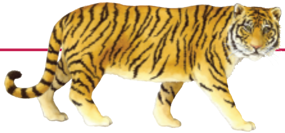

Special Topic 8.4

Alternative Forms of Instance and Static Variable Initialization

As you have seen, instance variables are initialized with a default value (`0`, `false`, or `null`, depending on their type). You can then set them to any desired value in a constructor, and that is the style that we prefer in this book.

However, there are two other mechanisms to specify an initial value. Just as with local variables, you can specify initialization values for instance variables. For example,

```
public class Coin
{
   private double value = 1;
   private String name = "Dollar";
   . . .
}
```

These default values are used for *every* object that is being constructed.

There is also another, much less common, syntax. You can place one or more *initialization blocks* inside the class declaration. All statements in that block are executed whenever an object is being constructed. Here is an example:

```
public class Coin
{
   private double value;
   private String name;
   {
      value = 1;
      name = "Dollar";
   }
   . . .
}
```

For static variables, you use a static initialization block:

```
public class BankAccount
{
   private static int lastAssignedNumber;
   static
   {
      lastAssignedNumber = 1000;
   }
   . . .
}
```

All statements in the static initialization block are executed once when the class is loaded. Initialization blocks are rarely used in practice.

When an object is constructed, the initializers and initialization blocks are executed in the order in which they appear. Then the code in the constructor is executed. Because the rules for the alternative initialization mechanisms are somewhat complex, we recommend that you simply use constructors to do the job of construction.