Array of Thing objects

Arrays

- Arrays are static
  - Cannot grow or shrink
- Can contain primitive variables
  - int[] numbers = new int[5];
  - Array with 5 integers;
- or references to objects
  - MyClass[] container = new MyClass[5];
  - Array with 5 references to MyClass objects
  - Do not place objects in array, but references that can point to objects
- Index of first element is 0
- Index of last element is length – 1
  - An array has an instance variable length

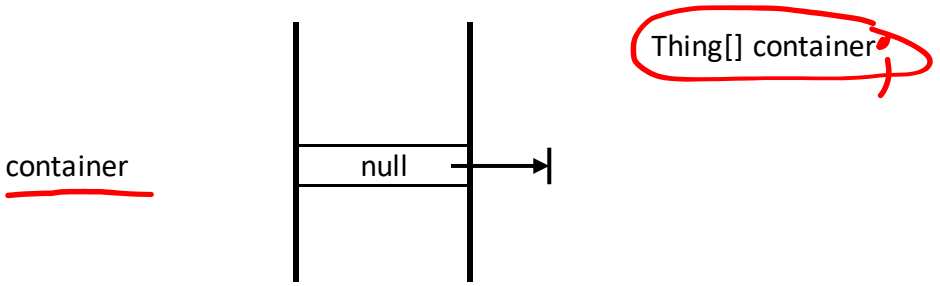
MyClass()

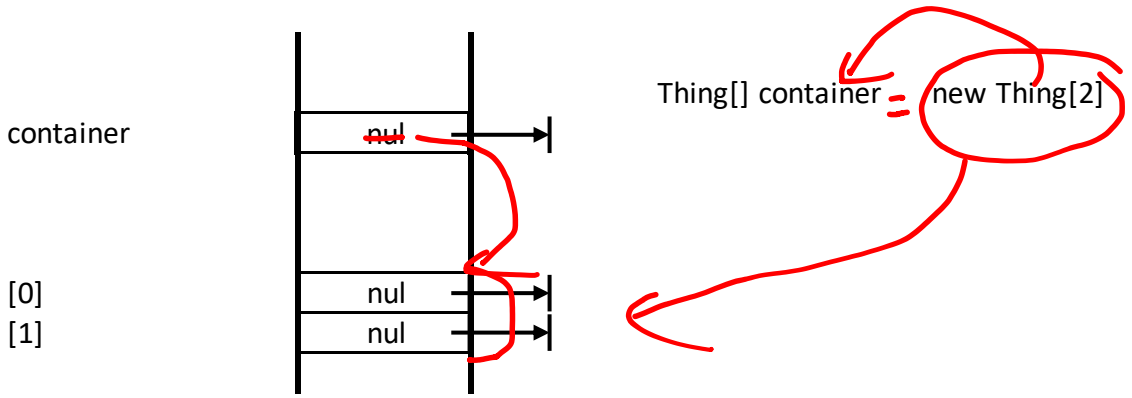
Arrays and loops

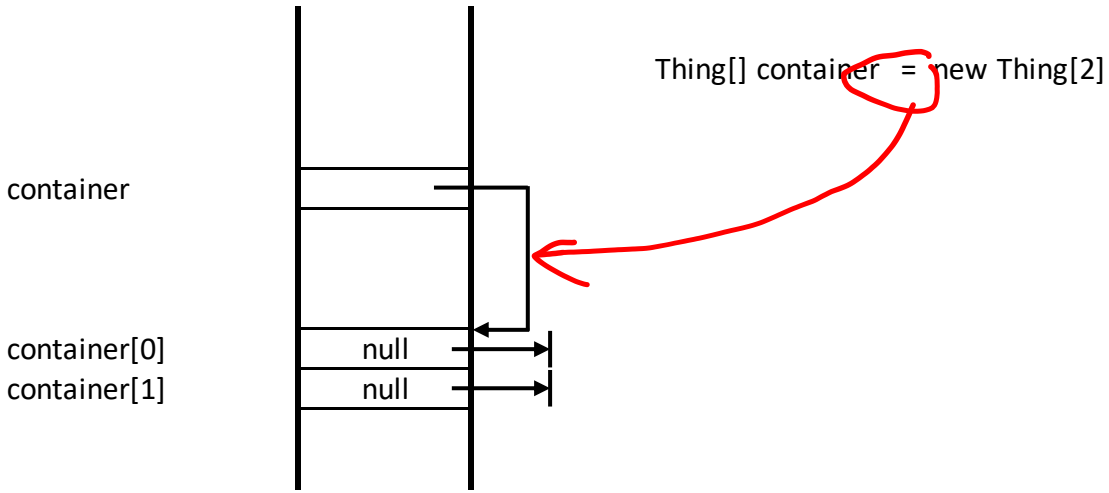
- Counter controlled for:
  - for (int i = 0; i < myArray.length; i++)
    System.out.println(myArray[i]);
- Also enhanced for:
  - Syntax of enhanced for:
    - for (parameter : arrayName)
      statement
    - parameter has type and identifier
    - arrayName is the array through which we are iterating
  - for (int num : myArray)
    System.out.println(num);
  - Can only get values of array elements, cannot modify them!
    - for (int num : myArray)
      num = num * 2;

