15.1 Introduction

Data stored in variables and arrays is *temporary*—it's lost when a local variable goes out of scope or when the program terminates. For long-term retention of data, even after the programs that create the data terminate, computers use **files**. You use files every day for tasks such as writing a document or creating a spreadsheet. Computers store files on **secondary storage devices**, including hard disks, flash drives, DVDs and more. Data maintained in files is **persistent data**—it exists beyond the duration of program execution. In this chapter, we explain how Java programs create, update and process files.

We begin by discussing Java's architecture for handling files programmatically. Next we explain that data can be stored in *text files* and *binary files*—and the differences between them. We demonstrate retrieving information about files and directories using classes `Paths` and `Files` and interfaces `Path` and `DirectoryStream` (package `java.nio.file`), then discuss writing to and reading from files. We create and manipulate text files. As you'll learn, however, it's awkward to read data from text files back into object form. Many object-oriented languages (including Java) provide convenient ways to write objects to and read objects from files (known as *serialization* and *deserialization*). To demonstrate this, we recreate some of our sequential programs that used text files, this time by storing objects in and

retrieving objects from files. We discuss databases in Chapter 24, Accessing Databases with JDBC, and Chapter 29, Java Persistence API (JPA).