

# Software and Programming I

## Lab 4: Exercises on Arrays

# Array Exercises

## Lab objectives:

- Understand how to pass and return an array as a parameter for a method.
- Use arrays to solve problems in a program.

One class (`ArrayExercises`) will be created with several methods that will be tested with calls from a main method.

**Note:** *Marked Exercise 4, the `repeatArray` method, is a marked exercise and you are required to complete it, then show and explain the finished program to one of the lab assistants, by the 20<sup>th</sup> February. Make sure you have backed up your work on a memory stick or similar.*

## Array Exercises (2)

**“An array is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed”** - The Java Tutorials, Oracle

Sample code:

```
/* declare an array of integers and allocate memory for 5 elements */
```

```
int[] array = new int[5];
```

NB: arrays correspond to lists in Python (however, the length is fixed)

## Array Exercises (3)

```
import java.util.Arrays;
```

The Java `Arrays` utility class contains various methods for manipulating arrays such as sorting and searching. It also includes the `toString()` method that returns a string representation of the contents of a specified array.

Sample code:

```
/* print a string representation of array*/
```

```
System.out.print (Arrays.toString (array) );
```

# Class ArrayExercises

## Exercise 1: main method

Create a main method in a class `ArrayExercises`.

Declare and initialise a **CONSTANT** that determines the length of the array:

```
final int SIZE = 5;
```

Declare an **array** to store 5 integers, using `SIZE` as the array length:

```
int[] array = new int[SIZE];
```

# Class ArrayExercises

## Exercise 1: main method (2)

Create code that asks a user to enter 5 integers and store them in the array. This code should only be written once.

Hint: Use a **for** loop utilising the `array.length` construct to avoid code repetition.

```
for(int i = 0; i < array.length; i++)
```

# Class ArrayExercises

## Exercise 1: main method (3) - code

```
1 import java.util.Scanner;
2 import java.util.Arrays;
3 public class ArrayExercises
4 {
5     public static void main(String[] args)
6     {
7         final int SIZE = 5;
8         int[] array = new int[SIZE];
9         Scanner scanner = new Scanner(System.in);
10        for (int i = 0; i < array.length; i++)
11        {
12            System.out.print("Please enter whole number " + (i + 1) + ": ");
13            int input = scanner.nextInt(); // get input from the user
14            array[i] = input; // store the value in the array
15        }
16        // call method sum and print out the result
17
18        // call method repeat and print out the result
19
20    }
21 }
```

# Class ArrayExercises

## Exercise 1: printArray method

Once the user has assigned all 5 values to the array, the program should print the array contents to the terminal window using the Java `Arrays` utility class as described in slide 4.

Write a helper method that takes 2 parameters, a String which describes the array, and an array, both of which should be printed to the terminal window.

```
public static void printArray(String msg, int[] array)
{
    System.out.println(msg + " " + Arrays.toString(array));
}
```

Example method call:

```
printArray("You input the following:", array);
```



# Class ArrayExercises

## Exercise 1: main method (4)

Returning to the `main` method, you will need to write calls for the methods in the following two exercises.

Use the array as an argument for the following methods, all of which should return results:

- `sum`
- `repeat`

You should print the returned result from each method before calling the next, using the `printArray` method if appropriate.

# Class ArrayExercises



## Exercise 2: sum method

Implement the method:

```
public static int sum(int[] array)
```

that returns the sum of all elements in an array:

For example, if you call it with

```
[1, 4, 9, 16, 9]
```

then it returns 39

```
(1 + 4 + 9 + 16 + 9)
```

Before writing the code, consider the following:

Will the method require new variable/s?

How can this be accomplished for arrays of any length?

# Class ArrayExercises



## Marked Exercise 4: repeat method

Implement the method:

```
public static int[] repeat(int[] array)
```

that returns an array containing the original array elements repeated 3 times.

For example, if you call it with `[1, 4, 9]`, then the method should return a *new* array containing

```
[1, 4, 9, 1, 4, 9, 1, 4, 9].
```

Hints:

1. Create a new array of size determined by `array.length`
2. Use a **for** loop(s) to copy values from one array to the other in the required order.

# Home Work

## Java for Everyone by C. Horstmann

Read Chapter 6 (Sections 6.1–6.4), which is available online from

<http://vufind.lib.bbk.ac.uk/vufind/Record/566484>

and complete the following exercises:

- Exercise R6.2
- Exercise R6.3
- Exercise R6.6
- Exercise P6.2
- Exercise P6.6
- Exercise P6.9