

Making Everything Easier!™

4th Edition

Beginning Programming with Java®

FOR
DUMMIES®
A Wiley Brand

Learn to:

- Use basic development concepts and techniques with Java
- Debug Java programs and create useful code
- Work with files on your computer's hard drive
- Work with the latest features of Java 8

Barry Burd, PhD

Author of Java For Dummies®



Get More and Do More at Dummies.com®



Start with **FREE** Cheat Sheets

Cheat Sheets include

- Checklists
- Charts
- Common Instructions
- And Other Good Stuff!

To access the Cheat Sheet created specifically for this book, go to www.dummies.com/cheatsheet/beginningprogrammingwithjava

Get Smart at Dummies.com

Dummies.com makes your life easier with 1,000s of answers on everything from removing wallpaper to using the latest version of Windows.

Check out our

- Videos
- Illustrated Articles
- Step-by-Step Instructions

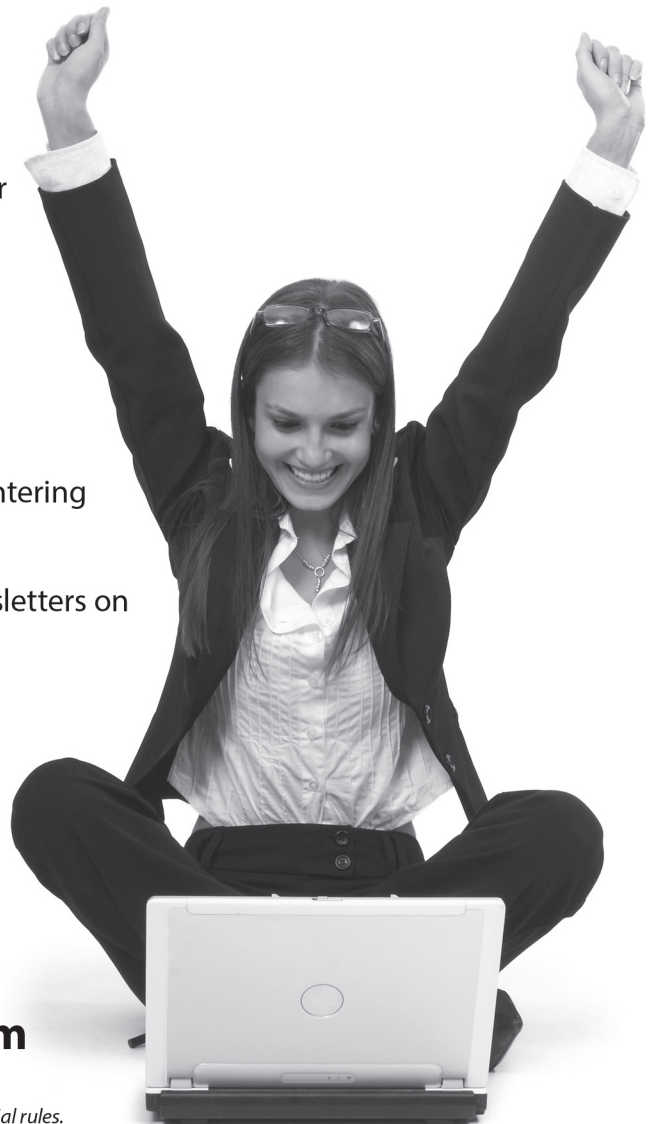
Plus, each month you can win valuable prizes by entering our Dummies.com sweepstakes. *

Want a weekly dose of Dummies? Sign up for Newsletters on

- Digital Photography
- Microsoft Windows & Office
- Personal Finance & Investing
- Health & Wellness
- Computing, iPods & Cell Phones
- eBay
- Internet
- Food, Home & Garden

Find out "HOW" at Dummies.com

*Sweepstakes not currently available in all countries; visit Dummies.com for official rules.



