

LEARNING MADE EASY



7th Edition

# Java<sup>®</sup>

for  
**dummies**<sup>®</sup>  
A Wiley Brand



Use the new features  
and tools in Java 9

—  
Create basic Java  
objects and reuse code

—  
Handle exceptions  
and events

**Barry Burd, PhD**

*Author of Java Programming for  
Android Developers For Dummies*

- » What Java is
- » Where Java came from
- » Why Java is so cool
- » How to orient yourself to object-oriented programming

# Chapter **1**

## All about Java

**S**ay what you want about computers. As far as I'm concerned, computers are good for just two simple reasons:

» **When computers do work, they feel no resistance, no stress, no boredom, and no fatigue.** Computers are our electronic slaves. I have my computer working 24/7 doing calculations for Cosmo1ogy@Home — a distributed computing project to investigate models describing the universe. Do I feel sorry for my computer because it's working so hard? Does the computer complain? Will the computer report me to the National Labor Relations Board? No.

I can make demands, give the computer its orders, and crack the whip. Do I (or should I) feel the least bit guilty? Not at all.

» **Computers move ideas, not paper.** Not long ago, when you wanted to send a message to someone, you hired a messenger. The messenger got on his or her horse and delivered your message personally. The message was on paper, parchment, a clay tablet, or whatever physical medium was available at the time.

This whole process seems wasteful now, but that's only because you and I are sitting comfortably in the electronic age. Messages are ideas, and physical things like ink, paper, and horses have little or nothing to do with real ideas; they're just temporary carriers for ideas (even though people used them to carry ideas for several centuries). Nevertheless, the ideas themselves are paperless, horseless, and messengerless.

The neat thing about computers is that they carry ideas efficiently. They carry nothing but the ideas, a couple of photons, and a little electrical power. They do this with no muss, no fuss, and no extra physical baggage.

When you start dealing efficiently with ideas, something very nice happens. Suddenly, all the overhead is gone. Instead of pushing paper and trees, you're pushing numbers and concepts. Without the overhead, you can do things much faster and do things that are far more complex than ever before.

## What You Can Do with Java

It would be so nice if all this complexity were free, but unfortunately, it isn't. Someone has to think hard and decide exactly what to ask the computer to do. After that thinking takes place, someone has to write a set of instructions for the computer to follow.

Given the current state of affairs, you can't write these instructions in English or any other language that people speak. Science fiction is filled with stories about people who say simple things to robots and get back disastrous, unexpected results. English and other such languages are unsuitable for communication with computers, for several reasons:

- » **An English sentence can be misinterpreted.** "Chew one tablet three times a day until finished."
- » **It's difficult to weave a very complicated command in English.** "Join flange A to protuberance B, making sure to connect only the outermost lip of flange A to the larger end of the protuberance B, while joining the middle and inner lips of flange A to grommet C."
- » **An English sentence has lots of extra baggage.** "Sentence has unneeded words."
- » **English is difficult to interpret.** "As part of this Publishing Agreement between John Wiley & Sons, Inc. ('Wiley') and the Author ('Barry Burd'), Wiley shall pay the sum of one-thousand-two-hundred-fifty-seven dollars and sixty-three cents (\$1,257.63) to the Author for partial submittal of *Java For Dummies*, 7th Edition ('the Work')."

To tell a computer what to do, you have to use a special language to write terse, unambiguous instructions. A special language of this kind is called a *computer programming language*. A set of instructions written in such a language is called a *program*. When looked at as a big blob, these instructions are called *software* or *code*. Here's what code looks like when it's written in Java:

```
public class PayBarry {  
  
    public static void main(String args[]) {  
        double checkAmount = 1257.63;  
        System.out.print("Pay to the order of ");  
        System.out.print("Dr. Barry Burd ");  
        System.out.print("$");  
        System.out.println(checkAmount);  
    }  
}
```

## Why You Should Use Java

It's time to celebrate! You've just picked up a copy of *Java For Dummies*, 7th Edition, and you're reading Chapter 1. At this rate, you'll be an expert Java programmer\* in no time at all, so rejoice in your eventual success by throwing a big party.

To prepare for the party, I'll bake a cake. I'm lazy, so I'll use a ready-to-bake cake mix. Let me see . . . add water to the mix and then add butter and eggs — hey, wait! I just looked at the list of ingredients. What's MSG? And what about propylene glycol? That's used in antifreeze, isn't it?

I'll change plans and make the cake from scratch. Sure, it's a little harder, but that way I get exactly what I want.

Computer programs work the same way. You can use somebody else's program or write your own. If you use somebody else's program, you use whatever you get. When you write your own program, you can tailor the program especially for your needs.

