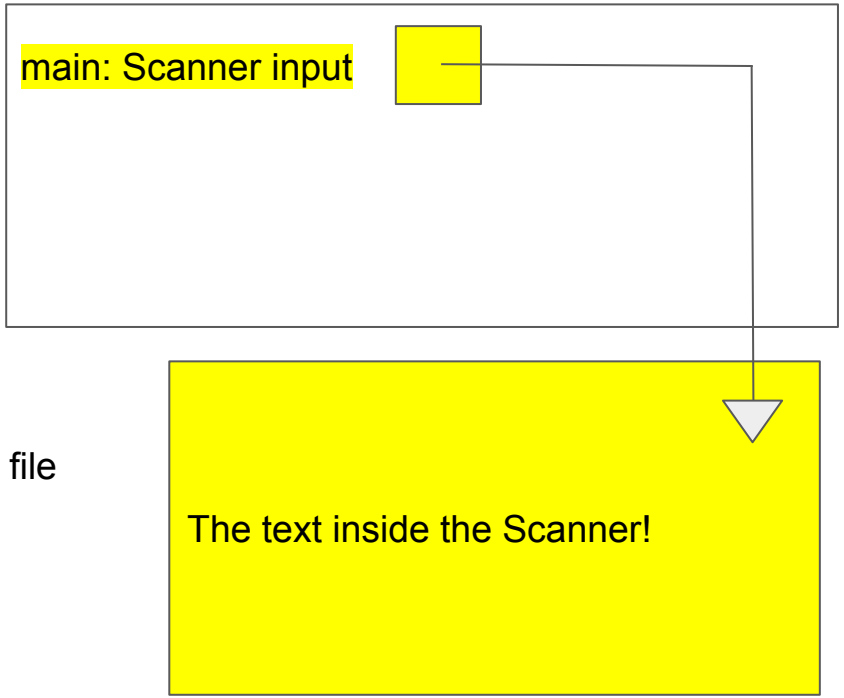
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```

The text inside the Scanner!

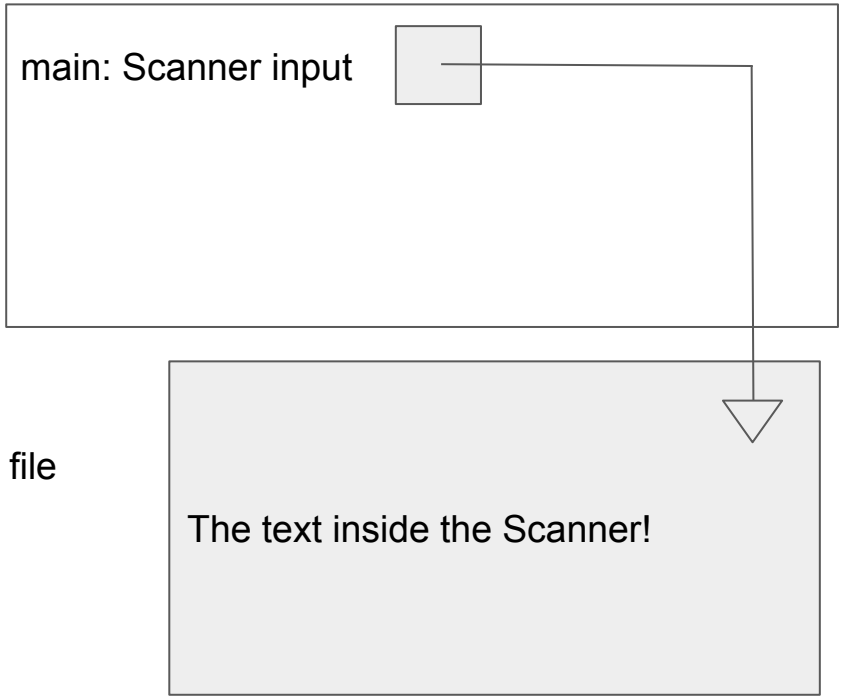
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



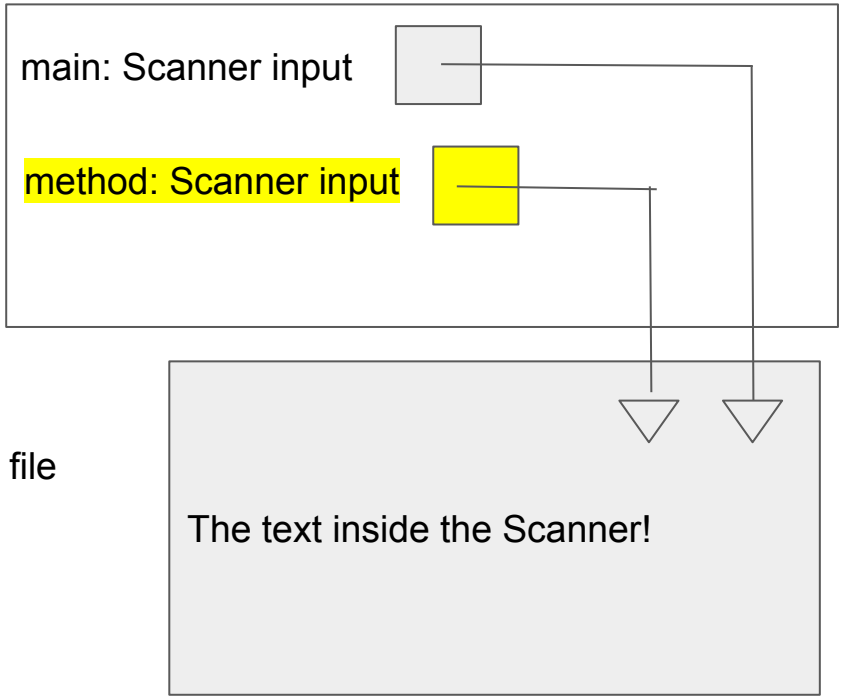
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



