

Give em the 'Ol Switcheroo

[Here is a video walkthrough of the solutions.](#)

For each function call in the main method, write out the x and y values of both foobar and baz after executing that line. (Spring '15, MT1)

```
1 public class Foo {
2     public int x, y;
3
4     public Foo (int x, int y) {
5         this.x = x;
6         this.y = y;
7     }
8     public static void switcheroo (Foo a, Foo b) {
9         Foo temp = a;
10        a = b;
11        b = temp;
12    }
13    public static void fliperoo (Foo a, Foo b) {
14        Foo temp = new Foo(a.x, a.y);
15        a.x = b.x;
16        a.y = b.y;
17        b.x = temp.x;
18        b.y = temp.y;
19    }
20    public static void swaperoo (Foo a, Foo b) {
21        Foo temp = a;
22        a.x = b.x;
23        a.y = b.y;
24        b.x = temp.x;
25        b.y = temp.y;
26    }
27
28    public static void main (String[] args) {
29        Foo foobar = new Foo(10, 20);
30        Foo baz = new Foo(30, 40);
31        switcheroo(foobar, baz);    foobar.x: ___ foobar.y: ___ baz.x: ___ baz.y: ___
32        fliperoo(foobar, baz);    foobar.x: ___ foobar.y: ___ baz.x: ___ baz.y: ___
33        swaperoo(foobar, baz);    foobar.x: ___ foobar.y: ___ baz.x: ___ baz.y: ___
34    }
35 }
```

Solution:

line 34: foobar.x: 10 foobar.y: 20 baz.x: 30 baz.y: 40
line 35: foobar.x: 30 foobar.y: 40 baz.x: 10 baz.y: 20
line 36: foobar.x: 10 foobar.y: 20 baz.x: 10 baz.y: 20

Explanation:

switcheroo: Note that `switcheroo` assigns a local variable `temp` to `a`, but never mutates objects, e.g. by reassigning `a.x`. This means that all `switcheroo` does is move around its local pointers to `temp`, `a`, and `b`; nothing in `foobar` or `baz` is actually changed.

fliperoo: Here, `a` points to `foobar` and `b` points to `baz`. `temp` refers to an object with the same initial `x` and `y` values as `a`, which are `10` and `20` respectively. Lines 15 and 16 change `foobar` to have `{x: 30, y: 40}`. Then, lines 17 and 18 allow `baz` to take on the same `x` and `y` values as `temp`, which are `{x: 10, y: 20}`.

swaperoo: In `swaperoo`, instead of creating a new object, we simply point `temp` to the same object as `a`. In lines 22 and 23, we override `foobar`'s `x` and `y` values to become the same as `baz`'s: `{x: 10, y: 20}`. In line 24 and 25, we assign `baz`'s `x` and `y` values to be equal to `temp`'s. But remember, `temp` is pointing to the same object as `a`, which points to `foobar`, and which we just modified to have `{x: 10, y: 20}`. Thus, `baz` does not change.