***Beginning Programming
with Java[®]***

FOR
DUMMIES[®]
A Wiley Brand

4th Edition

***Beginning Programming
with Java[®]***

FOR
DUMMIES[®]
A Wiley Brand

4th Edition

by Barry Burd
Author of Java For Dummies

FOR
DUMMIES[®]
A Wiley Brand

Beginning Programming with Java® For Dummies®, 4th Edition

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Copyright © 2014 by John Wiley & Sons, Inc., Hoboken, New Jersey

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Trademarks: Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. Java is a registered trademark of Oracle America, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit www.wiley.com/techsupport.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number: 2013954218

ISBN: 978-1-118-40781-3 (pbk); ISBN 978-1-118-41756-0 (ebk); ISBN 978-1-118-46106-8 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

<i>Introduction</i>	1
<i>Part I: Getting Started with Java Programming</i>	9
Chapter 1: Getting Started	11
Chapter 2: Setting Up Your Computer	23
Chapter 3: Running Programs	53
<i>Part II: Writing Your Own Java Programs</i>	75
Chapter 4: Exploring the Parts of a Program	77
Chapter 5: Composing a Program.....	97
Chapter 6: Using the Building Blocks: Variables, Values, and Types.....	121
Chapter 7: Numbers and Types	135
Chapter 8: Numbers? Who Needs Numbers?	153
<i>Part III: Controlling the Flow</i>	175
Chapter 9: Forks in the Road.....	177
Chapter 10: Which Way Did He Go?	193
Chapter 11: How to Flick a Virtual Switch	217
Chapter 12: Around and Around It Goes.....	233
Chapter 13: Piles of Files: Dealing with Information Overload	253
Chapter 14: Creating Loops within Loops	273
Chapter 15: The Old Runaround	285
<i>Part IV: Using Program Units</i>	309
Chapter 16: Using Loops and Arrays.....	311
Chapter 17: Programming with Objects and Classes	333
Chapter 18: Using Methods and Variables from a Java Class	347
Chapter 19: Creating New Java Methods	371
Chapter 20: Ooey GUI Was a Worm.....	393
<i>Part V: The Part of Tens</i>	423
Chapter 21: Ten Websites for Java	425
Chapter 22: Ten Useful Classes in the Java API	427
<i>Index</i>	431

Table of Contents

<i>Introduction</i>	1
About This Book	1
How to Use This Book	2
Conventions Used in This Book	2
What You Don't Have to Read	3
Foolish Assumptions	3
How This Book Is Organized	4
Part I: Getting Started with Java Programming	4
Part II: Writing Your Own Java Programs	4
Part III: Controlling the Flow	5
Part IV: Using Program Units	5
Part V: The Part of Tens	5
Icons Used in This Book	5
Beyond the Book	6
Where to Go from Here	7
<i>Part I: Getting Started with Java Programming</i>	9
Chapter 1: Getting Started	11
What's It All About?	11
Telling a computer what to do	12
Pick your poison	13
From Your Mind to the Computer's Processor	14
Translating your code	14
Running code	15
Code you can use	20
Your Java Programming Toolset	21
What's already on your hard drive?	22
Eclipse	22
Chapter 2: Setting Up Your Computer	23
If You Don't Like Reading Instructions	24
Getting This Book's Sample Programs	26
Setting Up Java	27
Finding Java on your computer	32
Setting Up the Eclipse Integrated Development Environment	38
Downloading Eclipse	39
Installing Eclipse	40
Running Eclipse for the first time	41
What's Next?	51

Chapter 3: Running Programs 53

Running a Canned Java Program	53
Typing and Running Your Own Code.....	58
Separating your programs from mine.....	59
Writing and running your program	60
What's All That Stuff in Eclipse's Window?.....	68
Understanding the big picture	69
Views, editors, and other stuff.....	70
What's inside a view or an editor?.....	72
Returning to the big picture	74

Part II: Writing Your Own Java Programs* 75*Chapter 4: Exploring the Parts of a Program 77**

Checking Out Java Code for the First Time.....	77
Behold! A program!.....	78
What the program's lines say.....	79
The Elements in a Java Program	80
Keywords	81
Identifiers that you or I can define	83
Identifiers with agreed-upon meanings.....	83
Literals.....	84
Punctuation	85
Comments	87
Understanding a Simple Java Program	88
What is a method?	88
The main method in a program	91
How you finally tell the computer to do something.....	92
The Java class	95

Chapter 5: Composing a Program 97

Computers Are Stupid.....	98
A Program to Echo Keyboard Input	98
Typing and running a program	100
How the EchoLine program works	103
Getting numbers, words, and other things.....	104
Type three lines of code and don't look back.....	105
Expecting the Unexpected.....	107
Diagnosing a problem.....	108
What problem? I don't see a problem.....	118

