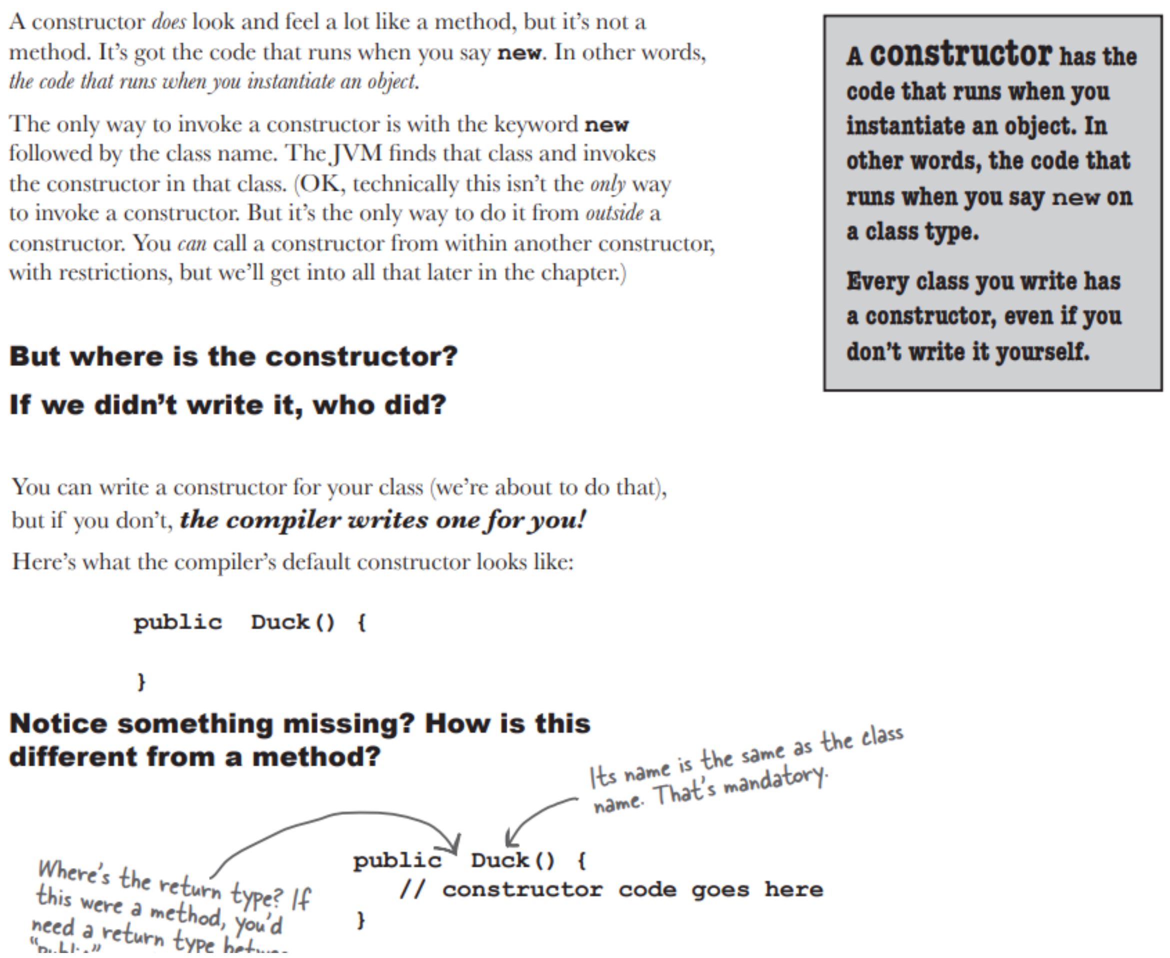
CHAPTER -9

constructors and garbage collection



A constructor *does* look and feel a lot like a method, but it's not a method. It's got the code that runs when you say **new**. In other words, *the code that runs when you instantiate an object.*

The only way to invoke a constructor is with the keyword **new** followed by the class name. The JVM finds that class and invokes the constructor in that class. (OK, technically this isn't the *only* way to invoke a constructor. But it's the only way to do it from *outside* a constructor. You *can* call a constructor from within another constructor, with restrictions, but we'll get into all that later in the chapter.)

A constructor has the code that runs when you instantiate an object. In other words, the code that runs when you say new on a class type.

Every class you write has a constructor, even if you don't write it yourself.

But where is the constructor?

If we didn't write it, who did?

You can write a constructor for your class (we're about to do that), but if you don't, ***the compiler writes one for you!***

Here's what the compiler's default constructor looks like:

```
public  Duck() {

}
```

Notice something missing? How is this different from a method?

```
public  Duck() {
   // constructor code goes here
}
```

Its name is the same as the class name. That's mandatory.
Where's the return type? If this were a method, you'd need a return type bet

EXERCISE

1.GARBAGE COLLECTOR

1. copyGC = null;
2. gc2 = null;
3. newGC = gc3;
4. gc1 = null;
5. newGC = null;
6. gc4 = null;
7. gc3 = gc2;
8. gc1 = gc4;
9. gc3 = null;

Analysis:

1. No—copygc variable is out of scope.
2.Yes—gc2 was the only reference variable
3. No—newgc variable is out of scope.
4. Yes—gc1 was the only reference variable
5. No—newgc is out of scope.
6. No—gc3 refers to object
7. No—gc4 refers to object
8. Yes -
9. No—gc4 refers to object

2.POPULAR OBJECT
Honey object referred by the honeyPot variable is the most "popular" object in this class.There are 13 refernces to that on=bject but atlast as kit ends up in null,there are 12 active references when the code ends.

3. Five-Minute Mystery
Constructor for the SimUnit class would have quickly highlighted the problem!