

PracticeProblem.java file:

```
1 public class PracticeProblem {
2     public static void main(String[] args) {
3         int[] a, b;
4         a = new int[2];
5         a[0] = 1; a[1] = 2;
6         b = a;
7         process(a);
8         System.out.println("a[0] = " + a[0]);
9         System.out.println("b[0] = " + b[0]);
10        Coin x, y;
11        x = new Coin(5);
12        y = new Coin(25);
13        x = y;
14        y.split();
15        System.out.println("x = " + x.getValue());
16        System.out.println("y = " + y.getValue());
17    }
18
19    public static void process(int[] z) {
20        for (int i = 0; i < z.length; i++) {
21            z[i] += 5;
22        }
23    }
24 }
```

Coin.java file:

```
1 public class Coin {
2     private int value;
3     public Coin(int v) {
4         value = v;
5     }
6     public int getValue() {
7         return value;
8     }
9     public void split() {
10        value = value/2;
11    }
12 }
```

Determine the code output:

This is a problem that covers references to objects.

First we are declaring two integer arrays on line 3 of the PracticeProblem.java file/class. Then we initialize array *a* as a two element integer array and assign the integers 1 and 2, respectively, to index 0 and 1 of the array. Next, we reference our integer array *b* to array *a*. Since *a* is an object, *b* is referencing the same object. Hence, any changes to *a* will be reflected in *b*. Once we run the method process(*a*), the values in index 0 and 1 will be 6 and 7, respectively. So, the output for both *a*[0] and *b*[0] will be the same (e.g. 6).

The outputs will be the same for *x* and *y* for the same reason as the outputs are the same for *a*[0] and *b*[0].

NOTE: If you are using the *new* keyword, any variable assigned to that same object will share any changes that happen to the object. If the data type is primitive (e.g. int, double, etc), when you reference it, the variable references the value of the primitive. For example: after the following code,

```
1 int x = 1; int y = x;
2 x += 5;
```

we end with *x* = 6 and *y* = 1.

Console output:

```
1 a[0] = 6
2 b[0] = 6
3 x = 12
4 y = 12
```