Chapter 6: Using the Building Blocks: Variables, Values, and Types 121

Using Variables	121
Using a variable.....	122
Understanding assignment statements.....	124
To wrap or not to wrap?	125
What Do All Those Zeros and Ones Mean?	126
Types and declarations.....	127
What's the point?	127
Reading Decimal Numbers from the Keyboard	128
Though these be methods, yet there is madness in 't.....	129
Methods and assignments	129
Variations on a Theme.....	131
Moving variables from place to place.....	131
Combining variable declarations.....	133

Chapter 7: Numbers and Types 135

Using Whole Numbers	135
Reading whole numbers from the keyboard	137
What you read is what you get.....	138
Creating New Values by Applying Operators	139
Finding a remainder.....	140
The increment and decrement operators	144
Assignment operators	147
Size Matters	150

Chapter 8: Numbers? Who Needs Numbers? 153

Characters	154
I digress	155
One character only, please.....	157
Variables and recycling.....	157
When not to reuse a variable	159
Reading characters.....	162
The boolean Type.....	164
Expressions and conditions	165
Comparing numbers; comparing characters	165
The Remaining Primitive Types.....	173

Part III: Controlling the Flow..... 175

Chapter 9: Forks in the Road 177

Decisions, Decisions!.....	177
Making Decisions (Java if Statements)	179
Looking carefully at if statements.....	179
A complete program.....	183
Indenting if statements in your code.....	185



Variations on the Theme	187
. . . Or else what?	187
Packing more stuff into an if statement	189
Some handy import declarations.....	192
Chapter 10: Which Way Did He Go?.....	193
Forming Bigger and Better Conditions	193
Combining conditions: An example.....	195
When to initialize?	198
More and more conditions	199
Using boolean variables.....	201
Mixing different logical operators together	203
Using parentheses	205
Building a Nest.....	206
Nested if statements.....	208
Cascading if statements	209
Enumerating the Possibilities	212
Creating an enum type.....	212
Using an enum type	213
Chapter 11: How to Flick a Virtual Switch	217
Meet the switch Statement.....	217
The cases in a switch statement.....	220
The default in a switch statement	221
Picky details about the switch statement.....	222
To break or not to break.....	225
Using Fall-Through to Your Advantage.....	227
Using a Conditional Operator	230
Chapter 12: Around and Around It Goes	233
Repeating Instructions Over and Over Again (Java while Statements).....	234
Following the action in a loop	235
No early bailout.....	238
Thinking about Loops (What Statements Go Where).....	238
Finding some pieces	239
Assembling the pieces.....	241
Getting values for variables.....	242
From infinity to affinity	243
Thinking about Loops (Priming)	245
Working on the problem.....	248
Fixing the problem.....	250
Chapter 13: Piles of Files: Dealing with Information Overload. . . .	253
Running a Disk-Oriented Program.....	254
A sample program.....	256
Creating code that messes with your hard drive	258

Running the sample program	261
Troubleshooting problems with disk files	264
Writing a Disk-Oriented Program	266
Reading from a file	268
Writing to a file	268
Writing, Rewriting, and Re-rewriting	271
Chapter 14: Creating Loops within Loops	273
Paying Your Old Code a Little Visit	274
Reworking some existing code	275
Running your code	276
Creating Useful Code	276
Checking for the end of a file	277
How it feels to be a computer	279
Why the computer accidentally pushes past the end of the file	280
Solving the problem	282
Chapter 15: The Old Runaround	285
Repeating Statements a Certain Number of Times (Java for Statements)	286
The anatomy of a for statement	288
Initializing a for loop	289
Using Nested for Loops	292
Repeating Until You Get What You Need (Java do Statements)	294
Getting a trustworthy response	295
Deleting a file	297
Using Java's do statement	299
A closer look at the do statement	299
Repeating with Predetermined Values (Java's Enhanced for Statement)	300
Creating an enhanced for loop	301
Nesting the enhanced for loops	303
Part IV: Using Program Units	309
Chapter 16: Using Loops and Arrays	311
Some Loops in Action	311
Deciding on a loop's limit at runtime	313
Using all kinds of conditions in a for loop	315
Reader, Meet Arrays; Arrays, Meet the Reader	317
Storing values in an array	321
Creating a report	322
Working with Arrays	324
Looping in Style	327
Deleting Several Files	329