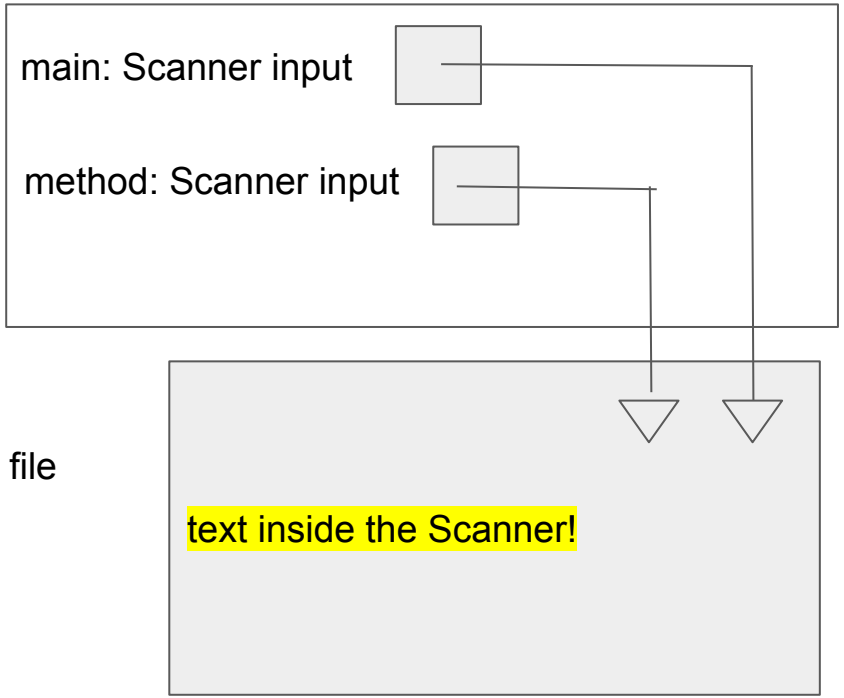
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



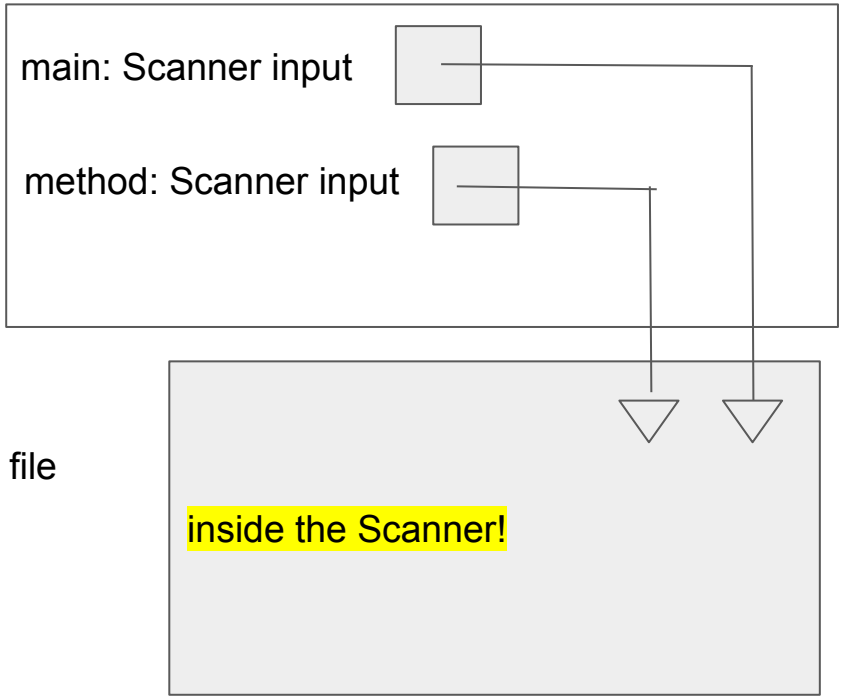
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



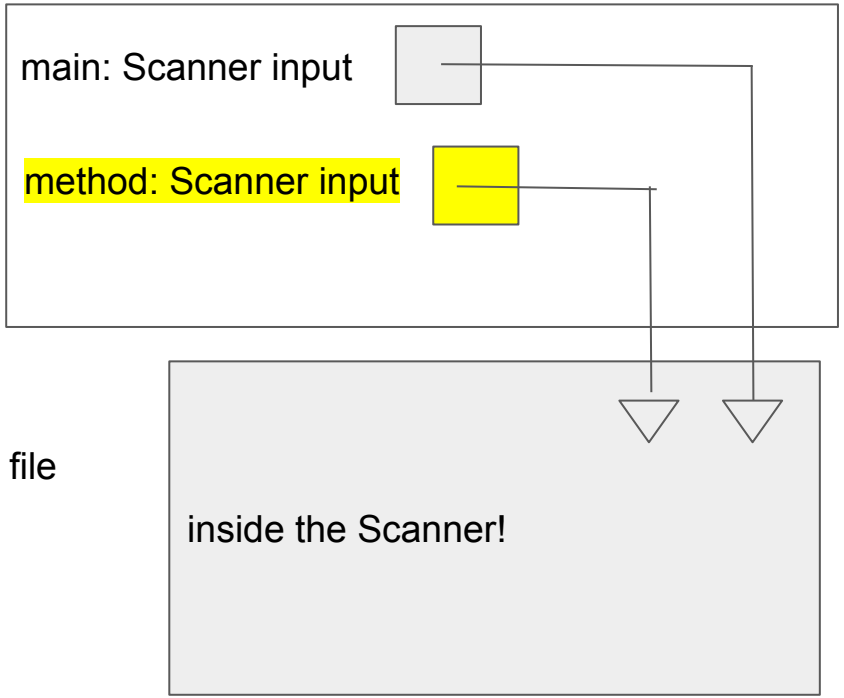
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



