CS1020: Data Structures and Algorithms I

Tutorial 1 – Simple OO with Java

(Week 3, starting 23 January 2017)

1. *Variables & Message Passing*

In the following three code fragments, **some methods are able to swap** the desired `int` values passed into the argument (on the caller end), while others are **not able to**. Use diagrams to illustrate why this is so.

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;
}
```

no (pass by value)

```
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;
      }
}
```

reference data type begins with capital letter; primitive lowercase letter.

notice the similarity btn arr[n] and a.n

memory address of objects is passed in

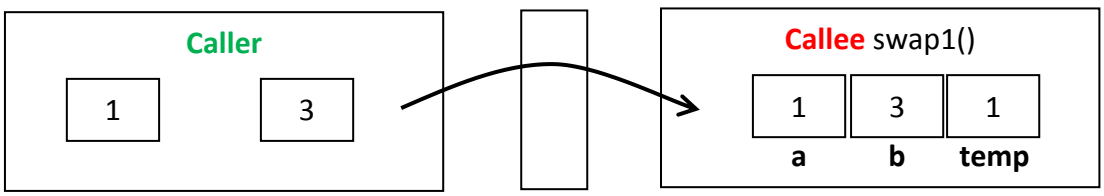
yes, swapped

```
// 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;
}
```

yes, swapped

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, parameters, and arguments, after all the statements in the method are executed?



2. *Arrays*

use
[ref name] = null
to make it not pointing anywhere

(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];
```

int2DArr ref points to int int2Darr obj, which points to 3 int1Darr ob

1 object; max 4 objects

create an array of 3 point objects and return their address
create a ref called "pointArray"
======
min 1 obj (point objects not created); max 4

`Point` here refers to the `java.awt.Point` class.

(b) After creating these arrays through the 4 statements, each array is then used to hold zero or more elements. If we then examine the memory for **each case**, 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 up to 3 objects.

3. *Simple OOP in Java*

You want to print out the lyrics of this song[1], to teach (or confuse =P) kids about the sounds animals make:

> *Dog* goes **woof**
> *Cat* goes **meow**
> *Bird* goes **tweet**
> *Mouse* goes **squeak**
> *Cow* goes **moo**

The lyrics can be generalized for different animals, each having a different *name* and **sound**. With knowledge of object-oriented programming, you want to demonstrate that it is possible to write a program that *displays* the song. To show that your program works, add the 5 animals above and test your program.

Use the skeleton on the next page to solve the problem:

[1] Adapted from "The Fox" by Ylvis, 2013

```
class Animal {
   /* TODO: Implement data and functionality of an Animal here */
}
public class Song {

   private Animal[] _animals;
   private static final int ANIMAL_COUNT = 5;
   public Song() {
      /* TODO: Create your animal array here */
   }

   public void display() {
      for (int i = 0; i < ANIMAL_COUNT; i++)
         System.out.println( /* TODO: Add the lyrics here... */ );
   }

   public static void main(String[] args) {
      (new Song()).display();
   }
}
```

2 att.s: name, sound
2 method: getName(), getSound();

static: variable belong to the Song class; no animal has AC att.; final: inalterable; (static&&final: constant)

create a song object

_animals[] = new Animal[ANIMAL_COUNT];
_animals[0] = new Animal("pig","")

_animals[i].toString();

return the add of the song obj -> dereference to display()

Before & After

As you are attempting each tutorial question, ask yourself what the related concepts learnt in lectures are. If you are unsure of an answer, revise the concept, try reasoning and writing out your answer again, and then test your answer by coding in vim!

If you have any queries on tutorials, please post on the "Tutorial" forum in IVLE.

Before all tutorial groups have finished discussing this tutorial, you may clarify your doubts on the questions, but please do NOT post your answers. After the answers document has been uploaded, you may clarify your answers, even on the forum. Do NOT let your doubts accumulate!

- Hope you had fun, prepare well for tutorial 2 ☺ -

Draw diagrams
Attempt tutorials
Test your solution