Chapter 17: Programming with Objects and Classes	333
Creating a Class	334
Reference types and Java classes.....	335
Using a newly defined class.....	335
Running code that straddles two separate files	337
Why bother?	337
From Classes Come Objects.....	338
Understanding (or ignoring) the subtleties.....	340
Making reference to an object's parts.....	341
Creating several objects	341
Another Way to Think about Classes.....	344
Classes, objects, and tables.....	344
Some questions and answers	345
Chapter 18: Using Methods and Variables from a Java Class	347
The String Class	347
A simple example.....	348
Putting String variables to good use	349
Reading and writing strings.....	350
Using an Object's Methods.....	351
Comparing strings	354
The truth about classes and methods	355
Calling an object's methods	357
Combining and using data	357
Static Methods	357
Calling static and non-static methods.....	358
Turning strings into numbers	359
Turning numbers into strings	361
How the NumberFormat works.....	363
Your country; your currency	363
Understanding the Big Picture.....	365
Packages and import declarations	365
Shedding light on the static darkness.....	367
Barry makes good on an age-old promise	368
Chapter 19: Creating New Java Methods	371
Defining a Method within a Class	371
Making a method.....	372
Examining the method's header	373
Examining the method's body.....	374
Calling the method.....	375
The flow of control	376
Using punctuation.....	377
The versatile plus sign	378
Let the Objects Do the Work.....	380

Passing Values to Methods	382
Handing off a value	384
Working with a method header.....	385
How the method uses the object's values.....	386
Getting a Value from a Method	387
An example	387
How return types and return values work.....	389
Working with the method header (again)	391
Chapter 20: Ooey GUI Was a Worm	393
The Java Swing Classes.....	394
Showing an image on the screen	395
Just another class	398
The Swing Classes: Round 2.....	403
Code Soup: Mixing XML with Java.....	406
Using JavaFX and Scene Builder	408
Installing Scene Builder.....	408
Installing e(fx)clipse	409
Creating a bare-bones JavaFX project.....	410
Running your bare-bones JavaFX project.....	411
Adding Stuff to Your JavaFX Project.....	412
Taking Action	417
 <i>Part V: The Part of Tens</i>	 <i>423</i>
Chapter 21: Ten Websites for Java	425
This Book's Website.....	425
The Horse's Mouth.....	425
Finding News, Reviews, and Sample Code	426
Looking for Java Jobs.....	426
Everyone's Favorite Sites	426
 Chapter 22: Ten Useful Classes in the Java API	 427
Applet.....	427
ArrayList	428
File	428
Integer	428
Math	429
NumberFormat.....	429
Scanner	429
String.....	429
StringTokenizer.....	430
System.....	430
 <i>Index</i>	 <i>431</i>

Introduction

What's your story?

- ✔ Are you a working stiff, interested in knowing more about the way your company's computers work?
- ✔ Are you a student who needs some extra reading in order to survive a beginning computer course?
- ✔ Are you a typical computer user — you've done lots of word processing, and you want to do something more interesting with your computer?
- ✔ Are you a job seeker with an interest in entering the fast-paced, glamorous, high-profile world of computer programming (or, at least, the decent-paying world of computer programming)?

Well, if you want to write computer programs, this book is for you. This book avoids the snobby “of-course-you-already-know” assumptions and describes computer programming from scratch.

About This Book

The book uses Java — a powerful, general-purpose computer programming language. But Java's subtleties and eccentricities aren't the book's main focus. Instead, this book emphasizes a process — the process of creating instructions for a computer to follow. Many highfalutin' books describe the mechanics of this process — the rules, the conventions, and the formalisms. But those other books aren't written for real people. Those books don't take you from where you are to where you want to be.

In this book, I assume very little about your experience with computers. As you read each section, you get to see inside my head. You see the problems that I face, the things that I think, and the solutions that I find. Some problems are the kind that I remember facing when I was a novice; other problems are the kind that I face as an expert. I help you understand, I help you visualize, and I help you create solutions on your own. I even get to tell a few funny stories.

How to Use This Book

I wish I could say, “Open to a random page of this book and start writing Java code. Just fill in the blanks and don’t look back.” In a sense, this is true. You can’t break anything by writing Java code, so you’re always free to experiment.

But I have to be honest. If you don’t understand the bigger picture, writing a program is difficult. That’s true with any computer programming language — not just Java. If you’re typing code without knowing what it’s about, and the code doesn’t do exactly what you want it to do, then you’re just plain stuck.

