

Returning Arrays

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.

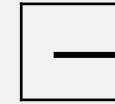
```
import java.util.Arrays;

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

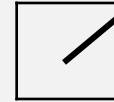
        arr = doubleSize(arr);
        System.out.println("After double size: " + Arrays.toString(arr));
    }

    public static int[] doubleSize (int[] arr) {
        arr
    }
}
```

arr



| [0] | [1] | [2] | [3] |
|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 |



Array as 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));

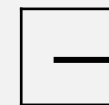
        arr = doubleSize(arr);
        System.out.println("After double size: " + Arrays.toString(arr));
    }

    public static int[] doubleSize (int[] arr) {
        arr
    }
}

```

Return type

arr



| [0] | [1] | [2] | [3] |
|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 |



```

import java.util.Arrays;

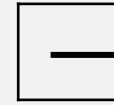
public class ExpandArray {
    public static void main (String[] args) {
        int[] arr = { 1, 2, 3, 4 };
        System.out.println("Before double size: " + Arrays.toString(arr));

        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];
    }
}

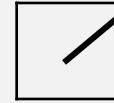
```

arr

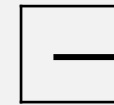


| [0] | [1] | [2] | [3] |
|-----|-----|-----|-----|
| 1 | 2 | 3 | 4 |

arr



largerArr



| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

```

import java.util.Arrays;

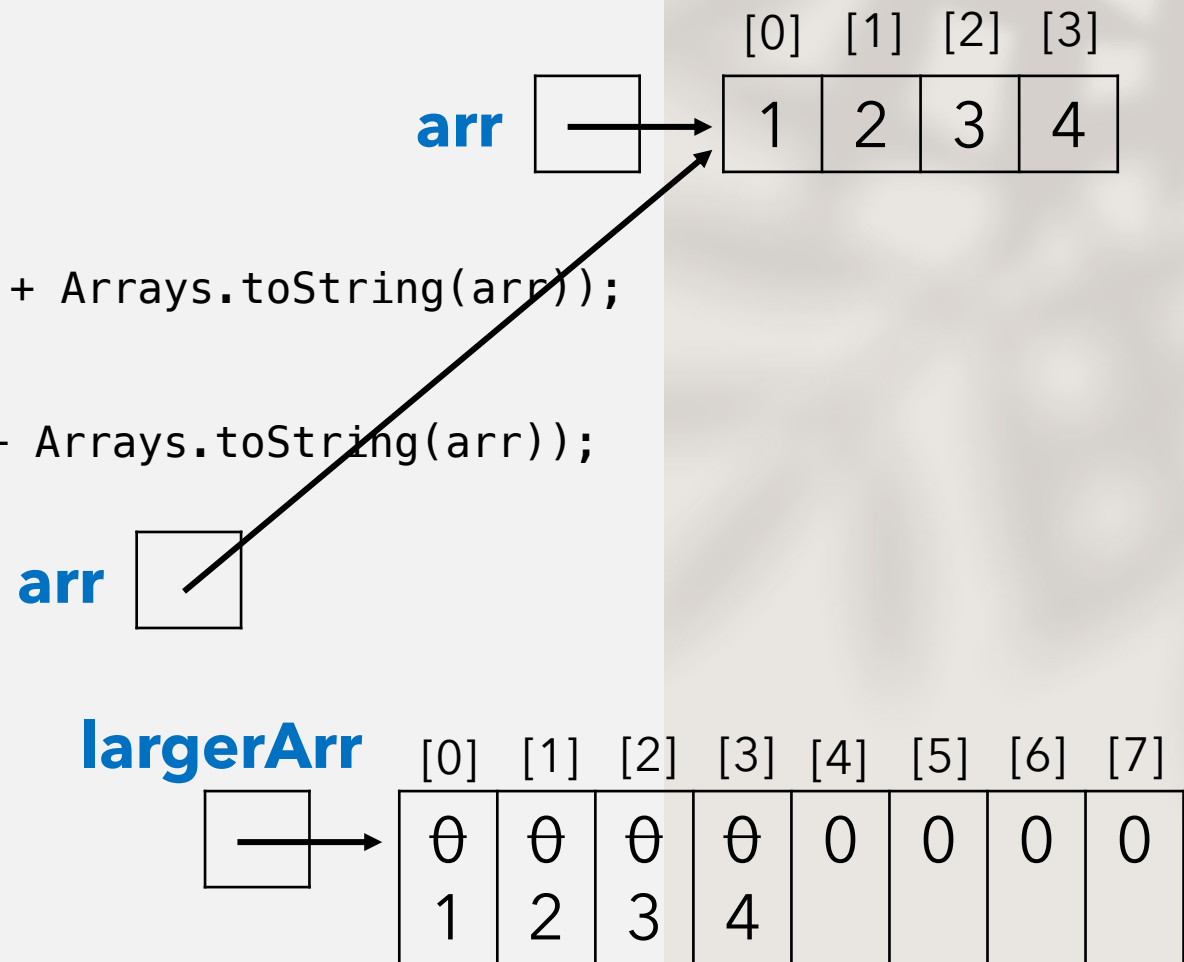
public class ExpandArray {
    public static void main (String[] args) {
        int[] arr = { 1, 2, 3, 4 };
        System.out.println("Before double size: " + Arrays.toString(arr));