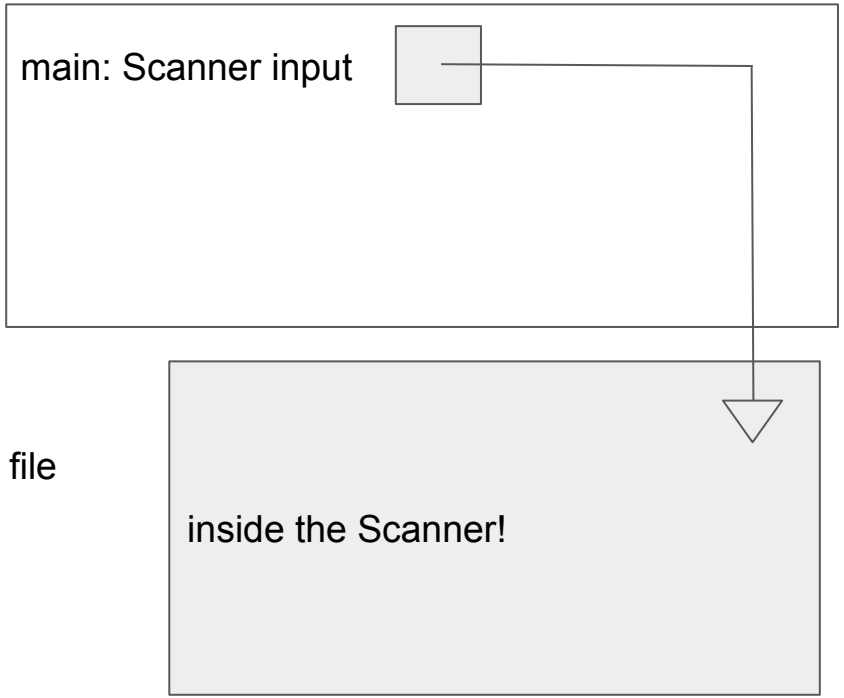
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



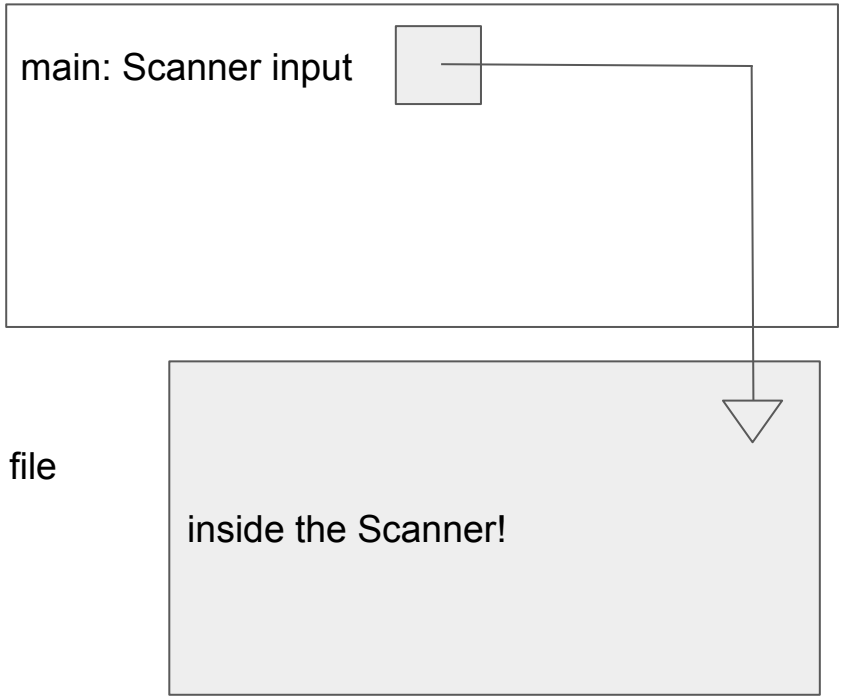
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);
        //
        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



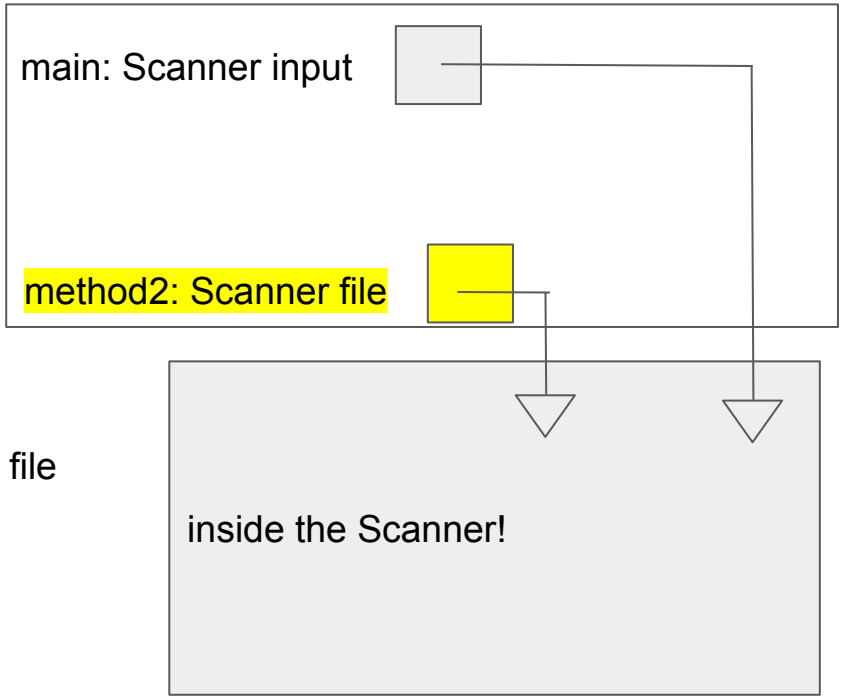
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



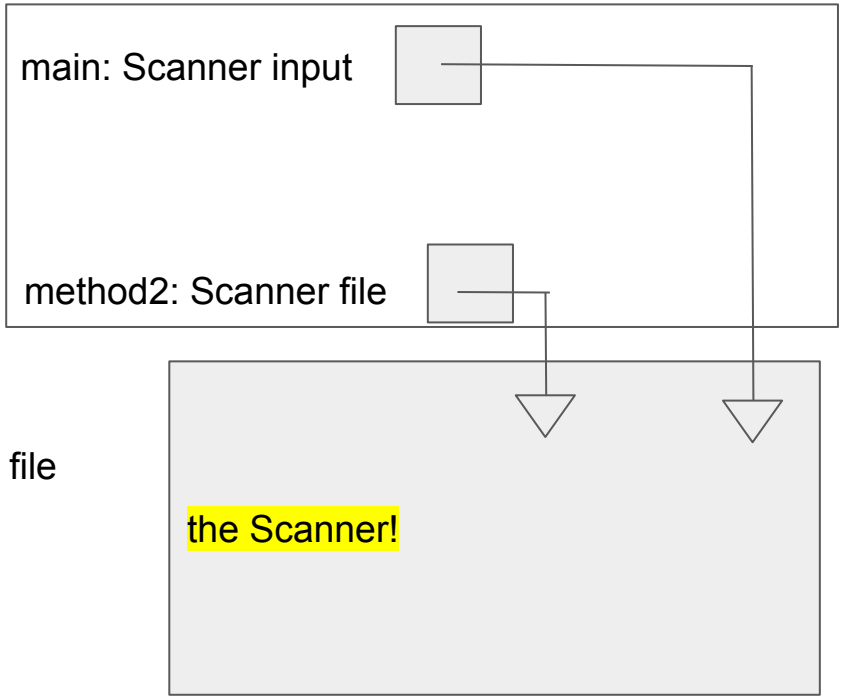
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