So in this book, I divide programming into manageable chunks. Each chunk is (more or less) a chapter. You can jump in anywhere you want — Chapter 5, Chapter 10, or wherever. You can even start by poking around in the middle of a chapter. I’ve tried to make the examples interesting without making one chapter depend on another. When I use an important idea from another chapter, I include a note to help you find your way around.

In general, my advice is as follows:

- ✓ If you already know something, don’t bother reading about it.
- ✓ If you’re curious, don’t be afraid to skip ahead. You can always sneak a peek at an earlier chapter if you really need to do so.

Conventions Used in This Book

Almost every technical book starts with a little typeface legend, and *Beginning Programming with Java For Dummies*, 4th Edition is no exception. What follows is a brief explanation of the typefaces used in this book:

- ✓ New terms are set in *italics*.
- ✓ When I want you to type something short or perform a step, I use **bold**.
- ✓ You’ll also see this `computerese` font. I use the `computerese` font for Java code, filenames, web page addresses (URLs), onscreen messages, and other such things. Also, if something you need to type is really long, it appears in `computerese` font on its own line (or lines).
- ✓ You need to change certain things when you type them on your own computer keyboard. For example, I may ask you to type

```
class Anyname
```

which means you should type **class** and then some name that you make up on your own. Words that you need to replace with your own words are set in *italicized computerese*.

What You Don't Have to Read

Pick the first chapter or section that has material you don't already know and start reading there. Of course, you may hate making decisions as much as I do. If so, here are some guidelines you can follow:

- ✔ If you already know what computer programming is all about, skip the first half of Chapter 1. Believe me, I won't mind.
- ✔ If you're required to use a development environment other than Eclipse, you can skip Chapter 2. This applies if you plan to use NetBeans, IntelliJ IDEA, or a number of other development environments.

Most of this book's examples require Java 5.0 or later, and some of the examples require Java 7 or later. So make sure that your system uses Java 7 or later. If you're not sure about your computer's Java version or if you have leeway in choosing a development environment, your safest move is to read Chapter 3.
- ✔ If you've already done a little computer programming, be prepared to skim Chapters 6 through 8. Dive fully into Chapter 9 and see whether it feels comfortable. (If so, then read on. If not, re-skim Chapters 6, 7, and 8.)
- ✔ If you feel comfortable writing programs in a language other than Java, this book isn't for you. Keep this book as a memento and buy my *Java For Dummies*, 6th Edition, also published by John Wiley & Sons, Inc.

If you want to skip the sidebars and the Technical Stuff icons, please do. In fact, if you want to skip anything at all, feel free.

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 . . . well, buy the book anyway.

- ✔ **I assume that you have access to a computer.** Here's good news. You can run the code in this book on almost any computer. The only computers you can't use to run this code are ancient things that are more than eight years old (give or take a few years). You can run the latest version of Java on Windows, Macintosh, and Linux computers.
- ✔ **I assume that you can navigate through 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 practice the stuff in this book, you're typing code on your keyboard, not pointing and clicking your mouse.

On those rare occasions when you need to drag and drop, cut and paste, or plug and play, I guide you carefully through the steps. But your computer may be configured in any of several billion ways, and my instructions may not quite fit your special situation. So when you reach one of these platform-specific tasks, try following the steps in this book. If the steps don't quite fit, send me an e-mail message, or consult a book with instructions tailored to your system.

- ✔ **I assume that you can think logically.** That's all there is to computer programming — thinking logically. If you can think logically, you've got it made. If you don't believe that you can think logically, read on. You may be pleasantly surprised.
- ✔ **I assume that you know little or nothing about computer programming.** This isn't one of those "all things to all people" books. I don't please the novice while I tease the expert. I aim this book specifically toward the novice — the person who has never programmed a computer or has never felt comfortable programming a computer. If you're one of these people, you're reading the right book.

How This Book Is Organized

This book is divided into subsections, which are grouped into sections, which come together to make chapters, which are lumped finally into five parts. (When you write a book, you get to know your book's structure pretty well. After months of writing, you find yourself dreaming in sections and chapters when you go to bed at night.) The parts of the book are listed here.

Part I: Getting Started with Java Programming

The chapters in Part I prepare you for the overall programming experience. In these chapters, you find out what programming is all about and get your computer ready for writing and testing programs.

Part II: Writing Your Own Java Programs

This part covers the basic building blocks — the elements in any Java program and in any program written using a Java-like language. In this part, you discover how to represent data and how to get new values from existing values. The program examples are short, but cute.

