Folders & file drawers — objects and classes

In a dentist office, you are likely to see file drawers filled with manila folders. The file drawer named *Patient* contains one manila folder for each patient, with each manila folder in the file drawer containing the same kind of information. Here, we see that manila folder a0 contains Jack Smith's name, address, and the amount he owes. In a real dentist office, the folder would contain much more information.

The file drawer is like a *class* in object-oriented programming. And the manila folders in the file drawer are called *objects*, or *instances* of the class. Each manila folder, or object, has a name on its tab, which identifies it.

Drawing objects

Throughout this course, we will need to draw manila folders, or objects, in order to explain various concepts, and you may have to draw such objects, too, when you are trying to find a bug in a program or to explain something to another person. For efficient communication, we must all use the same form of an object. Here it is:

1. It looks like a front view of a manila folder.

2. In the upper right, we draw a box that contains the name of the class —the file drawer in which it appears.

3. Its tab contains a name, which is different from any other name on a tab. When drawing your own folder, it doesn't matter what name you write, as long as it is different from all other such names.

4. The information in the folder is drawn in a series of *fields* —these are actually variables! The order in which these variables are placed does not matter. Here's a term for you to learn: variables placed in an object like this are called *fields*, or *instance variables*. Memorize that name.

The name on the tab

When you draw an object, put any name on it that you like, as long as it is different from all other such names. In the same way, when an object is created during execution of a Java program on a computer, some arbitrary name will be created and placed on the tab of the folder. You have no control —and don't need control— of the name. It is like the address on your house, which is given to you by the local authorities; you have no control over it and you cannot change it.

Methods in a manila folder (or object)

In dentist office, different people will perform different functions on a folder. The secretary will create the folder and enter the person's name, address, and so forth. The accountant will change the amount owed when the person pays a bill. The dentist himself will make notations within the folder.

To mimic this process, we put into the folder itself the sets of instructions that each person follows for each task. These sets of instructions are called *methods* in OO programming. In a dentist's office, putting detailed instructions into the folder would have the advantage that anyone could perform any task on the folder; just follow the instructions.

We work with two kinds of method here.

One kind of method is the *function*, which yields a value. Here, function getName will yield the name of the person whose folder this is: Jack Smith.

A second kind of method is the *procedure*, which performs some task but does not yield a value. For example, procedure deposit could be given the amount of money being deposited, a **double** value, and it would change field owes accordingly.

Later, we will show you how to write methods in Java. They are much like functions or procedures or subroutines in other programming languages. For now, it is just necessary for you to know that methods — functions and procedures— appear in each and every manila folder of the class. This is one of the important structural features of OO programming.