        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];
        }
    }
}

```



```

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));

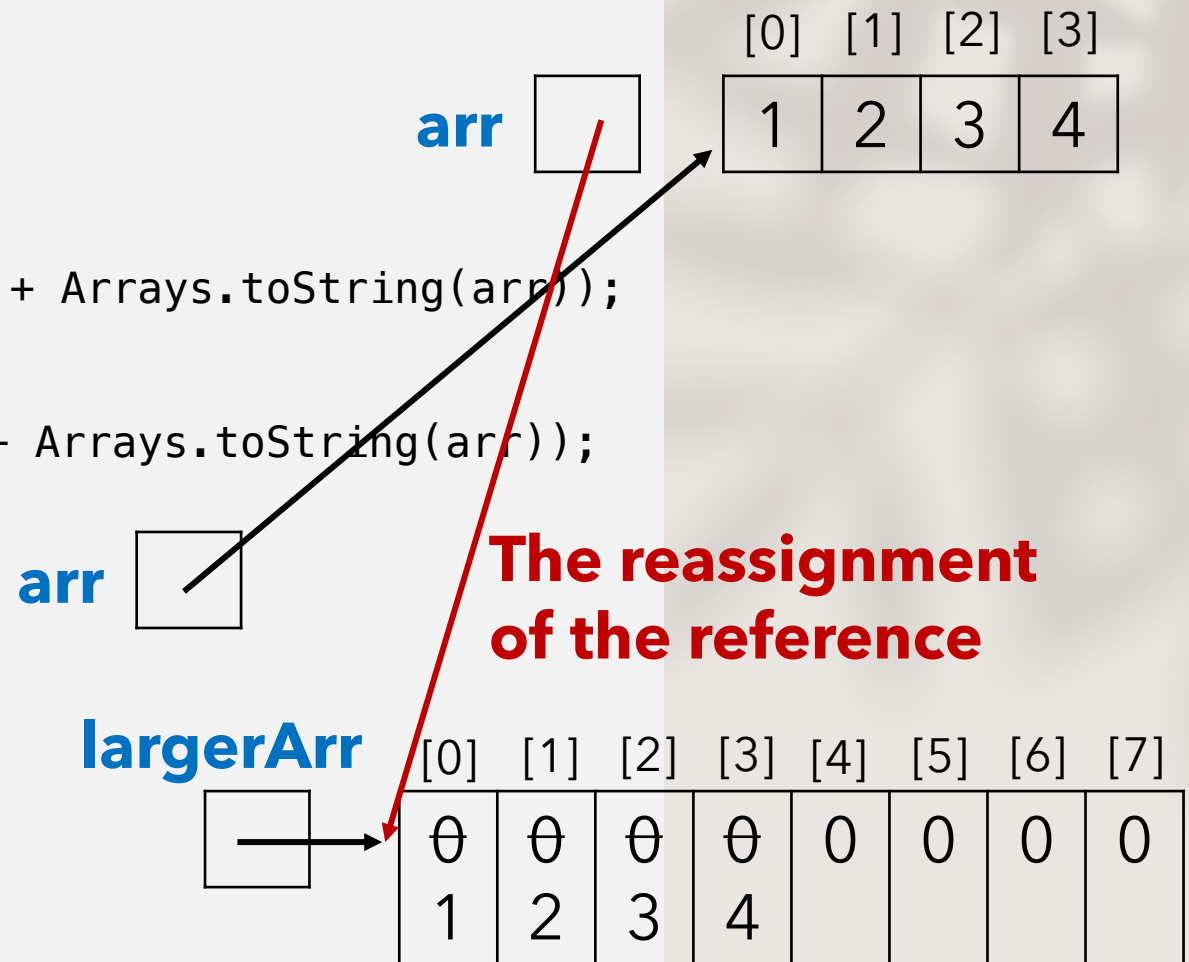
        arr = doubleSize(arr);
        System.out.println("After double size: " + Arrays.toString(arr));
    }

