

OOP Gotchas

1. Variables are only references or primitives. A variable is not the same thing as an object.

A variable stores either a primitive type (int, double, etc) or a *reference* to an object. Reference variables do not store objects themselves, which is a common source of confusion.

Examples:

```
int x = 3;
int y = 5;
int z = x;
// Q: How many separate integers are there now?  A: 3
```

```
SimpleCanvas bigCanvas = new SimpleCanvas(400, 400);
SimpleCanvas smallCanvas = new SimpleCanvas(200, 200);
SimpleCanvas thirdCanvas = canvas1;
// Q: How many SimpleCanvas references are there now?  A: 3
// Q: How many SimpleCanvas objects are there now?      A: 2
```

Note: Arrays in Java are *objects*, even if the data type the array holds is a primitive type.

2. When using == on reference types, Java compares the references, not their corresponding objects.

```
Color color1 = new Color(0, 0, 255);
Color color2 = new Color(0, 0, 255);
System.out.println(color1 == color2);           // prints false
System.out.println(color1.equals(color2));     // prints true
```

3. All arguments in Java are passed by value, both reference types and primitive types.

"Pass by value" means that arguments are copied from the calling function to the called function. We normally think of this as each function having its own copy of the variable. This is *technically* true for both primitive types and reference types, but causes confusion for reference types because it appears that no copy is made.

The confusion occurs because a copy of the **reference** is made, not the object that it refers to.

Note: You will sometimes hear that Java uses pass-by-value for primitives and pass-by-reference for objects. This is not true, though this is a common misconception because Java's pass-by-value behavior when references are passed (copying the reference) is very similar, though not identical, to what would happen if true pass-by-reference were used. (I only mention this here because you may see this "fact" online or in textbooks, but you don't have to actually know what pass by reference is.)