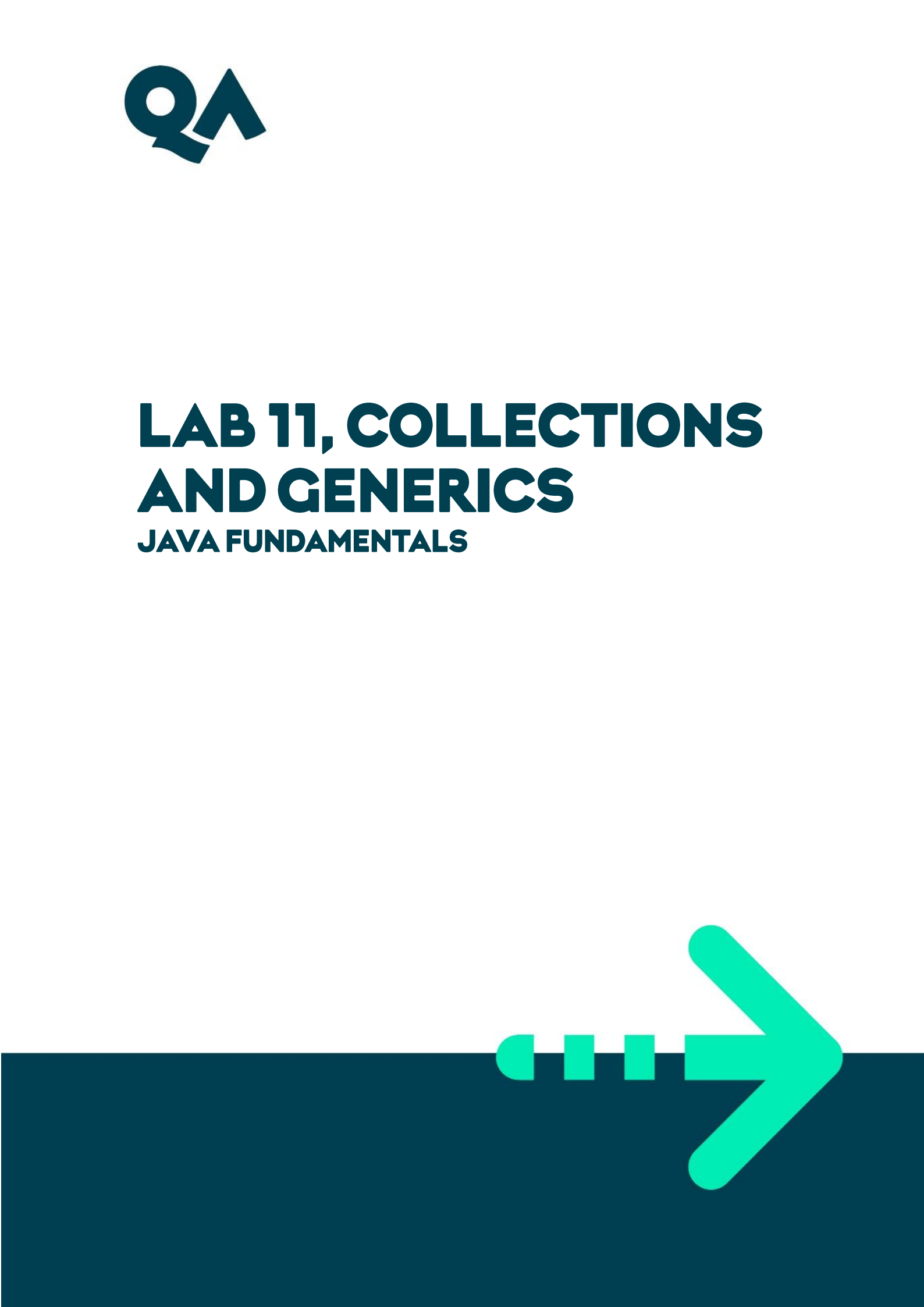

LAB 11, COLLECTIONS AND GENERICS

JAVA FUNDAMENTALS

Lab 11, Collections and Generics

Objective

In this lab you will practice using ArrayList, Queues and in a separate task, you'll work with a HashMap.

Part 1 – Use an ArrayList<Shape>

1. Open the bouncing shape exercise that you did in the previous lab.
2. Change the array of Shapes (in Game class) to ArrayList<Shape>.
   Tip: you'll need to use the ArrayList's add method in the constructor to add the three shapes;

Part 2 - Using a Queue and a Stack

1. Back in the labs project which you created in *Lab1*, add a new package called lab11.
2. Add a new class called *Program* to this package with a main() method.
3. Add another class called ShoppingBasket with following fields
   String productName.
   int quantity.
   double price
4. Create a constructor for the ShoppingBasket class and then create a method called displayDetails() to display the fields' values in Console.
5. In the Program class, create a Queue of ShoppingBasket called baskets.
   Since we are going to use this queue in main(), it needs to be static.
6. Create a static method in the Program class called buy().
   Call the buy() method from within main().
7. Write code in the buy() method to add a few ShoppingBaskets to the baskets Queue. We will process these in another method.
8. Create another static method called processBaskets()
   Write code to remove items from the baskets queue and call their displayDetails() method.

   This method simulates processing of shopping baskets for payment and shipping. For simplicity we just investigate the queue actions.
9. Call the processBaskets() method in main().
10. Run and test your code.

Part 2 Using HashMap<K,V>

Scenario: A Zoo has a number of current animals and is expecting new arrivals soon. They wish to keep track of which animal types they have and record the count of each animal type.

You will create HashMap<String, int>. The key of String to store animal type and a value of Integer for the count of occurrences (Lion 3, Zebra 2 etc).

Step by step instructions.

1. Comment out the code in main() to get ready for this exercise.
2. Create a class called Zoo with a default constructor.
3. Create an instance of Zoo in main(). This will kick start the constructor where all your code will be placed.
4. Declared and initialise the following fields.

   The following String arrays contain the names of the existing animals and the new animals we will add to our zoo. The HashMap will keep track of the animals and their count.

   ```
   HashMap<String, Integer> animalMap = null;
   String[] originalAnimals = {"Zebra", "Lion", "Buffalo"};
   String[] newAnimals = {"Zebra", "Gazelle", "Buffalo", "Zebra"};
   ```

5. Instantiate the animalMap (new it) in the class constructor.
6. Create a method in Zoo as void addAnimals(String[ ] animals)
7. Call `addAnimals()` twice from the constructor and pass each of the 2 arrays (*originalAnimals* and *newAnimals*) to this method.
8. addAnimals() should iterates over the `String[]` (method parameter) and add each String to the animalMap. If the animal is in the HashMap then its count will be increased by 1 otherwise its count will be set to 1.

   Tip: use HashMap's `containsKey()` to see if animal exists in the map.

   You will also use the HashMap's `put()` method to put an entry back in the collection and use the HashMap's `get()` method to get the count of an animal type.

9. Create a new method called `displayAnimalData`(). Call this method from the constructor.

   Tip: Best use and enhanced for loop.

   This method is going to display key/value pairs in 2 columns like this.

   ```
   Zebra      3
   Lion       2
   Buffalo    5
   ```

10. Run and test your code

** End **