    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;
    }
}

```



```

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));

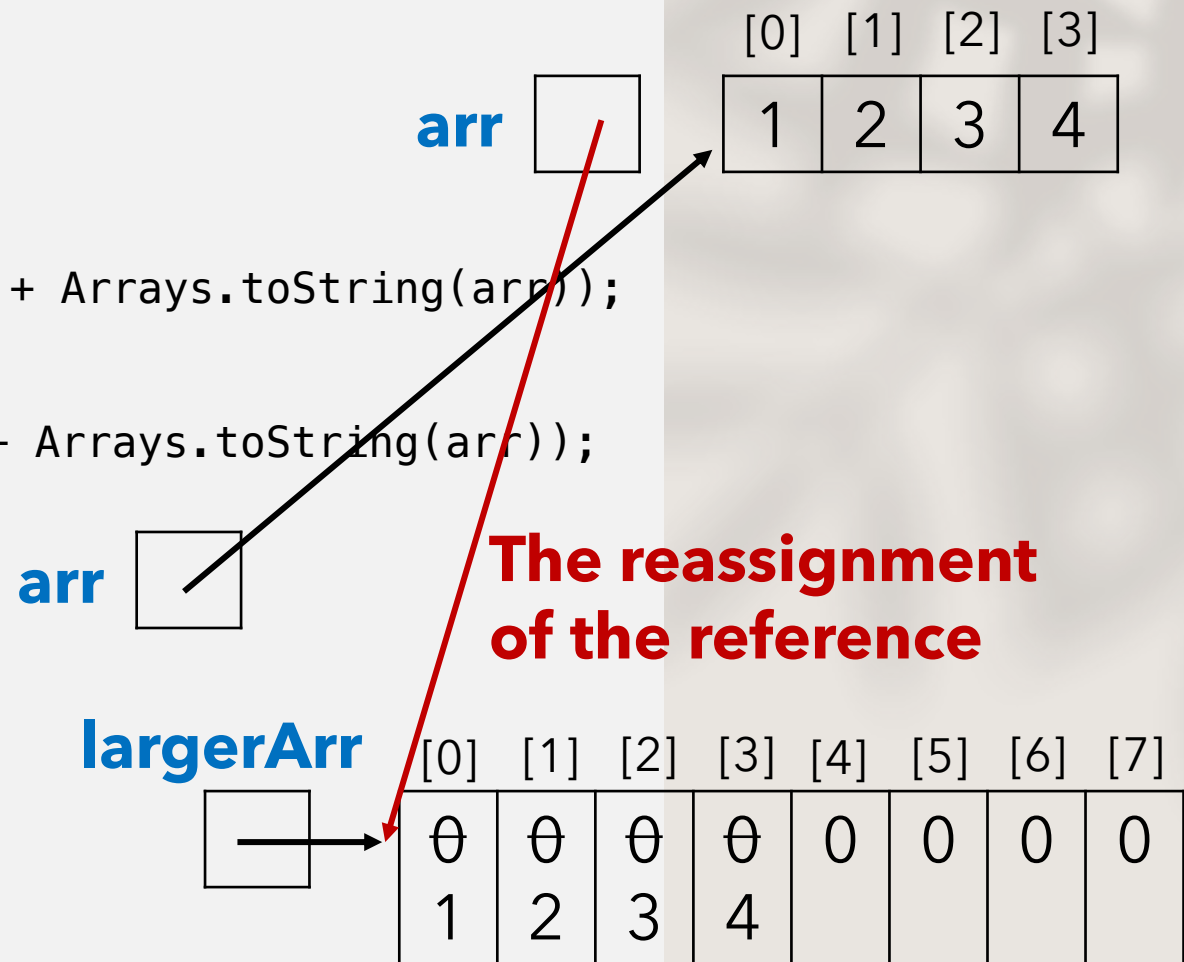
        arr = doubleSize(arr);
        System.out.println("After double size: " + Arrays.toString(arr));
    }

    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;
    }
}

```



```

$ javac ExpandArray.java
$ java ExpandArray
Before double size: [1, 2, 3, 4]
After double size: [1, 2, 3, 4, 0, 0, 0, 0]

```


TopHat Activity

Q: What does the following code segment print?

```
int[] a1 = {4, 5, 2, 12, 14, 14, 9};  
int[] a2 = a1; // refer to same array as a1  
a2[0] = 7;  
System.out.println(a1[0]);
```

Answer: 7

