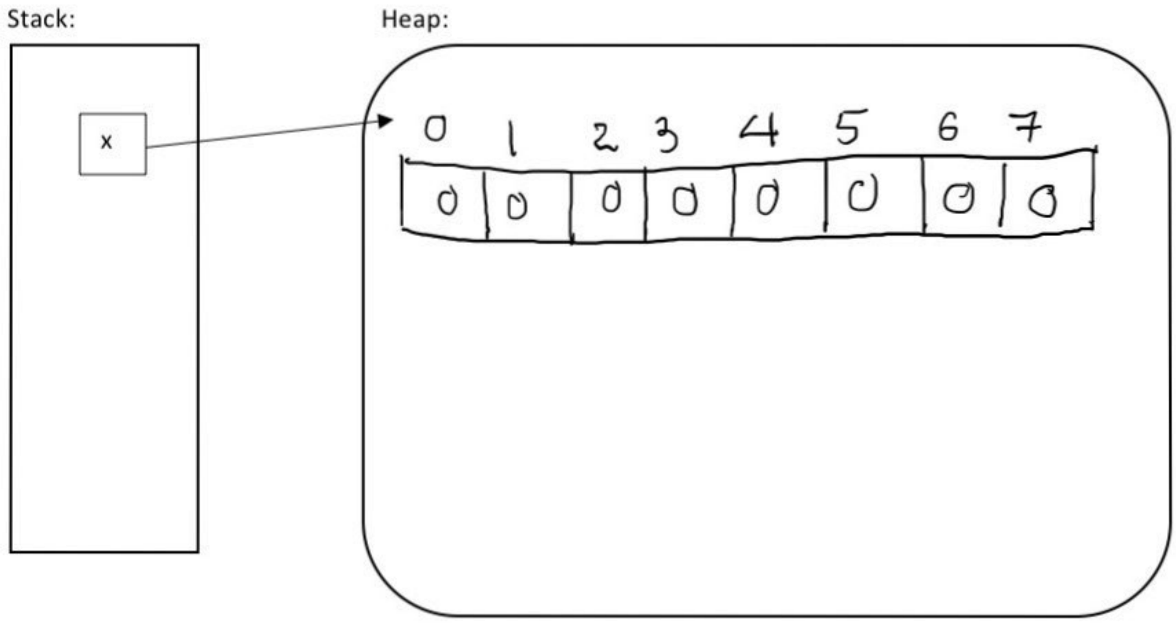
Sethatevy Bong

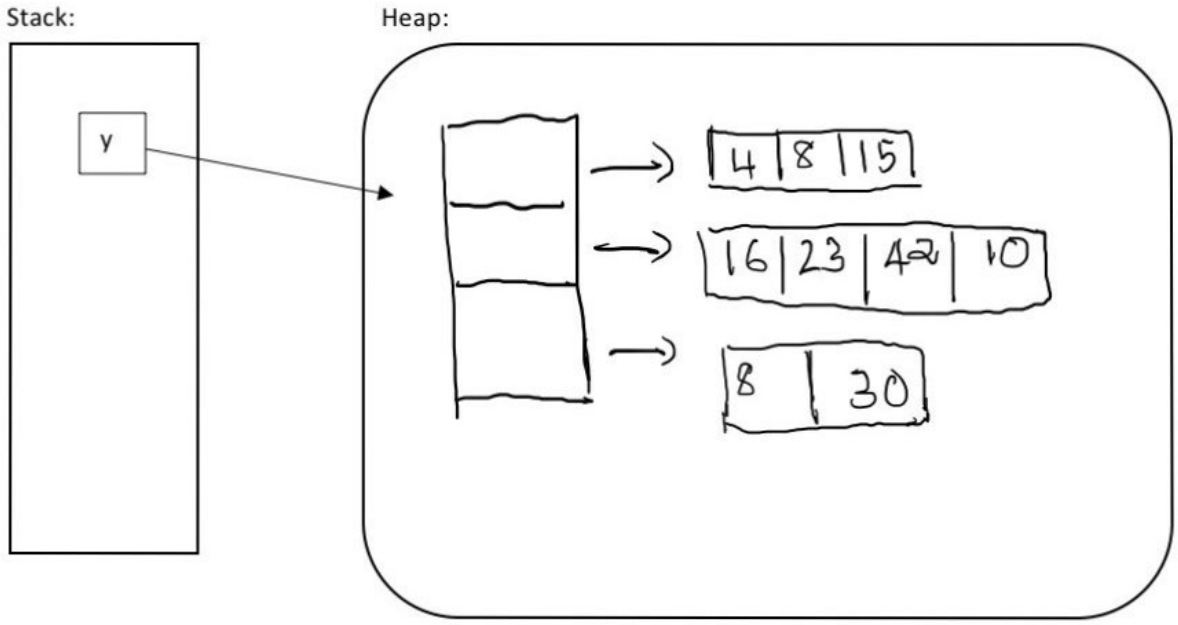
CMSC 203 Spring 2020
Memory Mapping Worksheet

