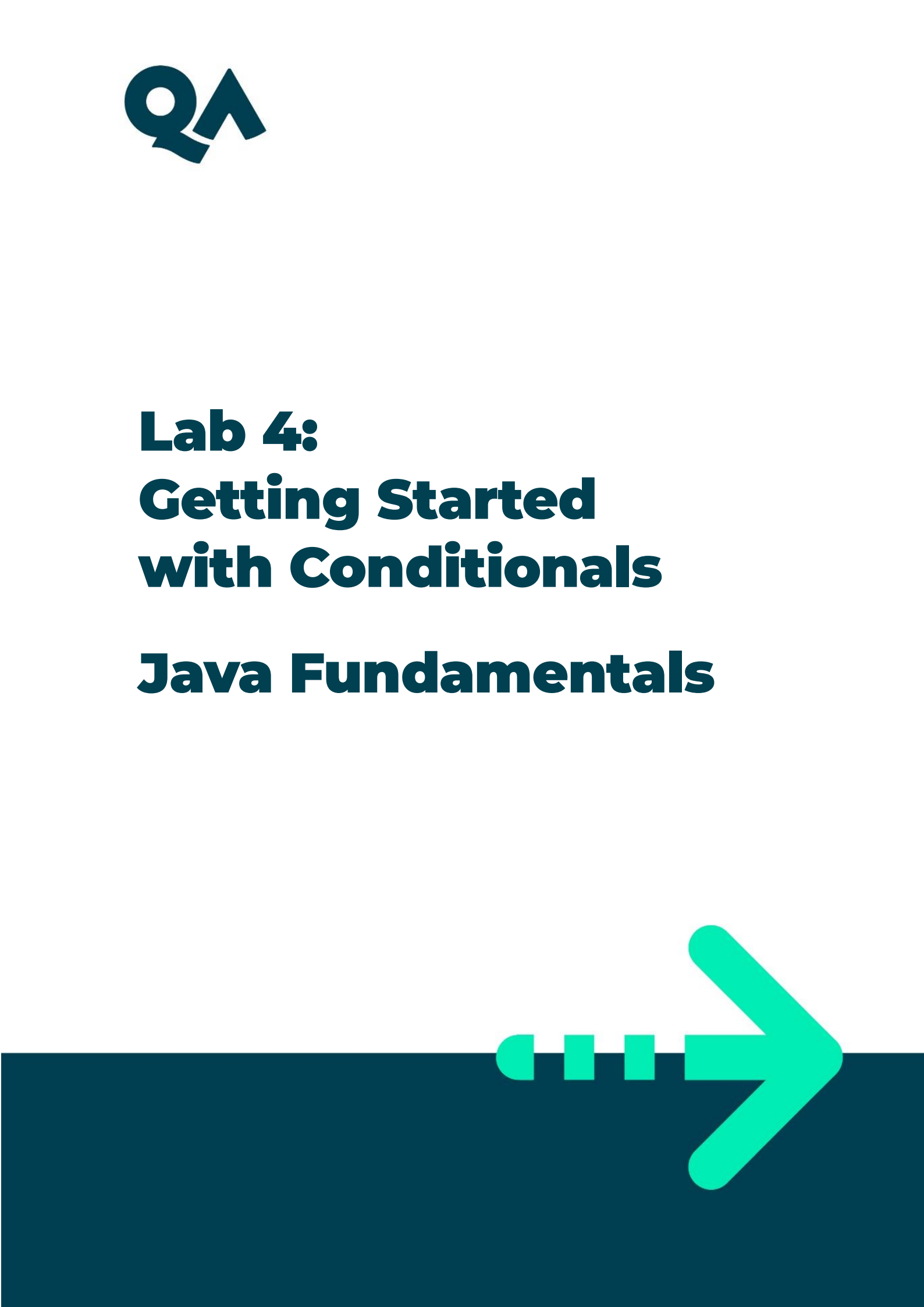

Lab 4: Getting Started with Conditionals

Java Fundamentals

Getting started with Conditionals

Objective

The objectives of this practical session are to practise using conditional expressions (if, else, and switch).

Overview

In this practical called 'Kids in a Candy Store', you'll practise using if else statements

Introduction to 'Kids in a Candy Store'

In this practical you will prompt the user for the price of a bag of sweets in pennies, count the amount of money they have (in pennies), then perform a calculation to work out how many bags of sweets they can afford. Then it will be possible to display a friendly message like:

"If the price is 'x'p and you have 'y'p, you will be able to buy 'z' bags".

Step by step

1. Back in the **labs** project, add a new package called lab04.

   Please refer to the first lab instructions if you need help.

2. Add a new class called Program to the *lab04* package with a *main()* method.

3. Add a class called **Lab4** to this package (with no main method).

4. Transfer the code for **getInt()** which you wrote in the previous lab to the Lab4 class.

5. Create a method in Lab4 as:

   ```
   public void part1() {
   }
   ```

6. Create an instance of Lab4 in the main() and call the part1() method to get ready for the rest of this exercise.

   ```
   Lab4  lab4 = new Lab4();
   lab4.part1();
   ```

   From now all your code will go in the part1() method.

7. Ask the user "Price of a bag please?" and hold the result in a variable.

8. Ask the user “How much money do you have?” and record it in a variable.
9. Calculate the number of bags they can afford and store it in a variable called numBag.
10. Now display a message in the format
“If the price is ‘x’p and you have ‘y’p, then you will be able to buy ‘z’ bags”.
11. Build and test your code using values of ‘20’ for price and ‘45’ for money. Ensure that it says ‘2’ bags can be purchased.
12. Test your code again using values of ‘20’ for price and ‘15’ for money. Ensure that it says ‘0’ bags can be purchased.

    The message looks odd and will look even odder if the user enters negative values. It will also crash the program if the user enters zero as the price of a bag. We need to bulletproof your code.
13. If the price of a bag is a negative number, display a suitable message and return from the part1() method.
14. Similarly, if the user has zero or a negative amount of money, then you should display a suitable message and return.
15. Is there any other way you can protect this application from users?
16. You should now test your code using the following test values:

Price	**Money**	**Outcome**	**Error Message**
-10	n/a		“Price must be…”
0	n/a		“Price must be…”
10	-5		“Money must be…”
10	0		“Come back with your pocket money”
10	30	…can afford 3 bags…	

17. You should now test your code using the above test script.
18. You have now avoided any divide by zero exceptions being thrown.

**** End ****