# Quickcheck 01: Solutions

**Name:**

## Reference Mystery

What does this program print out? (Note: you should probably try and draw diagrams to help you figure out the answer rather then doing it in your head)

```java
import java.util.Arrays;

public class Mystery {
    public static void main(String[] args) {
        int x = 1;
        int[] a = new int[2];
        mystery1(x, a);
        System.out.println(x + " " + Arrays.toString(a));

        x -= 1;
        a[1] = a.length;
        mystery1(x, a);
        System.out.println(x + " " + Arrays.toString(a));

        mystery2(x, a);
        System.out.println(x + " " + Arrays.toString(a));
    }

    public static void mystery1(int x, int[] list) {
        list[x] += 1;
        x += 1;
        System.out.println(x + " " + Arrays.toString(list));
    }

    public static void mystery2(int x, int[] list) {
        list = new int[]{x, x};
        System.out.println(x + " " + Arrays.toString(list));
    }
}
```

**Solution:**

Remember that Java "passes by value" for primitive types (like `ints`) and "passes a reference (aka pointer) by value" for Object types. That means that if you change an `int` inside a function call, it will not be altered at all in the original method. On the other hand, if you mutate an object (i.e. change its fields) in a function call, the changes will be reflected in the original method. Unless you change what the reference is pointing to (by doing an assignment with the "=" operator), any changes after reassignment will not be reflected in the caller.

You can find a line-by-line walkthrough of the code using memory diagrams here: https://docs.google.com/presentation/d/12oMD6vI0D6gDztXZfQ2eYJ1l5Xo68GfbxzW-HOPECpM/edit?usp=sharing. This is the final output of the program:

```
2 [0, 1]
1 [0, 1]
1 [1, 2]
0 [1, 2]
0 [0, 0]
0 [1, 2]
```

# Another question

Do you have any questions about this course? It could be about policy, content, instructors, TAs, etc.