1. Draw the memory map of the following one-dimensional array of type int.
   int[] x = new int[8];



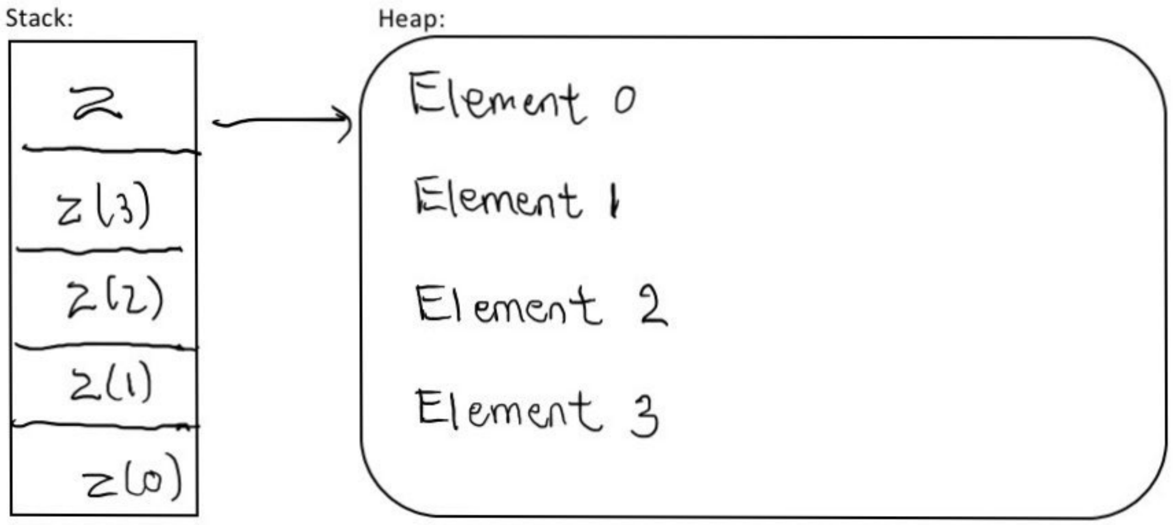
2. Draw the memory map of the following two-dimensional ragged array of ints.

   int[][] y = {{4, 8, 15}, {16, 23, 42, 10}, {8, 30}};



3. Draw the memory map of the following one-dimensional array of type String.

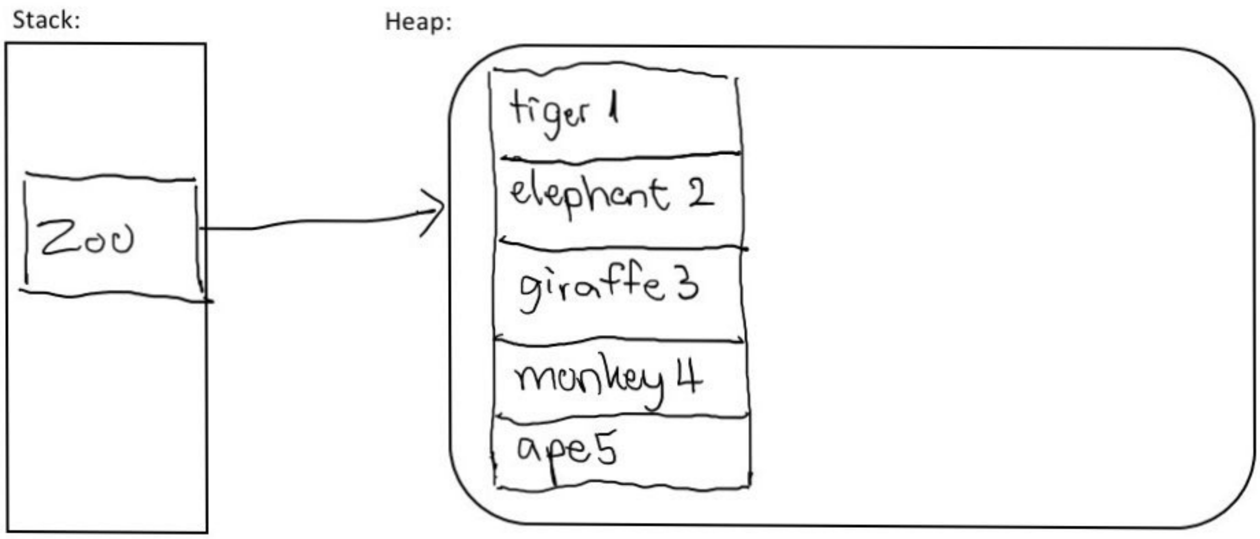
```
String[] z = new String[4];
for (int i = 0; i < z.length; i++) {
    z[i] = "element " + i;
}
```