Writing computer code is a big, worldwide industry. Companies do it, freelance professionals do it, hobbyists do it — all kinds of people do it. A typical big company has teams, departments, and divisions that write programs for the company. But you can write programs for yourself or someone else, for a living or for fun. In a recent estimate, the number of lines of code written each day by programmers in the United States alone exceeds the number of methane molecules on the planet Jupiter.\*\* Take almost anything that can be done with a computer. With the right amount of time, you can write your own program to do it. (Of course, the "right amount of time" may be very long, but that's not the point. Many interesting and useful programs can be written in hours or even minutes.)

---

\*In professional circles, a developer's responsibilities are usually broader than those of a programmer. But, in this book, I use the terms programmer and developer almost interchangeably.

\*\*I made up this fact all by myself.

# Getting Perspective: Where Java Fits In

Here's a brief history of modern computer programming:

»» **1954–1957: FORTRAN is developed.**

FORTRAN was the first modern computer programming language. For scientific programming, FORTRAN is a real racehorse. Year after year, FORTRAN is a leading language among computer programmers throughout the world.

»» **1959: Grace Hopper at Remington Rand develops the COBOL programming language.**

The letter *B* in COBOL stands for *Business*, and business is just what COBOL is all about. The language's primary feature is the processing of one record after another, one customer after another, or one employee after another.

Within a few years after its initial development, COBOL became the most widely used language for business data processing.

»» **1972: Dennis Ritchie at AT&T Bell Labs develops the C programming language.**

The “look and feel” that you see in this book's examples comes from the C programming language. Code written in C uses curly braces, `if` statements, `for` statements, and so on.

In terms of power, you can use C to solve the same problems that you can solve by using FORTRAN, Java, or any other modern programming language. (You can write a scientific calculator program in COBOL, but doing that sort of thing would feel really strange.) The difference between one programming language and another isn't power. The difference is ease and appropriateness of use. That's where the Java language excels.

»» **1986: Bjarne Stroustrup (again at AT&T Bell Labs) develops C++.**

Unlike its C language ancestor, the language C++ supports object-oriented programming. This support represents a huge step forward. (See the next section in this chapter.)

»» **May 23, 1995: Sun Microsystems releases its first official version of the Java programming language.**

Java improves upon the concepts in C++. Java's “Write Once, Run Anywhere” philosophy makes the language ideal for distributing code across the Internet.

Additionally, Java is a great general-purpose programming language. With Java, you can write windowed applications, build and explore databases,

control handheld devices, and more. Within five short years, the Java programming language had 2.5 million developers worldwide. (I know. I have a commemorative T-shirt to prove it.)

- » **November 2000: The College Board announces that, starting in the year 2003, the Computer Science Advanced Placement exams will be based on Java.**

Wanna know what that snot-nosed kid living down the street is learning in high school? You guessed it — Java.

- » **2002: Microsoft introduces a new language, named C#.**

Many of the C# language features come directly from features in Java.

- » **June 2004: Sys-Con Media reports that the demand for Java programmers tops the demand for C++ programmers by 50 percent** (<http://java.sys-con.com/node/48507>).

And there's more! The demand for Java programmers beats the combined demand for C++ and C# programmers by 8 percent. Java programmers are more employable than Visual Basic (VB) programmers by a whopping 190 percent.

- » **2007: Google adopts Java as the primary language for creating apps on Android mobile devices.**
- » **January 2010: Oracle Corporation purchases Sun Microsystems, bringing Java technology into the Oracle family of products.**
- » **June 2010: eWeek ranks Java first among its "Top 10 Programming Languages to Keep You Employed"** ([www.eweek.com/c/a/Application-Development/Top-10-Programming-Languages-to-Keep-You-Employed-719257](http://www.eweek.com/c/a/Application-Development/Top-10-Programming-Languages-to-Keep-You-Employed-719257)).
- » **2016: Java runs on 15 billion devices** (<http://java.com/en/about>), **with Android Java running on 87.6 percent of all mobile phones worldwide** ([www.idc.com/prodserv/smartphone-os-market-share.jsp](http://www.idc.com/prodserv/smartphone-os-market-share.jsp)).

Additionally, Java technology provides interactive capabilities to all Blu-ray devices and is the most popular programming language in the TIOBE Programming Community Index ([www.tiobe.com/index.php/content/paperinfo/tpci](http://www.tiobe.com/index.php/content/paperinfo/tpci)), on PYPL: the Popularity of Programming Language Index (<http://sites.google.com/site/pydata/pypl/PyPL-Popularity-of-Programming-Language>), and on other indexes.

Well, I'm impressed.