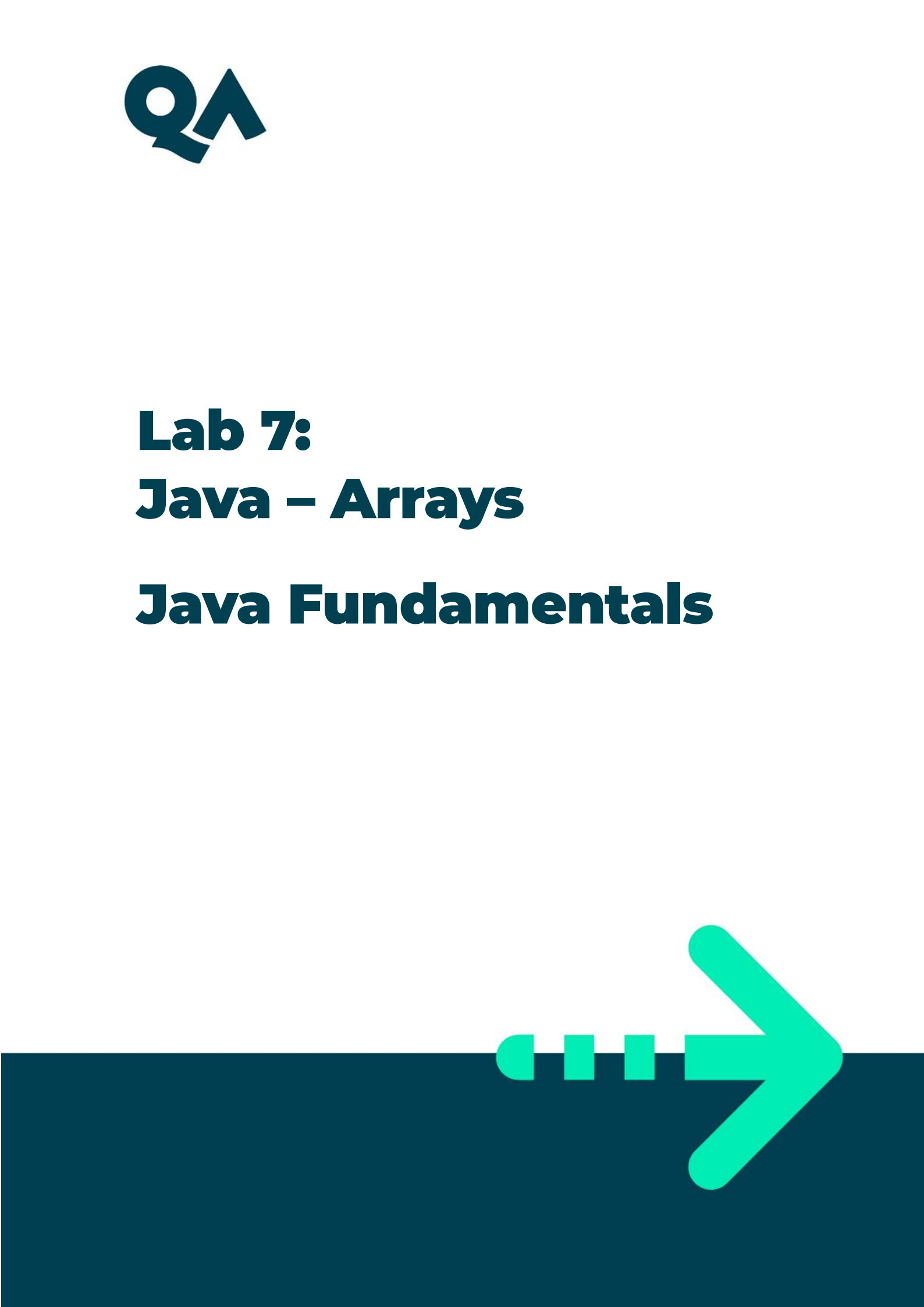

Lab 7:
Java – Arrays

Java Fundamentals

Lab 7: Java - Arrays

Objective

The objectives of this practical are to get fully up to speed with array definition, creating, filling, and manipulation and to fully understand when is it appropriate to use a 'foreach' loop as opposed to a 'for' loop to process the array.

Part 1

Step by step

1. Back in the labs project which you created in **Lab1**, add a new package called **lab07**.

   Please refer to Lab1's instructions if you need help.

2. Add a class called **Program** with a main() method to this package.
3. Add a class called Lab7.
4. Add a method called start() to Lab7 class.
5. Call the start() method from main().

   Tip: Refer to instructions in previous labs if you're not sure how to do this.

6. Declare an array of integers in the start method as:

   ```
   int[] numbers = { 1, 3, -5, 7, 0, 4, 6, 8 };
   ```

   You can put all your code in the start() method or (even better) create individual methods for each one of the following tasks and then call them from the start() method.

7. Task 1: Write code to find the sum of every number in numbers.
8. Task 2: Find the average of these numbers.
9. Task 3: Find the minimum number in numbers.
10. Task 4: Find the maximum number in numbers.
11. Task 5: Find the index of number zero in numbers.

Part 2

In this section you'll implement the bubble sort algorithm.

Step by step

1. Create a method called sort() in Lab7 class.
   This method should accept a parameter of type int[ ]
2. Call the sort method from the start() method and pass numbers as parameter.
3. Implement Bubble sort.

The Bubble Sort Algorithm is a sorting algorithm that works by comparing adjacent elements and swapping them if they are in the wrong order. For example, let's say we have an array with the elements below and we want to use the Bubble sort to put them in numerical order:

1, 10, 9, 20, 41, 12

The bubble sort would start by comparing the first two elements and, based on code that would compare the two values, would compare, and see which is bigger. If it is smaller, move to the next elements. If it is bigger, then swap the elements!

First Run: 1, 10, 9, 20, 41, 12 → 1, 9, 10, 20, 12, 41 (end of the array, so start another run!)

Second Run: 1, 9, 10, 20, 12, 41 → 1, 9, 10, 12, 20, 41

**** End ****