4. a. Write a shallow copy of the following in code. (Assume the five animal objects are already instantiated.)

```
Animal[] zoo = {tiger1, elephant2, giraffe3, monkey4, ape5};
Animal[] copy;
Animal[] copy = zoo;
```

b. Draw the memory map.



Sethatevy Bong

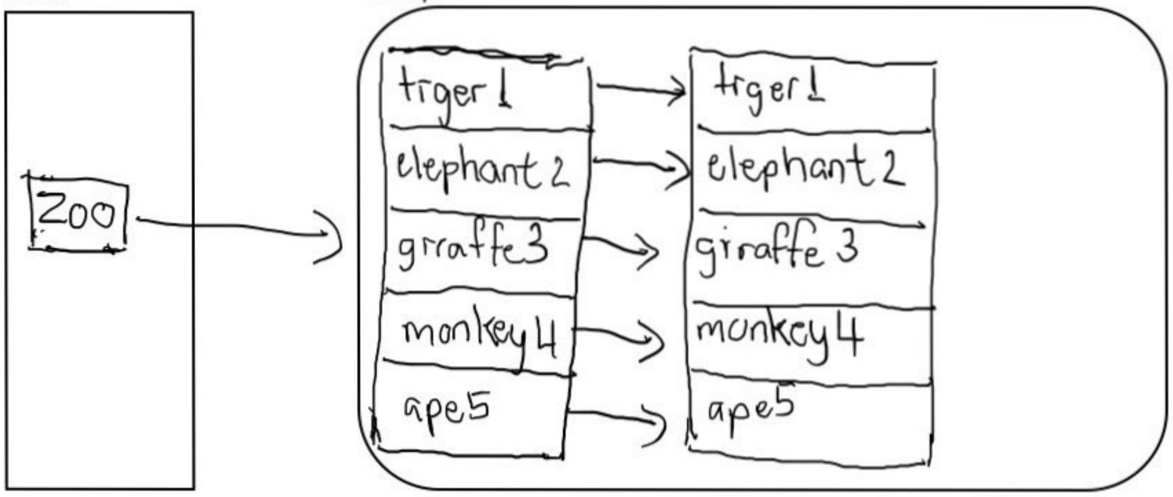
5. a. Write the deep copy of the following in code.

```
Animal[] zoo = {tiger1, elephant2, giraffe3, monkey4, ape5};
Animal[] copy;
```

```
Animal [] copy = (Animal[])zoo.clone();
```

b. Draw the memory map.

Stack: Heap:



6. What is garbage collection? Where does it happen?

Garbage Collector is a program that manages memory automatically by de-allocation of objects is handled by Java rather than the programmer. Garbage Collection runs on the heap memory to free the memory used by objects that don't have any reference. Any object created in the heap space has global access and can be referenced from anywhere of the application.

In Java, Garbage collection happens automatically during the lifetime of the program, eliminating the need to de-allocate memory, which avoids memory leaks. When an object is created, it uses some memory, and the memory remains allocated until there are references for the use of the object. When there are no references to an object, it is assumed to be no longer needed, and the memory occupied by the object can be recover. This process is known as Garbage Collection.There is no explicit need to destroy an object as Java handles the de-allocation automatically. Without de-allocating memory, programs may crash when there is no memory left in the system to allocate. These programs are said to have memory leaks.

A concurrent garbage collection runs in the background, cleaning up while the program is running. Some garbage collection might run as part of every memory allocation. An incremental collector might do that by scanning a few objects at every memory allocation.

7. What is the difference between the two operators, equals() and ==?

The difference of equals() and == in java is to the extend of primitives while equals() is recommended to check equality of objects. If both == and equals() is used to compare objects than equals() return true and false based on its overridden implementation, whereas == returns true only when both references point to the same object. The most common example in Java can be demonstrated by comparing two string in Java in which the results of the use of these two operators are different.