

# Instructor's Summary for *Murach's Java Programming* (6<sup>th</sup> Edition)

Welcome to the instructor's materials for *Murach's Java Programming (6<sup>th</sup> Edition)*. The purpose of these materials is to help any college instructor or corporate trainer run an effective course based on the book. This summary introduces you to these materials and helps you get started using them.

<b>What's included in the student download .....</b>	<b>2</b>
Book applications .....	2
Exercise starts .....	2
Exercise solutions .....	2
<b>What's included in the instructor's materials .....</b>	<b>2</b>
Book applications, exercise starts, and solutions .....	2
Objectives .....	3
Test banks .....	3
Projects .....	3
Case studies .....	4
PowerPoint slides .....	4
<b>How to get started with our materials .....</b>	<b>4</b>
How to use the zip file .....	4
The folders in the student download .....	5
The folders and files in the instructor's materials .....	5
<b>Need anything else? .....</b>	<b>5</b>



**Mike Murach & Associates, Inc.**

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

[murachbooks@murach.com](mailto:murachbooks@murach.com) • [www.MurachForInstructors.com](http://www.MurachForInstructors.com)

Copyright © 2022 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 retail website at [www.murach.com](http://www.murach.com) lets them download the source code for (1) the applications presented in the book, (2) the starting points for the exercises at the end of each chapter, and (3) *the solutions to those exercises*. All of this code is provided in both NetBeans and Eclipse format, so you can use the IDE you prefer. And the appendixes in the book show your students how to download and set up these materials on Windows or macOS.

### Book applications

---

All of the Java applications that are presented in the book are included in the download. Once your students have set up these applications on their own systems, they can run them to see how they work, they can review the code, and they can copy the code for use in their own applications.

### Exercise starts

---

Each chapter in the book ends with exercises to help your students master the skills covered in the chapter. To make these exercises as useful as possible, each one has your students start from folders and files that contain some of the routine code that the exercise requires. This allows your students to focus on practicing the critical new skills presented in the chapter, without redoing trivial code. Because of this, they get more real-world practice in less time.

### Exercise solutions

---

The solutions to the exercises in the book are also included in the download. As a result, your students can overcome learning obstacles on their own, avoid getting hung up on trivial coding errors, and check their solutions to see what, if anything, they could have done better. We think this helps students learn faster and better.

But we realize that since we provide the solutions, you can't use the book exercises for testing your students. That's why the instructor's materials include a set of projects and two case studies that you can use for evaluation purposes. The solutions for these projects and case studies are only available to instructors.

## What's included in the instructor's materials

---

The instructor's materials are designed to make it easier for you to teach a course based on the text, to ensure that your students gain the Java skills they'll need on the job, and to evaluate their progress. So besides the materials in the student download, we provide instructional objectives, PowerPoint slides, test banks, projects, and two case studies.

### Book applications, exercise starts, and solutions

---

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

## Objectives

---

We believe that instructional objectives should be the start of any educational methodology. So we provide a set of objectives for each chapter in the book that we developed based on the principles presented in Robert F. Mager's classic book, *Preparing Instructional Objectives*. As a result, (1) our objectives describe the skills that your students should have when they complete a chapter, and (2) 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. These are the critical objectives of a programming course, and they are best tested by having the students do projects and case studies 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 multiple-choice test banks.

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

## Test banks

---

To test comprehension, you can use the test banks that we've created for each chapter. We developed these test banks in ExamView, and we provide them in multiple formats, including those that can be used in various LMSs (like Blackboard, D2L Brightspace, and Canvas), as well as Rich Text Format (for Microsoft Word).

Each test bank provides questions that are designed to test the skills 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 will only be expected to have the skills that are described by the objectives.

In our test banks, we use only multiple-choice questions because they not only are easy to score but also have the highest validity. In other words, the students who get the best scores are typically the ones with the best knowledge and skills. In contrast, matching and true/false questions have low validity, so we don't use them.

