *Extra exercises*, you'll find a second set of chapter exercises that aren't in the book. Since each of these exercises can be done in 15 to 45 minutes, they provide a quick way to test the progress of your students. Some of these exercises also give the students a chance to apply what they've learned in a new context.

We've provided these exercises as a Word document, as well as a PDF, so you can easily modify them if you want to customize them for your class. We've also provided the starting folders and files for these exercises so you can distribute them to your students, as well as the folders and files for the solutions so you can demonstrate and review them in class. And in contrast to the book exercises, the solutions to the extra exercises are provided *only* to instructors and trainers.

Projects and solutions

The best way to test your students' ability to develop Java applications is to have them develop complete applications from scratch. To that end, we provide a document named *Projects*. In it, each project coincides with a designated chapter and has the student create a Java application using the skills that have been learned so far.

To make the projects as useful as possible, we've tried to provide a range of difficulty levels. Your students should be able to finish some in an hour or two. As a result, you can use these projects as tests that can be done in computer lab. That's the only sure way to see whether your students have the skills described by the applied objectives for this book. Other projects may take a couple of days to finish. As a result, you can assign these projects as midterm or final projects.

As with the extra exercises, we provide these project descriptions in a Word document, as well as a PDF, so it's easy for you to modify them. That way, you can make them easier or harder so they're more appropriate for your class.

How to get started

You can get the instructor's materials as a download from our instructor website (www.murachforinstructors.com) or on an Instructor's CD. If you download the materials, you'll need to install them on your computer as described on the following page. If you get the CD, you can do a preliminary review of our materials by opening the CD files. But if you adopt the book, you'll want to install them on your computer.

The directories and files that are installed

The table that follows presents the directories and files that get installed on your computer. All of them are stored in the `murach\java_netbeans` directory. Within that directory, some of the subdirectories (`book_apps`, `db`, `ex_starts`, `ex_solutions`, and `files`) are the same as the directories that are installed on your students' computers when they download the source code for this book from our website.

However, the rest of the directories are only available to instructors. These include the solutions to the extra set of exercises, the solutions to the projects, and the starting points for both, which you can distribute to your students. In addition, the directory named *instructors* includes the course objectives as well as the slides and test banks.

	<code>murach\java_netbeans\</code>	Contents
Students	<code>book_apps\</code>	One project for each book app.
	<code>db\</code>	The SQL script that creates the database that's used by chapters 19-22.
	<code>ex_starts\</code>	One starting point for each exercise in the book.
	<code>ex_solutions\</code>	One solution for each exercise in the book.
	<code>files\</code>	The files that are used in chapter 17.
Instructors only	<code>extra_ex_starts\</code>	One starting point for each exercise in the <i>Extra exercises</i> document.
	<code>extra_ex_starts\Extra exercises.docx</code> <code>extra_ex_starts\Extra exercises.pdf</code>	A document in both Word and PDF formats with additional exercises that are similar to those in the book.
	<code>extra_ex_solutions\</code>	One solution for each exercise in the <i>Extra exercises</i> document.
	<code>project_starts\</code>	Starting files for the projects in the <i>Projects</i> document that use starting files.
	<code>project_starts\Projects.docx</code> <code>project_starts\Projects.pdf</code>	A document in both Word and PDF formats with descriptions of projects that are done from scratch.
	<code>project_solutions\</code>	One solution for each project in the <i>Projects</i> document.
	<code>instructors\Instructor's summary.pdf</code>	This instructor's summary in PDF format.
	<code>instructors\Objectives.docx</code> <code>instructors\Objectives.pdf</code>	A document in both Word and PDF formats that contains the instructional objectives for all chapters.
	<code>instructors\slides\</code>	One PowerPoint file for each chapter.
	<code>instructors\test_banks\</code>	ExamView, RTF, Blackboard, and Respondus directories that contain one test bank for each chapter in the book.

How to install on a PC

1. Find the .exe file that you downloaded from our instructor website *or* the file named `Install.exe` that's in the root of the Instructor's CD.
2. Double-click on the .exe file and respond to the dialog boxes that follow. This will install the folders and files onto your C drive in a folder structure that starts with:

`C:\murach\java_netbeans`

How to install on a Mac

1. Create a directory named *murach* directly on your hard drive.
2. Find the .zip file that you downloaded from our instructor website or the file named `MacInstall.zip` that's in the root of the Instructor's CD.
3. Drag the .zip file to the *murach* directory on your Mac.
4. Double-click on the .zip file, and it will unzip the files and folders into a folder structure that starts with

`murach/java_netbeans`

Need anything else?

If you have any comments about or suggestions for our book or its instructional materials, we would be delighted to hear from you. We'll also be glad to answer any questions that you have. Thanks for reviewing our book and course materials!

Joel Murach, Author
joel@murach.com

Mike Murach, Publisher
murachbooks@murach.com