Part III: Controlling the Flow

Part III has some of my favorite chapters. In these chapters, you make the computer navigate from one part of your program to another. Think of your program as a big mansion, with the computer moving from room to room. Sometimes the computer chooses between two or more hallways, and sometimes the computer revisits rooms. As a programmer, your job is to plan the computer's rounds through the mansion. It's great fun.

Part IV: Using Program Units

Have you ever solved a big problem by breaking it into smaller, more manageable pieces? That's exactly what you do in Part IV of this book. You discover the best ways to break programming problems into pieces and to create solutions for the newly found pieces. You also find out how to use other peoples' solutions. It feels like stealing, but it's not.

This part also contains a chapter about programming with windows, buttons, and other graphical items. If your mouse feels ignored by the examples in this book, read Chapter 20.

Part V: The Part of Tens

The Part of Tens is a little beginning programmer's candy store. In the Part of Tens, you can find lists — lists of tips, resources, and all kinds of interesting goodies.

I added an article at www.dummies.com/extras/beginningprogrammingwithjava to help you feel comfortable with Java's documentation. I can't write programs without my Java programming documentation. In fact, no Java programmer can write programs without those all-important docs. These docs are in web page format, so they're easy to find and easy to navigate. But if you're not used to all the terminology, the documentation can be overwhelming.

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 several times in my head. When I have an extra thought, a side comment, something 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 of setting 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:



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



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



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 something more modest. I create a Remember icon.



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 (more geeky) books about Java.



This icon calls attention to useful material that you can find online. (You don't have to wait long to see one of these icons. I use one at the end of this introduction!)

Beyond the Book

I've written a lot of extra content that you won't find in this book. Go online to find the following:

- ✓ **Cheat Sheet:** Check out www.dummies.com/cheatsheet/beginningprogrammingwithjava.
- ✓ **Online Articles:** On several of the pages that open each of this book's parts, you can find links to what the folks at *For Dummies* call Web Extras, which expand on some concept I've discussed in that particular section. You can find all such extras bundled together at www.dummies.com/extras/beginningprogrammingwithjava.

Where to Go from Here

If you've gotten this far, you're ready to start reading about computer programming. 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, help you understand.



If you like what you read, send me an e-mail, post on my Facebook wall, or tweet me a tweet. My e-mail address, which I created just for comments and questions about this book, is `BeginProg@allmycode.com`. My Facebook page is `/allmycode`, and my Twitter handle is `@allmycode`. And don't forget — to get the latest information, visit one of this book's support websites. Mine is at <http://allmycode.com/BeginProg>, or you can visit www.dummies.com/go/beginningprogrammingwithjavafd.

Occasionally, we have updates to our technology books. If this book does have technical updates, they will be posted at www.dummies.com/go/beginningprogrammingwithjavafdupdates and at <http://allmycode.com/BeginProg>.

Part I

Getting Started with Java Programming



Visit www.dummies.com for great Dummies content online.

In this part . . .

- ✓ Getting psyched up to be a Java developer
- ✓ Installing the software
- ✓ Running some sample programs

Chapter 1

Getting Started

In This Chapter

- ▶ Realizing what computer programming is all about
 - ▶ Understanding the software that enables you to write programs
 - ▶ Revving up to use an integrated development environment
-

Computer programming? What's that? Is it technical? Does it hurt? Is it politically correct? Does Bill Gates control it? Why would anyone want to do it? And what about me? Can I learn to do it?

What's It All About?

You've probably used a computer to do word processing. Type a letter, print it, and then send the printout to someone you love. If you have easy access to a computer, then you've probably surfed the web. Visit a page, click a link, and see another page. It's easy, right?

Well, it's easy only because someone told the computer exactly what to do. If you take a computer right from the factory and give no instructions to this computer, the computer can't do word processing, it can't surf the web, and it can't do anything. All a computer can do is follow the instructions that people give to it.

Now imagine that you're using Microsoft Word to write the great American novel, and you come to the end of a line. (You're not at the end of a sentence; just the end of a line.) As you type the next word, the computer's cursor jumps automatically to the next line of type. What's going on here?

Well, someone wrote a *computer program* — a set of instructions telling the computer what to do. Another name for a program (or part of a program) is *code*. Listing 1-1 shows you what some of Microsoft Word's code may look like.