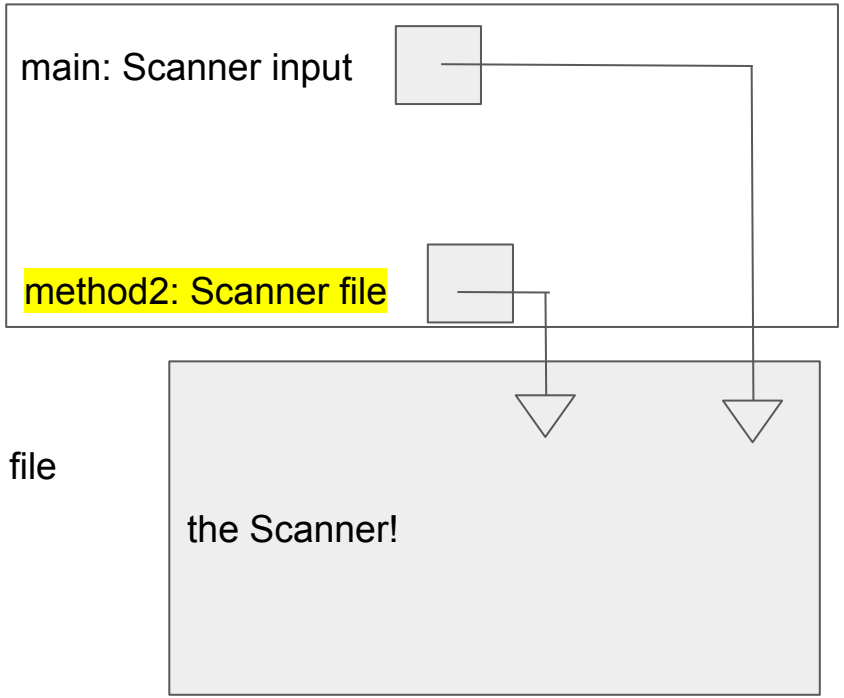
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



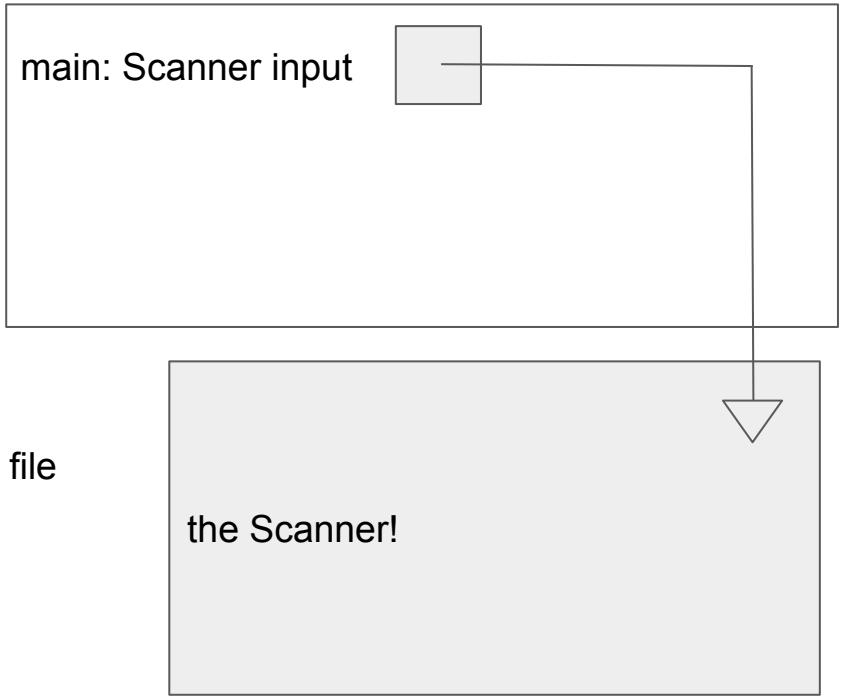
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```



Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        File input = new File("input.txt");
        Scanner input = new Scanner(input);
        method(input);

        method2(input);
}

public static void method(Scanner input) {
        // Code that uses the input Scanner to process file
        input.next();
        input.next();
}