## Projects

---

To give your students practice and to test whether they can develop their own applications, the instructor's materials include over 60 projects. For each chapter, the projects range from simple to complex so you can assign the ones that are appropriate for your students. If your students can do the more difficult projects for each chapter, that's proof that they're developing valuable skills.

The instructor's materials also include the project solutions. That way, you can run the projects in class to demonstrate how they should work. You can also review the code for the solutions, which is likely to be written in a way that is more professional than the solutions that your students will develop.

## Case studies

---

To provide a more extensive way to test the programming skills of your students, the instructor's materials include two case studies: one for a Blackjack game and one for a Sales Data application. These case studies can be assigned at key points in your class, and each assignment builds on the previous solution so that your students end up creating a substantial application by the end of the course. To facilitate that, the instructor's materials provide the specifications and solutions for the case studies at various points in the book. For example, you can assign the Blackjack case study after chapter 6, create an object-oriented version after chapter 9, and create a GUI version after chapter 12.

## PowerPoint slides

---

In our book, the figures (or illustrations) on the righthand pages present all of the critical information, including screenshots, syntax, and code. Then, our PowerPoint slides are based on these figures. As a result, you don't have to worry about the slides introducing material that isn't explained fully in the book. Instead, they make it easy for you to review any skills that your students are having trouble with or to answer any questions in class. Beyond the book information, the slides for each chapter start with the instructional objectives so you can review them in class and keep your students focused on what's important.

## How to get started with our materials

---

You can request the instructor's materials for our book at our instructor website ([www.murachforinstructors.com](http://www.murachforinstructors.com)) and download them from your account page there. The download is available as a zip file. Then, you can install the materials on your computer as detailed below.

Once the installation is done, you can thoroughly review all of the materials. In particular, you'll want to run some of the book applications, exercise solutions, project solutions, and case study solutions to see the level of competence that our book develops. You'll also want to click through some of the PowerPoint slides to see how they can help you review and reinforce the information that's presented in the book.

To help you find what you're looking for, the entire directory structure for the instructor's materials is shown on the next page.

## How to use the zip file

---

1. Download the .zip file of instructor's materials from your Murach account page.
2. If necessary, double-click on the .zip file to unzip it into a folder named *java*.
3. If necessary, use the Explorer (Windows) or Finder (macOS) to create a folder named *murach* directly on your hard drive.
4. Move the *java* folder into the *murach* folder to create a directory structure that starts with:

```
/murach/java
```

## The folders in the student download

murach/java/student_download	Contents
db/	The database files presented in chapter 19.
files/	The files used by the applications presented in chapters 9 and 17.
netbeans/	The NetBeans projects for the book applications, exercise starts, and exercise solutions.
eclipse/	The Eclipse projects for the book applications, exercise starts, and exercise solutions.

## The folders and files in the instructor's materials

murach/java/instructors	Contents
Objectives.docx Objectives.pdf	The instructional objectives for all chapters in both Word and PDF format.
projects/ Projects.docx Projects.pdf sqlite-jdbc.jar	The specifications for more than 60 projects, arranged by chapter, in both Word and PDF format. The database driver that's needed by the projects for chapter 19.
projects/netbeans/	The solutions to the projects in NetBeans format.
projects/eclipse/	The solutions to the projects in Eclipse format.
case_studies/ Blackjack.docx Blackjack.pdf Sales Data.docx Sales Data.pdf	The specifications for two case studies in both Word and PDF format.
case_studies/netbeans/	The solutions to the case studies in NetBeans format.
case_studies/eclipse/	The solutions to the case studies in Eclipse format.
slides/	One PowerPoint presentation for each chapter.
test_banks/	One test bank for each chapter, organized by format: ExamView, RTF (Word), and Blackboard (which can be imported into Canvas and D2L Brightspace).

## 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 your interest in our Java book!

Joel Murach, Author  
[joel@murach.com](mailto:joel@murach.com)

Shawn Allen, Academic Rep  
[shawn@murach.com](mailto:shawn@murach.com)