

KEY IDEA

`printf`'s *format string* says how to format the other arguments.

FIND THE CODE

cho7/formatNumbers/

One of the easiest approaches uses the `printf` method in the `System.out` object. It was added in Java 1.5, and is not available in earlier versions of Java.

The `printf` method is unusual in that it takes a variable number of arguments. It always takes at least one, called the **format string**, that includes embedded codes describing how the other arguments should be printed.

Here's an example where `printf` has three arguments.

```
System.out.printf("%-10s%10s", "Car:", money.format(carPrice));
```

The first argument is the format string. It includes two **format specifiers**, each one beginning with a percent (%) sign and ending with a character indicating what kind of data to print. The first format specifier is for the second argument; the second specifier is for the third argument. Additional specifiers and arguments could easily be added.

In each case, the `s` indicates that the argument to print should be a string. The `10` instructs `printf` to print the string in a field that is 10 characters wide. The minus sign (`-`) in one says to print that string **left justified** (starting on the left side of the column). The specifier without the minus sign will print the string **right justified** (on the right side of the column).

This line, as specified, does not print a newline character at the end; thus, any subsequent output would be on the same line. We could call `println()` to end the line, or we could add another format specifier. The specifier `%n` is often added to the format string to begin a new line. It does *not* correspond to one of the arguments.

Table 7-3 gives several examples of the most common format specifiers and the results they produce. A `d` is used to print a decimal number, such as an `int`. An `f` is used to print a floating-point number, such as a `double`. In addition to the total field width, it specifies how many decimal places to print. More examples and a complete description are available in the online documentation for the `java.util.Formatter` class.

(table 7-3)

Examples of common
format specifiers; dots
signify spaces

Format Specifier and Argument	Result
"%-10s", "Car:"	Car:.....
"%10s", "Car:"Car:
"%10d", 314314
"%10.4f", 3.1415926	3.1416....
"%-10.4f", 3.1415926	...3.1416

The `printf` method has many other options that are documented in the `Formatter` class. Discussing them further, however, is beyond the scope of this book.