public static void method2(Scanner file) {
        file.next();
}
```

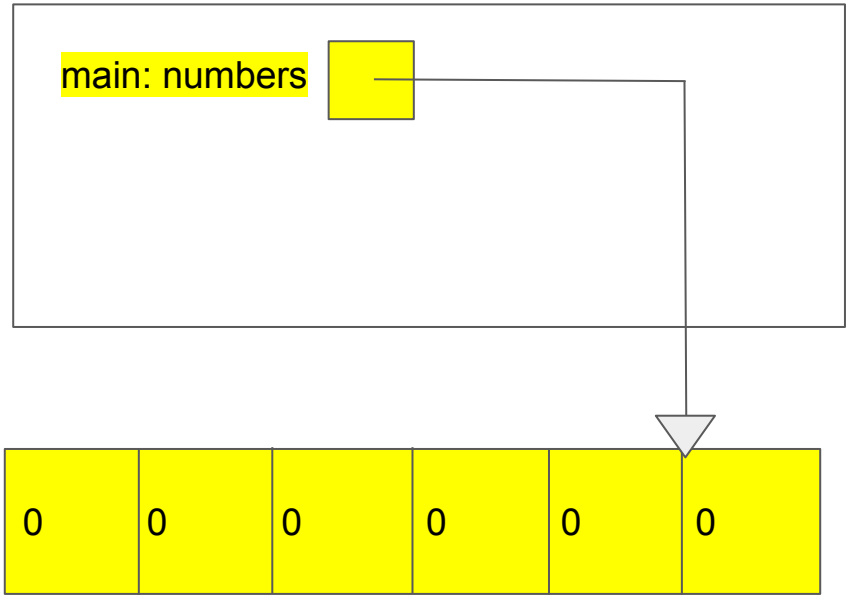


Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```

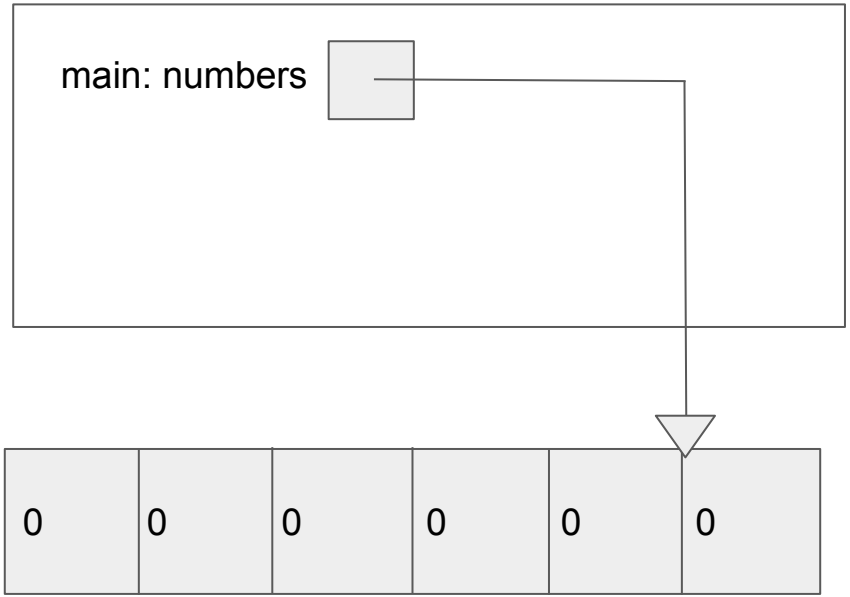
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



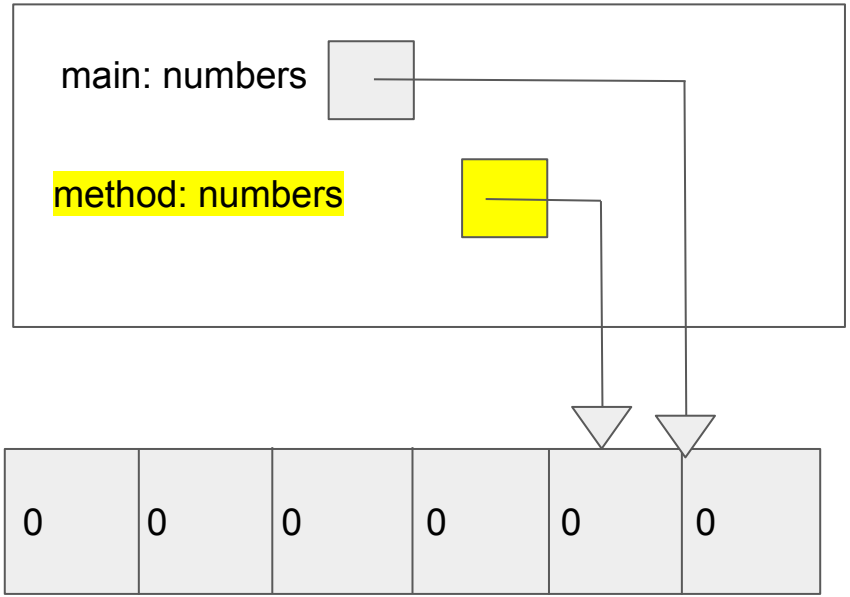
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



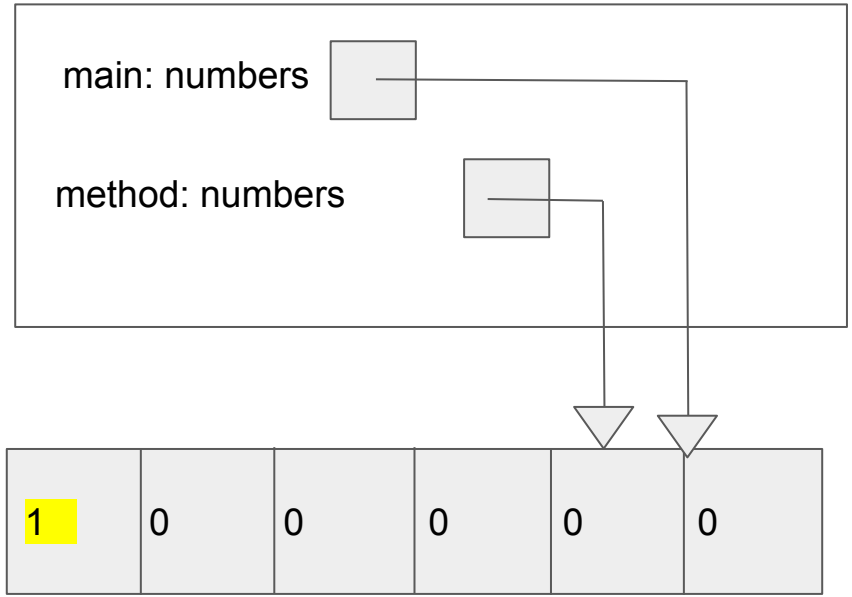
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



