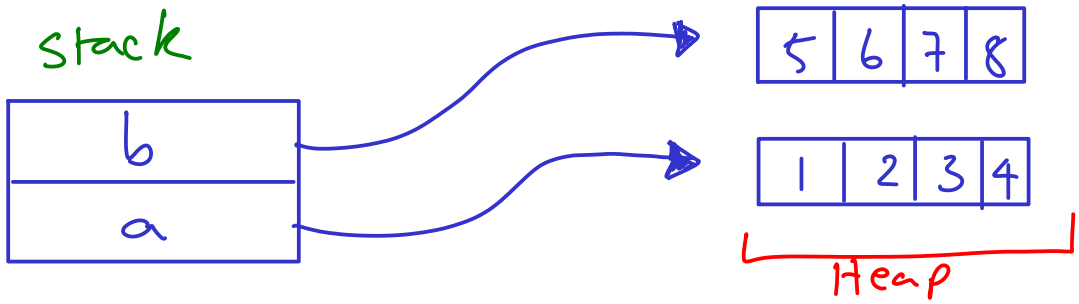
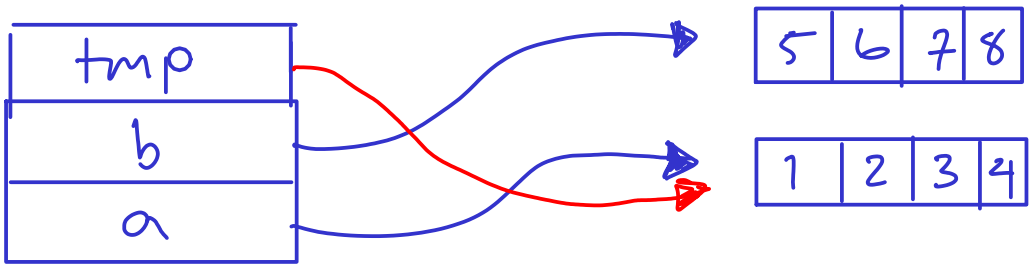
Understanding the reference swapping examples

```
1. public class Check2 {
2.       public static void main(String[] args) {
3.               int a[]={1,2,3,4};
4.               int b[]={5,6,7,8};

5.               int[] tmp = a;
6.               a=b;
7.               b=tmp;
8.               System.out.println(a[0]+" "+b[0]);
9.       }
10. }
```

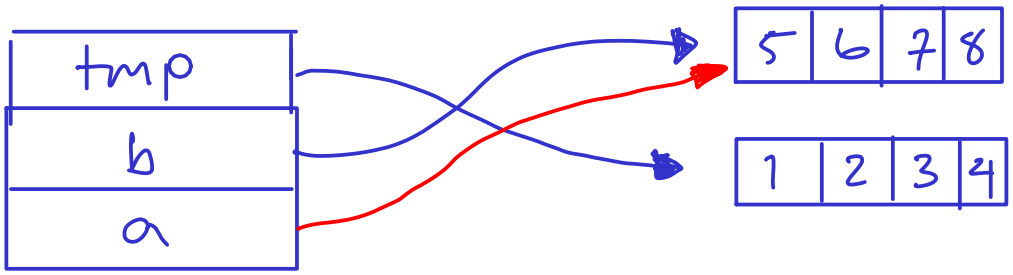
Line 3 declares a to be a (label for a) reference to an integer array, and then initialises it to point to an array that is sitting in the heap with values {1,2,3,4}. Line 4 does similarly for b with the array {5,6,7,8}. i.e.



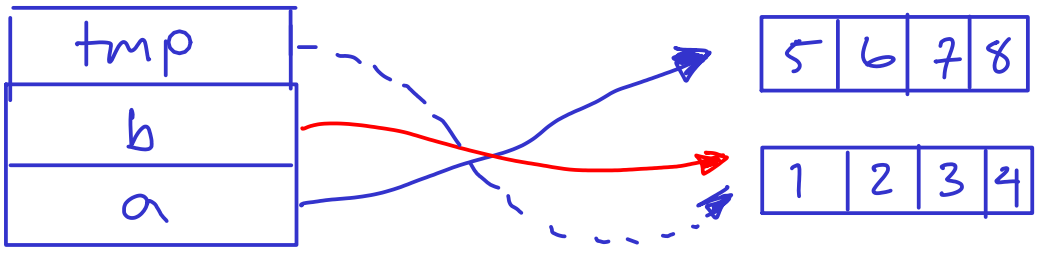
Line 5 declares a new reference to an integer array called tmp that is made equal to a. i.e. both tmp and a contain the same memory address (for the {1,2,3,4} array).



Line 6 reassigns reference a to point to the same as b,

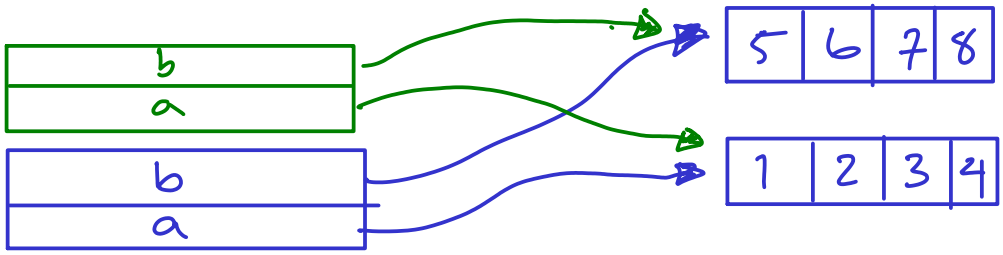


Line 7 then reassigns b to point to the same as tmp. Therefore we get 5 for a[0] and 1 for b[0].

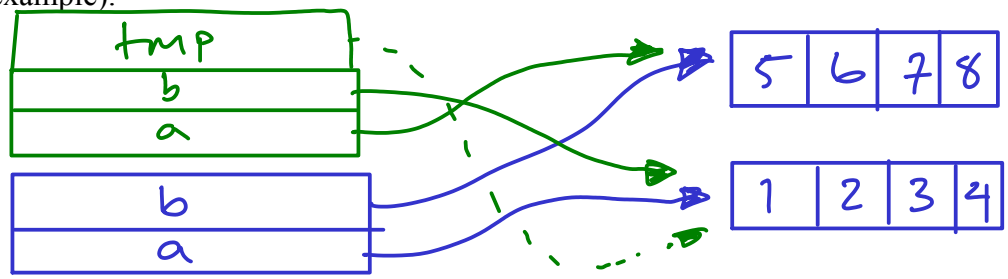


```
1. public class Check2 {

2.      public static void swap(int[] a, int b[]) {
3.              int[] tmp = a;
4.              a=b;
5.              b=tmp;
6.      }

7.      public static void main(String[] args) {
8.              int a[]={1,2,3,4};
9.              int b[]={5,6,7,8};

10.             swap(a,b);
11.             System.out.println(a[0]+" "+b[0]);
12.     }
13. }
```

On the face of it, this looks very similar – I've just moved the swapping code into a function. Unfortunately it has a very different result. When we call swap(...), a new frame is created on the stack. It has two parameters that are references to integer arrays. These parameters are also called a and b. We know Java passes everything by value, so the new a and b and copies of the initial a and b.



The code in the function then succeeds in swapping over the new references (just as it did in the first example).



Now the swap(...) function finishes, and so it is wiped from the stack. We are left with the original a and b, so printing just gives us "1 5" since a[0] is 1 and b[0] is 5: