6.1 Introduction

Experience has shown that the best way to develop and maintain a large program is to construct it from small, simple pieces. This technique is called **divide and conquer**. Methods, which we introduced in Chapter 3, help you modularize programs. In this chapter, we study methods in more depth.

You'll learn more about `static` methods, which can be called without the need for an object of the class to exist. You'll also learn how Java is able to keep track of which method is currently executing, how local variables of methods are maintained in memory and how a method knows where to return after it completes execution.

We'll take a brief diversion into simulation techniques with random-number generation and develop a version of the dice game called craps that uses most of the programming techniques you've used to this point in the book. In addition, you'll learn how to declare constants in your programs.

Many of the classes you'll use or create while developing applications will have more than one method of the same name. This technique, called *overloading*, is used to implement methods that perform similar tasks for arguments of different types or for different numbers of arguments. We continue our discussion of methods in Chapter 18, Recursion. Recursion provides an intriguing way of thinking about

methods and algorithms.