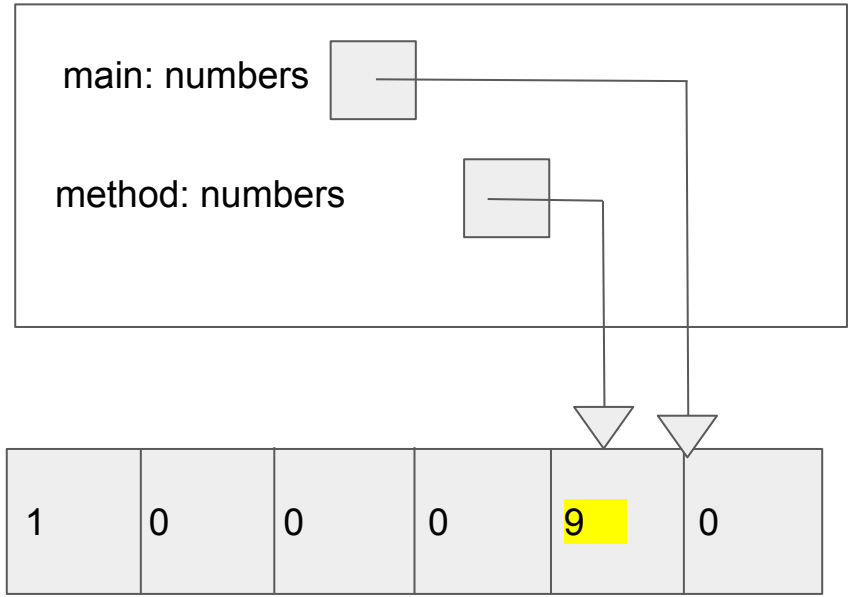
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



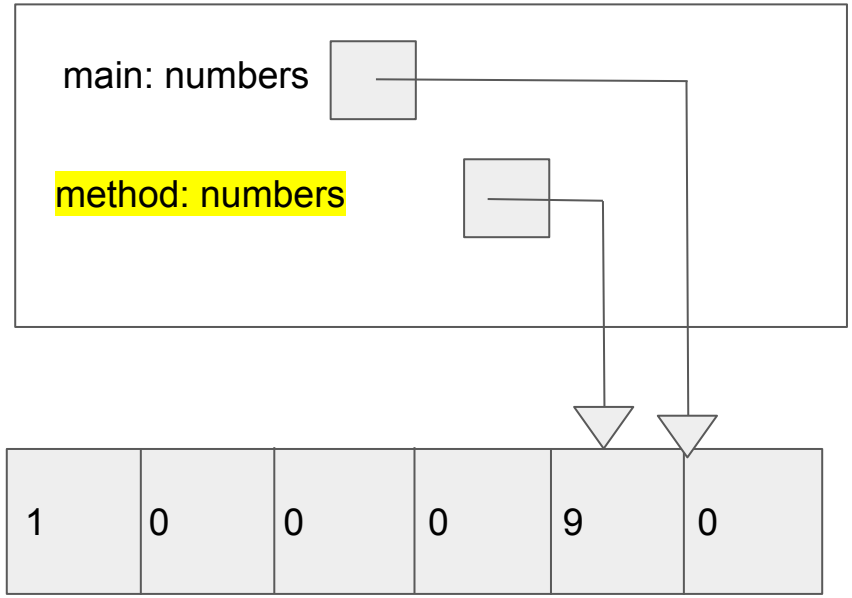
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



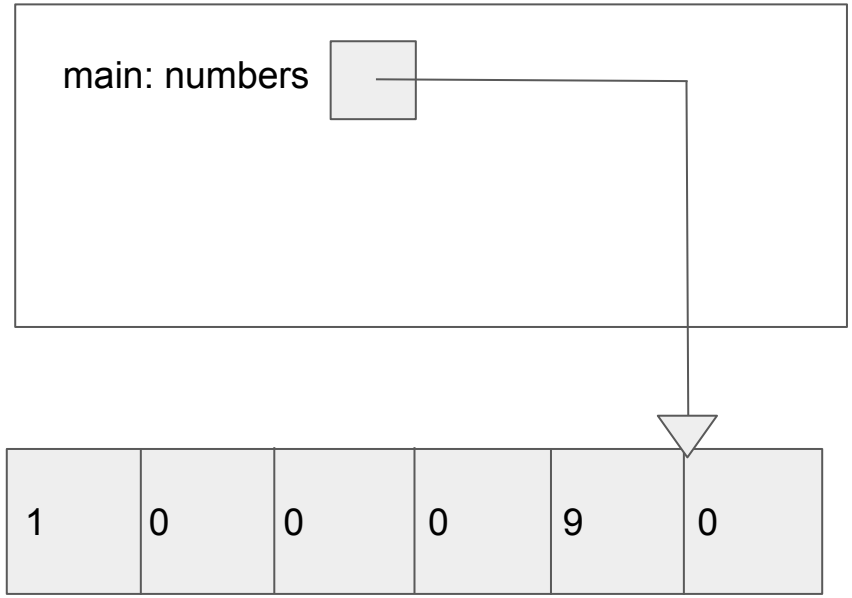
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



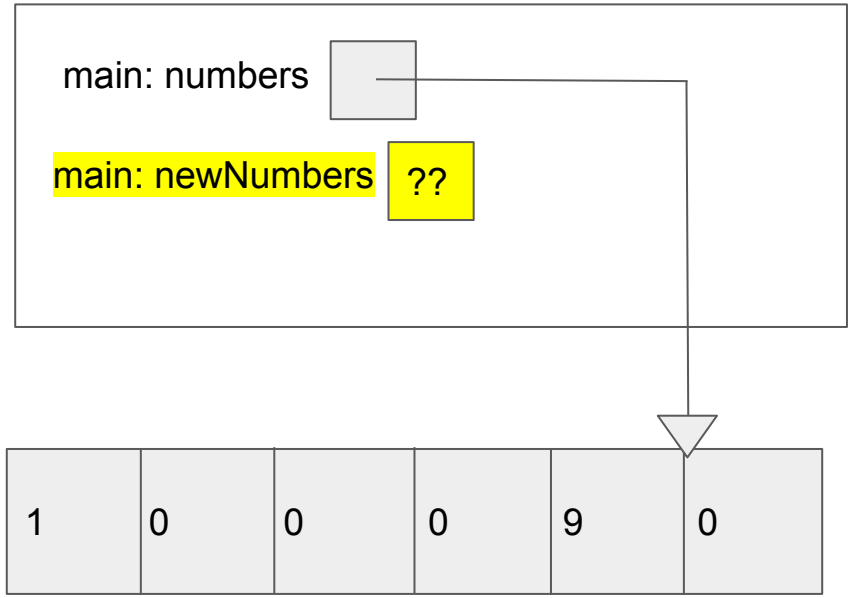
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



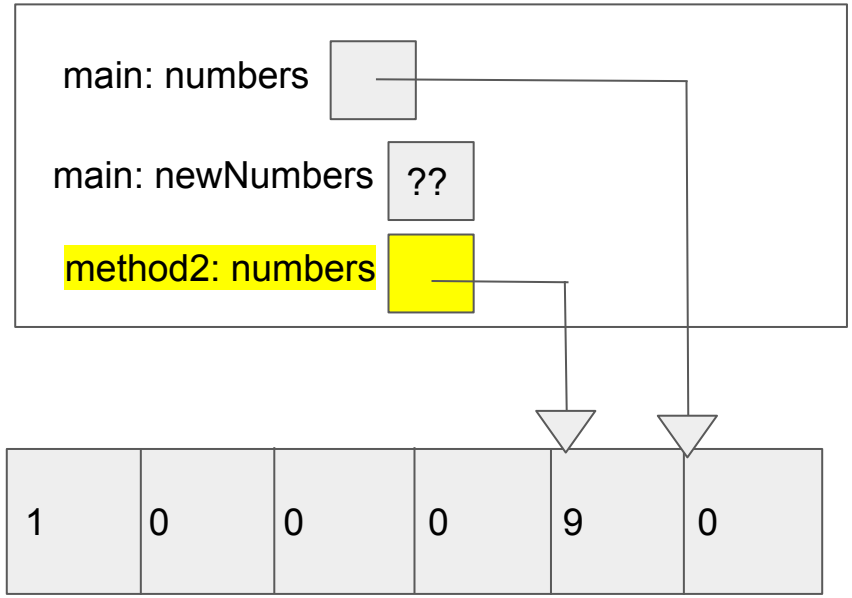
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



