

Static variables belong to a class, not to any particular object.

## 1 Intuition: a family

A great example of this can be seen in the following class,

```
public class Person {  
    public String firstName;  
    public static String lastName;  
}
```

wherein the expected behavior is that each object (“person”) can have a first name that is unique, or a last name that is unique to that object. However, running *main* in

```
public class m1d21 {  
    public static void main(String[] args) {  
        Person p1 = new Person();  
        Person p2 = new Person();  
  
        p1.firstName = "Alice";  
        p1.lastName = "Adams";  
  
        p2.firstName = "Beverly";  
        p2.lastName = "Boone";  
  
        System.out.println(p1.firstName + " " + p1.lastName);  
        System.out.println(p2.firstName + " " + p2.lastName);  
    }  
}
```

prints the lines,

```
Alice Boone  
Beverly Boone
```

demonstrating that all objects of class *Person* must have the same *lastName*, just like the people in some families will all share a last name.

## 2 Static means “class”

Program *m1d21.java* prints

```
Alice Boone  
Beverly Boone
```

because static variables belong to the class *Person*, not any particular object. So, there is only one copy of *lastName* shared between *p1* and *p2*. In fact, static

variables can be used even if there are no objects, as in program *m1d22.java*, shown below.

```
public class m1d22 {  
    public static void main(String[] args) {  
        Person.lastName = "Boone";  
        System.out.println(Person.lastName);  
    }  
}
```

Program *m1d23.java* will not compile, since *firstName* is not a static variable. Like static variables, static methods can be called using the class name. To prevent a runtime error, static methods are only allowed to use the variables guaranteed to exist when they are called: static variables.

```
public class m1d23 {  
    public static void main(String[] args) {  
        Person.lastName = "Boone";  
        System.out.println(Person.firstName);  
    }  
}
```

### 3 Exercise

Explain why *m1d24.java* prints

0  
1

```
public class m1d24 {  
    public static int x = 1;  
    public static void main(String[] args) {  
        int x = 0;  
        System.out.println(x);  
        System.out.println(m1d24.x);  
    }  
}
```

Can the keyword *this* be used to print 1? Why or why not?