Figure 4.3 More than one object.

If we're going to instantiate more than one object of the **WashingMachine** class, we'll need a way to uniquely identify each object. For now, we'll do this by storing the machine's identity in a separate data field of the object.

```
class WashingMachine {
    private String name; // the machine's identity
    private String contents; // what's in the machine

    // setIdentity sets the machine's identity.
    public void setIdentity (String id) {
        name = id;
    }

    // getIdentity returns the machine's identity.
    public String getIdentity () {
        return name;
    }
}
```

And here's an example of how we could use these new methods in **TestWash**:

```
public class TestWash {
    public static void main (String[] args) {
        WashingMachine one = new WashingMachine ();
        one.setIdentity("01");
        WashingMachine two = new WashingMachine ();
        two.setIdentity("02");
        System.out.println("Washing machine list: "
            + one.getIdentity() + ", " + two.getIdentity());
    }
}
```

This will print out "Washing machine list: 01, 02".

Note that the two washing machines, **one** and **two**, have distinct data fields. In **one**'s field **name** is the string "01", while in **two**'s field **name** is the string "02". These two **name** fields are *distinct*—**one** and **two** are two *distinct* **WashingMachine** objects.