

CS1020: DATA STRUCTURES AND ALGORITHMS I

Tutorial 1 – Simple OO with Java

(Week 3, starting 25 January 2016)

1. Variables & Message Passing

Explain the behavior of the following three code fragments. Specifically, **why some methods are able to swap** the desired `int` value passed into the argument (on the caller end), while others are **not able to**.

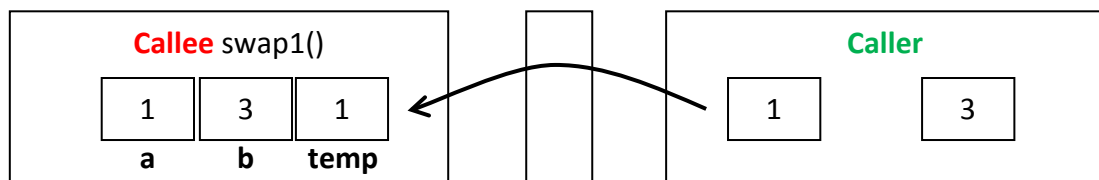
Tip: Try it out! Create a program in vim, paste the fragment in the right place. Compile and run in sunfire.

```
// Are the int values within a and b swapped?
public static void swap1(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
}
```

```
class MyInteger {
    public int x;
    public MyInteger(int n) { x = n; }
    // Are the int values swapped?
    public static void swap2(MyInteger a, MyInteger b) {
        int temp = a.x;
        a.x = b.x;
        b.x = temp;
    }
}
```

```
// Are the int values within a[i] and a[j] swapped?
public static void swap3(int[] a, int i, int j) {
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}
```

Hint: Use diagrams to explain. For example, the following diagram shows the call `swap1(1, 3)` and the values of the local variables `a`, `b` and `temp` after the first statement in the method is executed. What are the values of the local variables after all the statements in the method are executed?



2. Arrays

(a) Draw out what each of these 4 statements does in memory:

```
int[] int1DArray = new int[3];
int[][] int2DInitArray = new int[3][5];
int[][] int2DPartialArray = new int[3][];
Point[] pointArray = new Point[3];
```

(b) After creating these 4 arrays, each array is then used to hold zero or more elements. If we then examine the memory for **each array**, what is the **minimum** and **maximum** number of objects that could possibly be present?

For example, the `pointArray` variable could have 1 to 4 objects present. When the array is first created, it is the only object being referred to by the `pointArray` reference. It could also hold 3 more objects.

3. Objects

The `java.awt.Point` class was mentioned in lecture 3. View the API documentation at:

<https://docs.oracle.com/javase/7/docs/api/java/awt/Point.html>

Complete the following method to “draw” a few identical diamonds at different points on the Cartesian plane. Each diamond is made up of an array of Points.

You are provided with the height and width of each diamond, which is a positive even integer. You are also provided with the centres of where the diamonds are to be drawn, in a 1D `Point` array. The method returns an array of diamonds, i.e. a 2D `Point` array.

```
public static Point[][] drawDiamonds(int width, int ht, Point[] ctrs) {
    /* TODO: create diamonds and return array of diamonds */
}
```

Note: If you have any queries on tutorials, please post on the “Tutorial” forum in IVLE.

- Have fun 😊 -