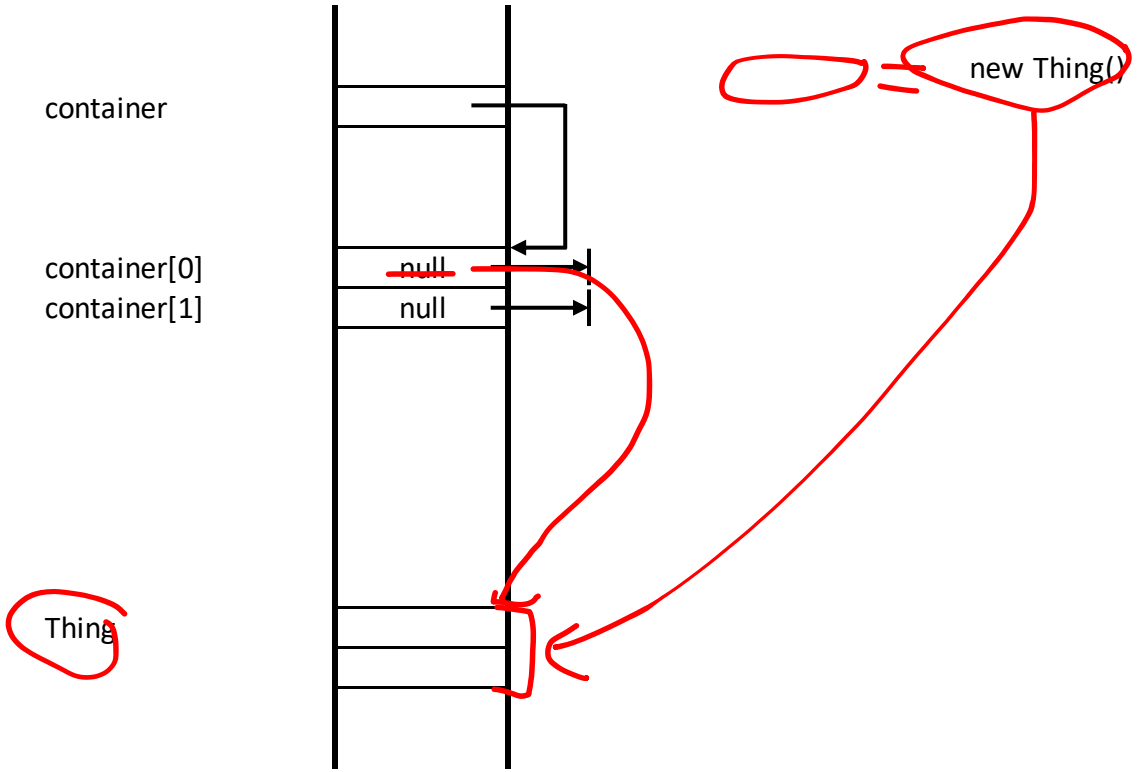
Arrays of objects

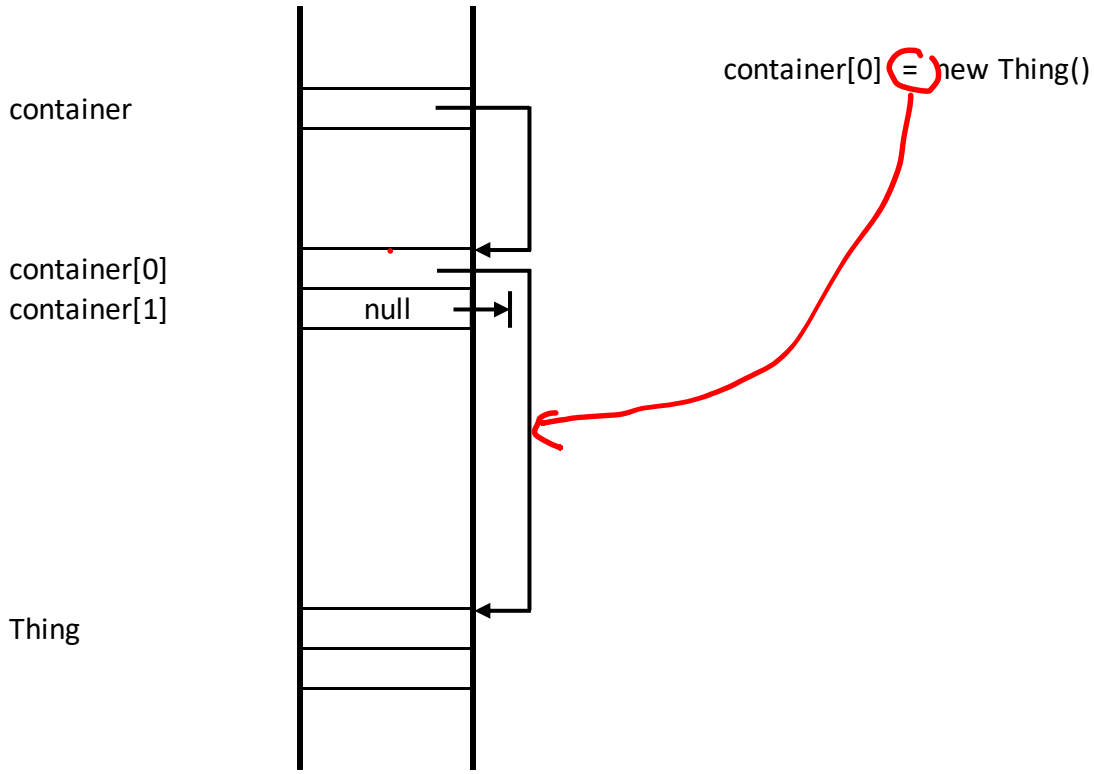
- Creating an array:
  - Thing[] container
    - Reference variable
  - new Thing[n]
    - Array of n references
  - Thing[] container = new Thing[n];
    - Assign address of array to the reference container
- Placing object into array:
