

Quiz 2 solution

1)

a)

```
public boolean equals(Object obj) {
    if (obj instanceof Palette) {
        Palette p = (Palette) obj;
        // Same name?
        if (!this.name.equals(p.name)) {
            return false;
        }
        // Same number of colors?
        if (this.colors.length != p.colors.length) {
            return false;
        }
        // Same colors in the same order?
        for (int i = 0; i < this.colors.length; i++) {
            if (!this.colors[i].equals(p.colors[i])) {
                return false;
            }
        }
        // At this point, the contents of this and p
        // are the same
        return true;
    } else {
        return false; // not a Palette
    }
}
```

b)

Replace `if (obj instanceof Palette)` with `if (obj != null && obj.getClass() == this.getClass())`.

Using `getClass` assures that the two objects that are compared have the same type and are not just related via an `is_a` relation.

For example, if `Palette` is inherited by `CanvasPalette` (`public class CanvasPalette extends Palette`), and if we define two variables `p` and `cp` of type `Palette` and `CanvasPalette` with the same `Palette` contents.

```
Palette p = new Palette();
```

```
CanvasPalette cp = new CanvasPalette();
```

`p.equals(cp)` returns true if we use `instanceof`, but returns false if we use `getClass()`.

2)

a)

	Case 1	Case 2	Case 3
<code>this.b == copy.b</code>	True	True	False
<code>this.b.intArray == copy.b.intArray</code>	True	True	False
<code>this.b.intArray[0] == copy.b.intArray[0]</code>	True	True	True

Case 1: the copy is an alias

Case 2: the copy is shallow (the copy and the original share the same B object)

Case 3: the copy is deep

b) with an intArray of size 3 and elements 12, -7, and 101