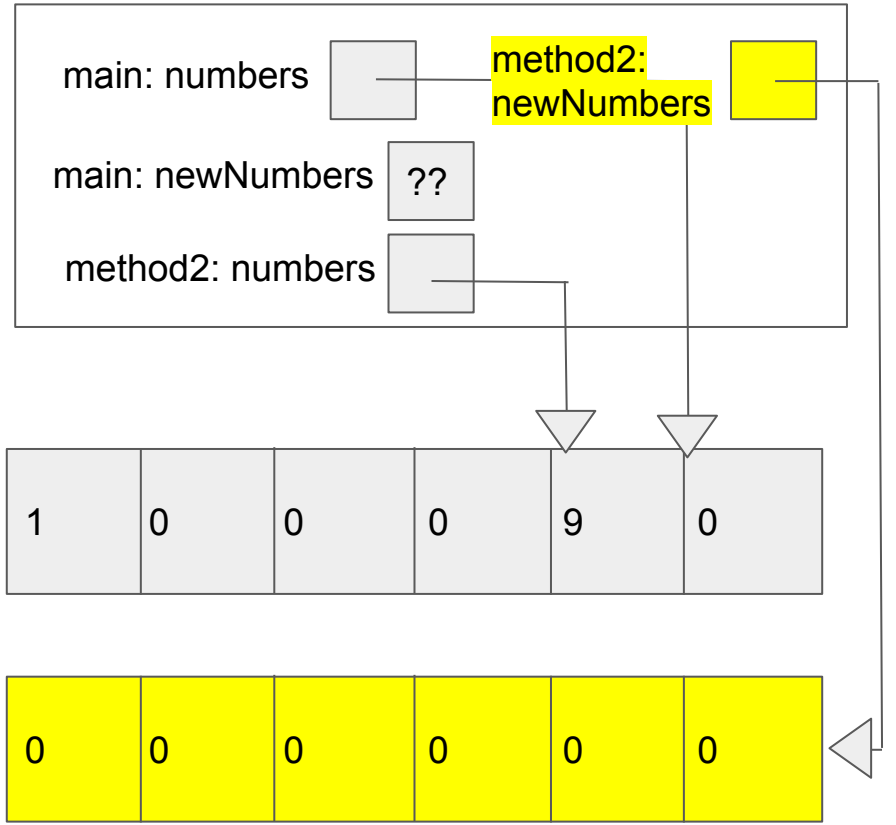
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



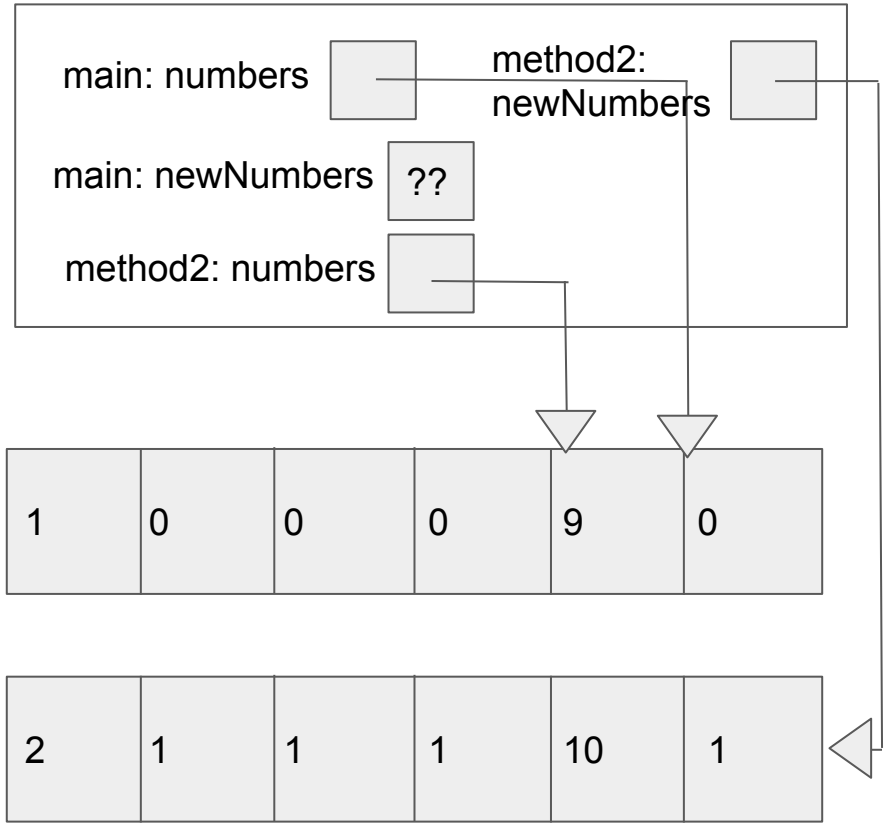
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



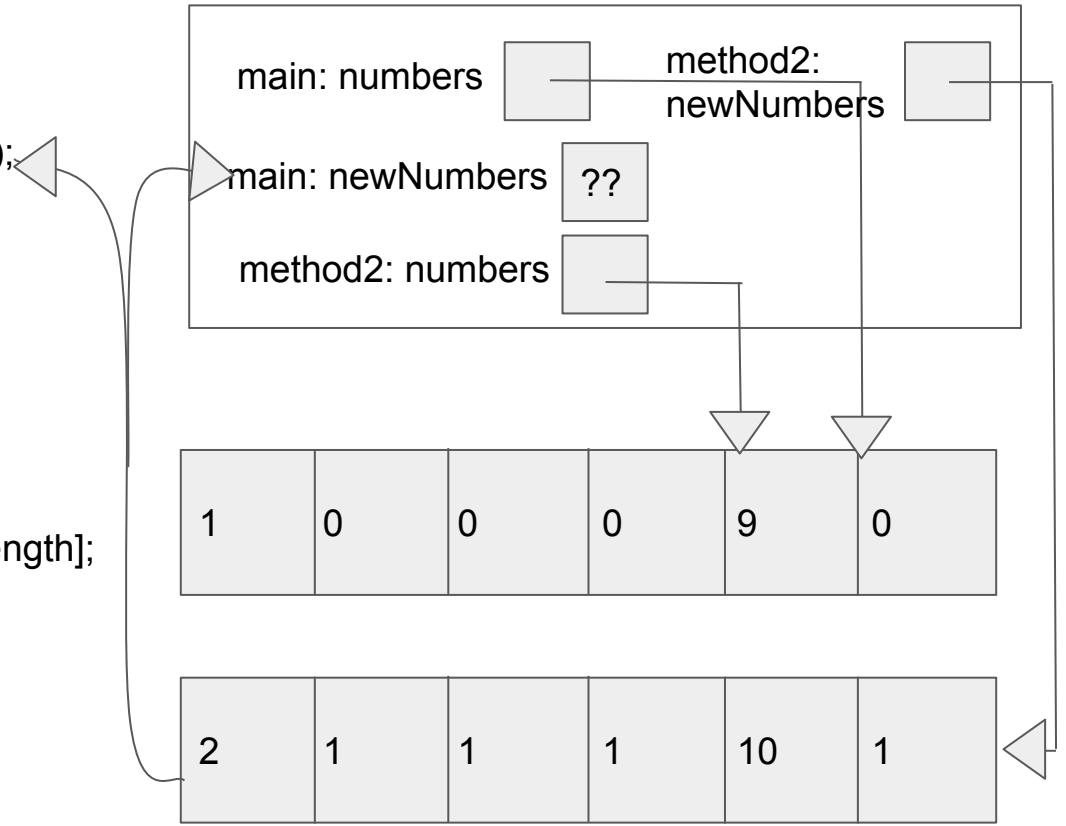
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



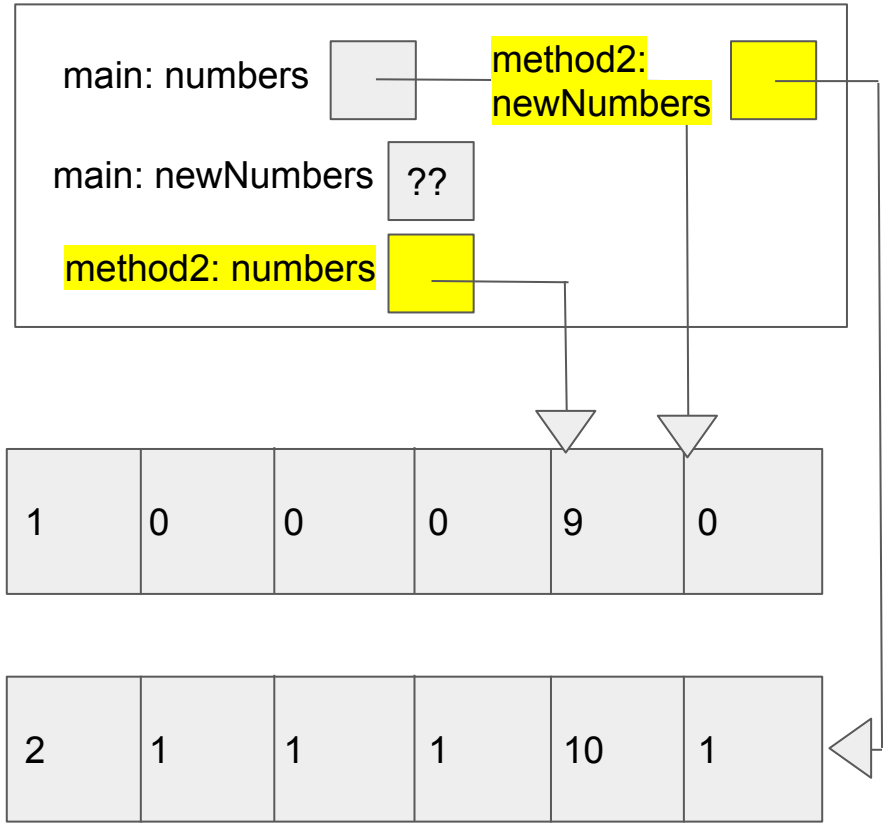
Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```



Array Reference Semantics- when not to return

```
public static void main(String[] args) {
        int[] numbers = new int[6];
        method(numbers);
        int[] newNumbers = method2(numbers);
}

public static void method(int[] numbers) {
        numbers[0] = 1;
        numbers[4] = 9;
}

public static int[] method2(int[] numbers) {
        int[] newNumbers = new int[numbers.length];
        for (int i = 0; i < numbers.length; i++) {
                newNumbers[i] = numbers[i] + 1;
        }
        return newNumbers;
}
```