RETURNING ARRAYS

zyBook 7.7

@ Dr. Jessica Young Schmidt and NCSU Computer Science Faculty

RETURNING ARRAYS

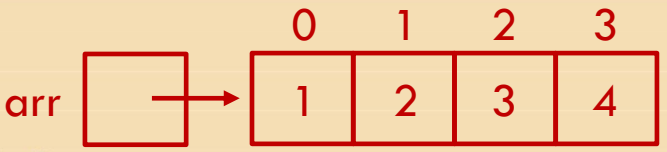
The return type for a method can be an array. Returning an array typically occurs when a new array is created within a method rather than modifying an array parameter.

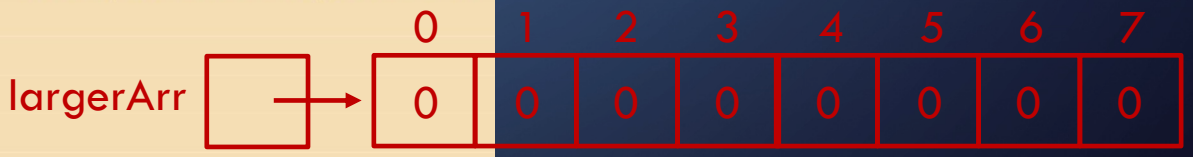
```
import java.util.Arrays;

public class ExpandArray {

    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4 };
        System.out.println("Before double size: " + Arrays.toString(arr));
        arr = doubleSize(arr);
        System.out.println("After double size:  " + Arrays.toString(arr));
    }

    public static int[] doubleSize(int[] arr) {
        int[] largerArr = new int[arr.length * 2];
        for (int i = 0; i < arr.length; i++) {
            largerArr[i] = arr[i];
        }
        return largerArr;
    }
}
```






```
$ java -cp bin ExpandArray
Before double size: [1, 2, 3, 4]
After double size:  [1, 2, 3, 4, 0, 0, 0, 0]
```

RETURNING ARRAYS

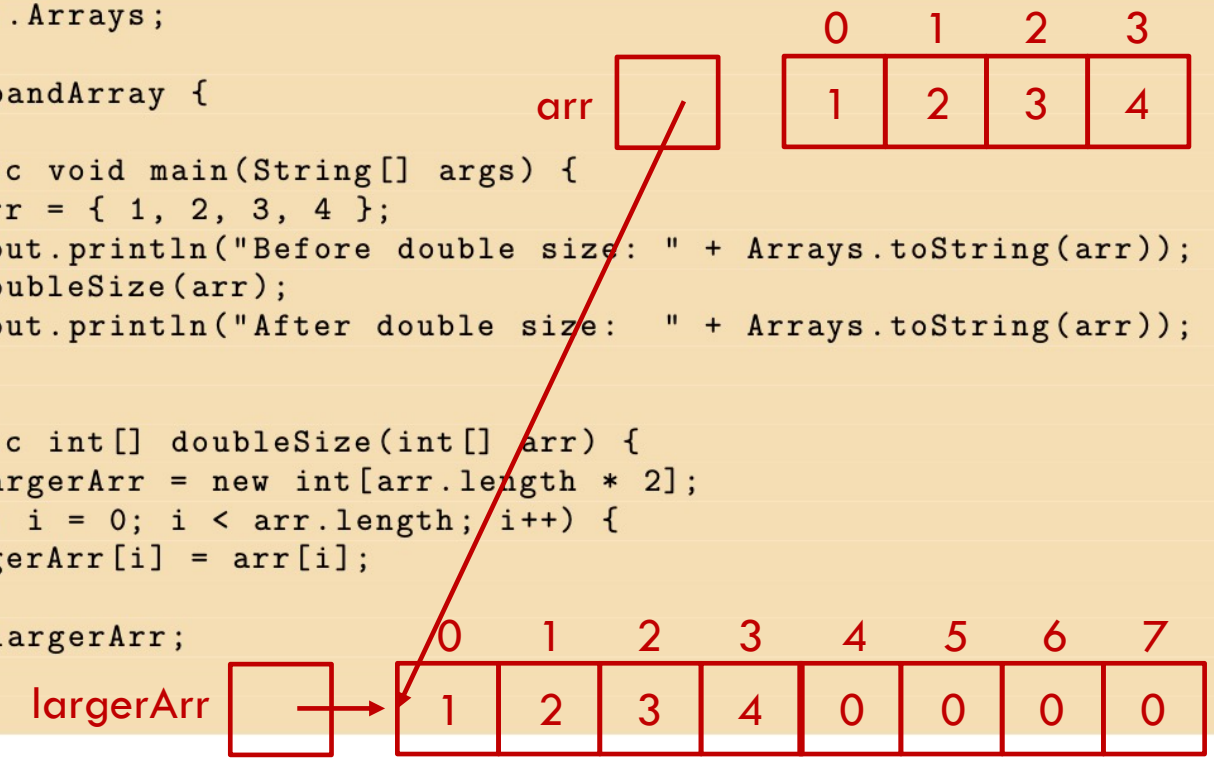
The return type for a method can be an array. Returning an array typically occurs when a new array is created within a method rather than modifying an array parameter.

```
import java.util.Arrays;

public class ExpandArray {

    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4 };
        System.out.println("Before double size: " + Arrays.toString(arr));
        arr = doubleSize(arr);
        System.out.println("After double size:  " + Arrays.toString(arr));
    }

    public static int[] doubleSize(int[] arr) {
        int[] largerArr = new int[arr.length * 2];
        for (int i = 0; i < arr.length; i++) {
            largerArr[i] = arr[i];
        }
        return largerArr;
    }
}
```




```
$ java -cp bin ExpandArray
Before double size: [1, 2, 3, 4]
After double size:  [1, 2, 3, 4, 0, 0, 0, 0]
```