Listing 1-1: A Few Lines in a Computer Program

```
if (columnNumber > 60) {  
    wrapToNextLine();  
} else {  
    continueSameLine();  
}
```

If you translate Listing 1-1 into plain English, you get something like this:

```
If the column number is greater than 60,  
    then go to the next line.  
Otherwise (if the column number isn't greater than 60),  
    then stay on the same line.
```

Somebody has to write code of the kind shown in Listing 1-1. This code, along with millions of other lines of code, makes up the program called Microsoft Word.

And what about web surfing? You click a link that's supposed to take you directly to Yahoo.com. Behind the scenes, someone has written code of the following kind:

```
Go to <a href="http://www.yahoo.com">Yahoo</a>.
```

One way or another, someone has to write a program. That someone is called a *programmer*.

Telling a computer what to do

Everything you do with a computer involves gobs and gobs of code. For example, every computer game is really a big (make that “very big!”) bunch of computer code. At some point, someone had to write the game program:

```
if (person.touches(goldenRing)) {  
    person.getPoints(10);  
}
```

Without a doubt, the people who write programs have valuable skills. These people have two important qualities:

- ✓ They know how to break big problems into smaller step-by-step procedures.
- ✓ They can express these steps in a very precise language.

A language for writing steps is called a *programming language*, and Java is just one of several thousand useful programming languages. The stuff in Listing 1-1 is written in the Java programming language.

Pick your poison

This book isn't about the differences among programming languages, but you should see code in some other languages so you understand the bigger picture. For example, there's another language, Visual Basic, whose code looks a bit different from code written in Java. An excerpt from a Visual Basic program may look like this:

```
If columnNumber > 60 Then
    Call wrapToNextLine
Else
    Call continueSameLine
End If
```

The Visual Basic code looks more like ordinary English than the Java code in Listing 1-1. But, if you think that Visual Basic is like English, then just look at some code written in COBOL:

```
IF COLUMN-NUMBER IS GREATER THAN 60 THEN
    PERFORM WRAP-TO-NEXT-LINE
ELSE
    PERFORM CONTINUE-SAME-LINE
END-IF.
```

At the other end of the spectrum, you find languages like Haskell. Here's a short Haskell program, along with the program's input and output:

```
median aList =
  [ x | x <- aList,
    length([y | y <- aList, y < x]) ==
    length([y | y <- aList, y > x])]
*Main> median [4,7,2,1,0,9,6]
[4]
```

Computer languages can be very different from one another, but in some ways, they're all the same. When you get used to writing `IF COLUMN-NUMBER IS GREATER THAN 60`, you can also become comfortable writing `if (columnNumber > 60)`. It's just a mental substitution of one set of symbols for another. Eventually, writing things like `if (columnNumber > 60)` becomes second nature.

From Your Mind to the Computer's Processor

When you create a new computer program, you go through a multistep process. The process involves three important tools:

- ✔ **Compiler:** A compiler translates your code into computer-friendly (human-unfriendly) instructions.
- ✔ **Virtual machine:** A virtual machine steps through the computer-friendly instructions.
- ✔ **Application programming interface:** An application programming interface contains useful prewritten code.

The next three sections describe each of the three tools.

Translating your code

You may have heard that computers deal with zeros and ones. That's certainly true, but what does it mean? Well, for starters, computer circuits don't deal directly with letters of the alphabet. When you see the word *Start* on your computer screen, the computer stores the word internally as 01010011 01110100 01100001 01110010 01110100. That feeling you get of seeing a friendly looking five-letter word is your interpretation of the computer screen's pixels, and nothing more. Computers break everything down into very low-level, unfriendly sequences of zeros and ones and then put things back together so that humans can deal with the results.

So what happens when you write a computer program? Well, the program has to get translated into zeros and ones. The official name for the translation process is *compilation*. Without compilation, the computer can't run your program.

I compiled the code in Listing 1-1. Then I did some harmless hacking to help me see the resulting zeros and ones. What I saw was the mishmash in Figure 1-1.

The compiled mumbo jumbo in Figure 1-1 goes by many different names:

- ✔ Most Java programmers call it *bytecode*.
- ✔ I often call it a *.class file*. That's because, in Java, the bytecode gets stored in files named *SomethingOrOther.class*.
- ✔ To emphasize the difference, Java programmers call Listing 1-1 the *source code* and refer to the zeros and ones in Figure 1-1 as *object code*.