  - new Thing()
    - Create new object (can also use existing object)
  - container[i] = new Thing();
    - Assign address of new Thing object to i’th element of container

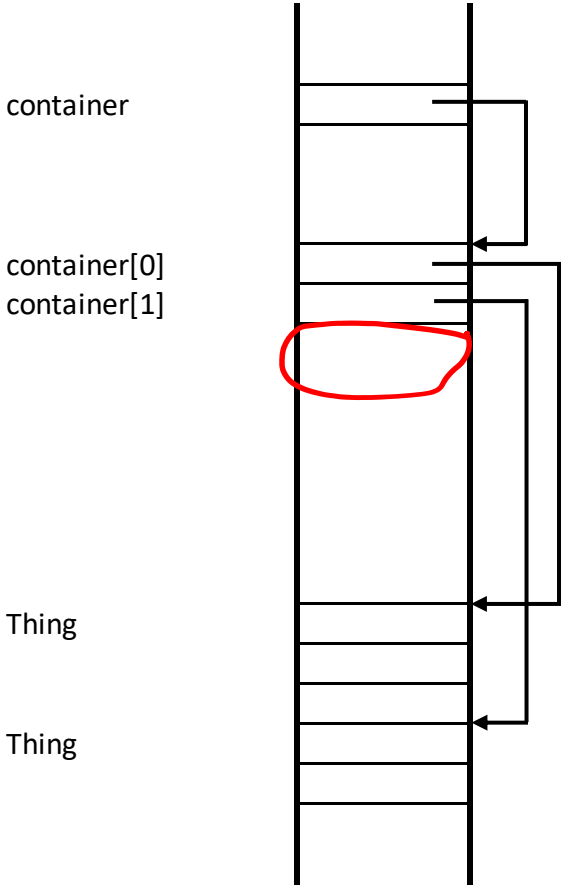

Thing[] container
container
null


container
nul
[0]
nul
[1]
nul
Thing[] container = new Thing[2]


