Introduction

What's all the fuss about Java? To help answer that question, I offer a few facts:

- More than a third of the world's programmers use Java as one of their primary programming languages. That's at least 5.2 million programmers. And Java powers more than 52 percent of the world's back-end web services.*
- Ninety percent of all Fortune 500 companies use Java.**
- Websites that use Java include Google, YouTube, LinkedIn, Amazon, and eBay.***
- In 2021, Glassdoor, Inc., ranked jobs based on earnings potential, job satisfaction, and number of available job openings. Among the company's "50 Best Jobs in America for 2021," a career as a Java developer ranked number one.****

Sounds good. Right?

Please, read on.

About This Book

This book isn't the usual dry techie guide. It's written for normal human beings — people with little or no programming experience. In this book, I divide Java into manageable chunks. Each chunk is (more or less) a chapter on its own. I explain concepts in plain language using complete code examples that you can download

* `https://blog.jetbrains.com/idea/2020/09/a-picture-of-java-in-2020`

** `https://blogs.oracle.com/oracleuniversity/post/why-does-java-remain-so-popular`

*** `www.frgconsulting.com/blog/why-is-java-so-popular-developers`

**** `www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm` and run. I keep each code example focused on a few key concepts. I resist the urge to use fancy tricks that impress professional programmers. I expand on concepts that may be difficult for newcomers. I add diagrams to help you visualize important ideas. I provide exercises with each chapter along with solutions to the exercises on the book's website.

Finally, and most importantly — and without question the most significant of all this book's features — I throw in some jokes. I've written some good jokes and lots of bad jokes. (I should say "lots and lots" of bad jokes.) I've hidden Easter eggs in the text. I've added anecdotes about all kinds of topics. Some of the anecdotes are true, and many of them are . . . well, you figure it out.

Foolish Assumptions

In this book, I make a few assumptions about you, the reader. If one of these assumptions is incorrect, you're probably okay. If all these assumptions are incorrect, please buy the book anyway:

- **I assume that you have access to a computer.** Here's the good news: You can run most of the code in this book on almost any computer. The only computers you can't use to run this code are ancient boxes that are more than ten years old (give or take a few years).
- **I assume that you can navigate your computer's common menus and dialog boxes.** You don't have to be a Windows, Linux, or Macintosh power user, but you should be able to start a program, find a file, put a file into a certain directory — that sort of thing. Most of the time, when you follow instructions in this book, you're typing code on the keyboard, not pointing-and-clicking the mouse.
- **I assume that you can think logically.** That's all there is to programming in Java — thinking logically. If you can think logically, you have it made. If you don't believe that you can think logically, read on. You may be pleasantly surprised.
- **I make few assumptions about your computer programming experience (or your lack of such experience).** In writing this book, I've tried to do the impossible: Make the book interesting for experienced programmers yet accessible to people with little or no programming experience. So I assume no particular programming background on your part. If you've never created a loop or indexed an array, that's okay.

On the other hand, if you've done these things (maybe in Visual Basic, Python, or C++), you'll discover some interesting plot twists in Java. The developers of Java took the best ideas in object-oriented programming, streamlined them, reworked them, and reorganized them into a sleek, powerful way of thinking about problems. You'll find many new, thought-provoking features in Java. As you find out about these features, many of them will seem quite natural to you. One way or another, you'll feel good about using Java.

Icons Used in This Book

If you could watch me write this book, you'd see me sitting at my computer, talking to myself. I say each sentence in my head. Most of the sentences, I mutter several times. When I have an extra thought or a side comment that doesn't belong in the regular stream, I twist my head a little bit. That way, whoever's listening to me (usually, nobody) knows that I'm off on a momentary tangent.

Of course, in print, you can't see me twisting my head. I need some other way to set a side thought in a corner by itself. I do it with icons. When you see a Tip icon or a Remember icon, you know that I'm taking a quick detour.

Here's a list of icons that I use in this book:

TIP

A tip is an extra piece of information — a helpful tidbit that the other books may forget to tell you.

WARNING

Everyone makes mistakes. Heaven knows that I've made a few in my time. Anyway, when I think people are especially prone to make a mistake, I mark it with a Warning icon.

REMEMBER

Sometimes I want to hire a skywriting airplane crew. "Barry," says the white smoky cloud, "if you want to compare two numbers, use the double equal sign. Please don't forget to do this." Because I can't afford skywriting, I have to settle for a more modest option: I create a paragraph marked with the Remember icon.

CROSS REFERENCE

"If you don't remember what such-and-such means, see blah-blah-blah," or "For more information, read blahbity-blah-blah."

TRY IT OUT

Writing computer code is an activity, and the best way to learn an activity is to practice it. That's why I've created things for you to try in order to reinforce your knowledge. Many of these are confidence-builders, and some are more challenging. When you first start putting concepts into practice, you'll discover all kinds of issues, quandaries, and roadblocks that didn't occur to you when you started reading about the material. But that's a good thing. Keep at it! Don't become frustrated. Or, if you do become frustrated, visit this book's website (`http://javafordummies.allmycode.com`) for hints and solutions.

This icon calls attention to useful material that you can find online. Check it out!

TECHNICAL STUFF

Occasionally, I run across a technical tidbit. The tidbit may help you understand what the people behind the scenes (the people who developed Java) were thinking. You don't have to read it, but you may find it useful. You may also find the tidbit helpful if you plan to read other (geekier) books about Java.

Beyond the Book

In addition to what you're reading right now, this book comes with a free, access-anywhere Cheat Sheet containing code that you can copy and paste into your own Java program. To get this Cheat Sheet, simply go to `www.dummies.com` and type **Java For Dummies Cheat Sheet** in the Search box.

Where to Go from Here

If you've gotten this far, you're ready to start reading about Java application development. Think of me (the author) as your guide, your host, your personal assistant. I do everything I can to keep things interesting and, most importantly, to help you understand.

If you like what you read, send me a note. My email address, which I created just for comments and questions about this book, is `JavaForDummies@allmycode.com`. If email and chat aren't your favorites, you can reach me instead on Twitter (`@allmycode`) and on Facebook (`www.facebook.com/allmycode`). And don't forget — for the latest updates, visit this book's website. The site's address is `http://javafordummies.allmycode.com`.

1 Getting Started with Java