Thing[] container = new Thing[2]
container
container[0]
null
container[1]
null


container
container[0]
container[1]
null
null
new Thing()
Thing

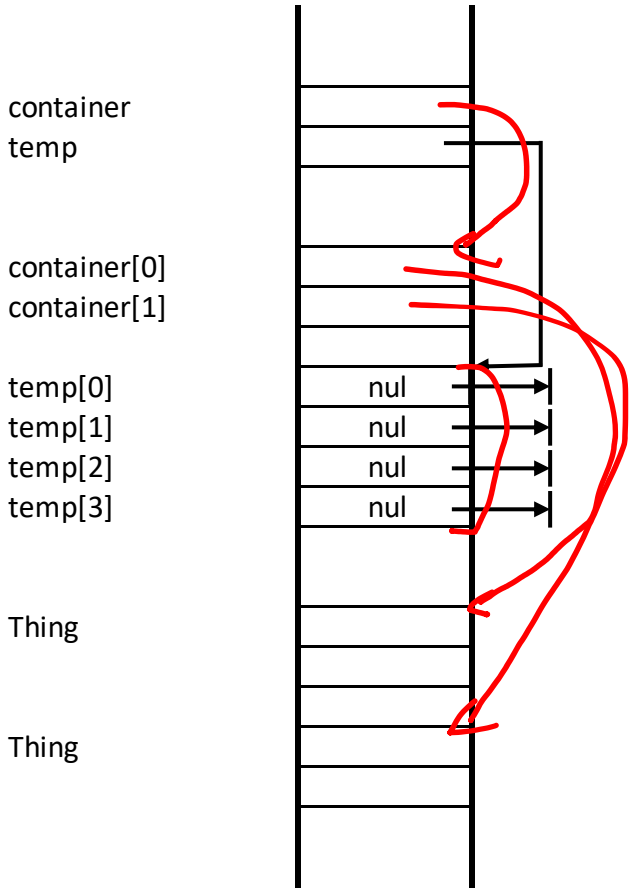

container[0] = new Thing()
container
container[0]
container[1]
null
Thing



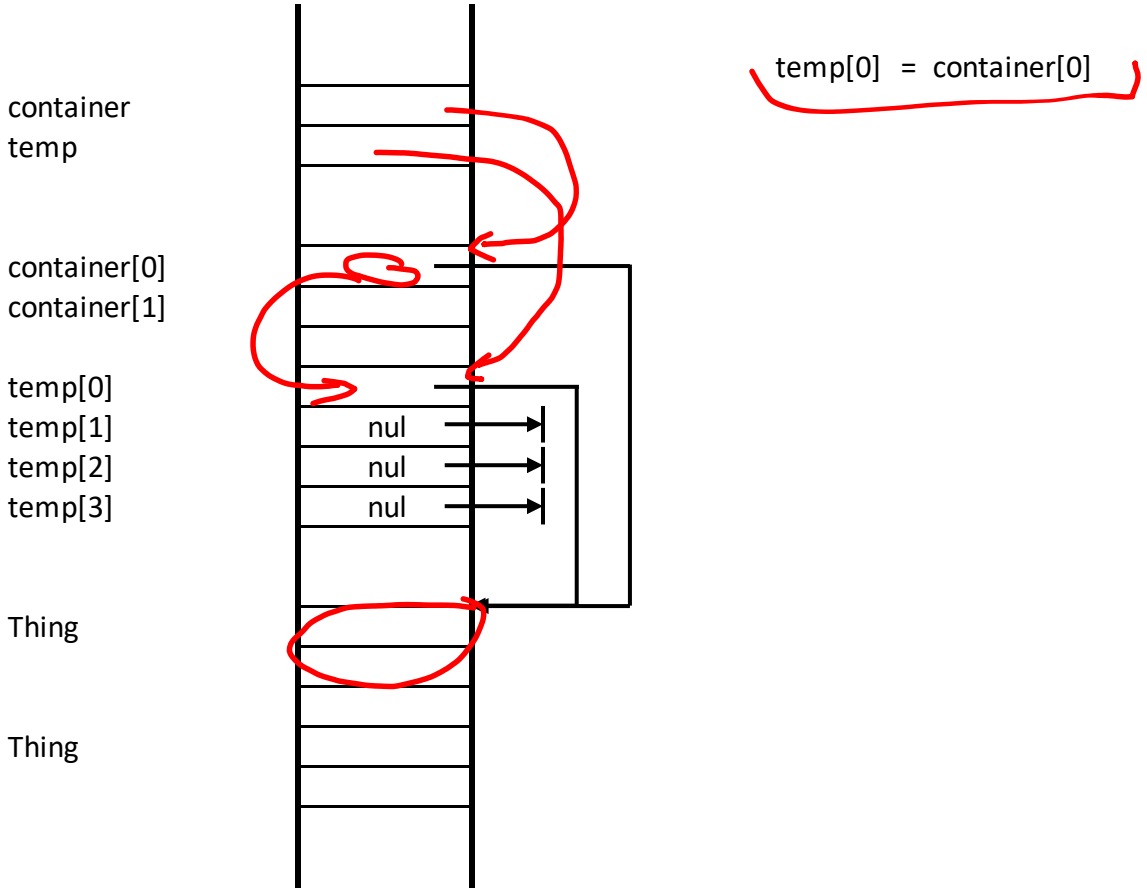
container[1] = new Thing()

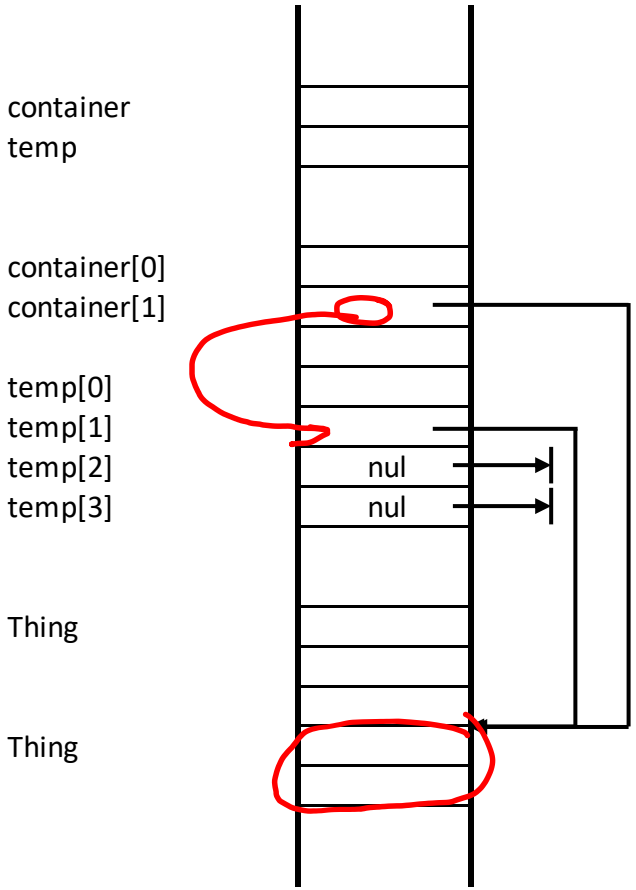
Growing (and shrinking) arrays

- Create temporary larger (or smaller) array
- Assign values of original array to temporary array
- Assign temporary array's reference (address) to original array's reference
- Both original and temporary arrays' references point to new array
- Nothing points to original array
  - Original array is "orphaned"
  - Garbage collector will make that memory available for use



Thing[] temp = new Thing[4]


container
temp
container[0]
container[1]
temp[0]
temp[1]
temp[2]
temp[3]
nul
nul
nul
Thing
Thing
temp[0] = container[0]



For(i:0 - - -

temp[i] = container[i]

container = temp