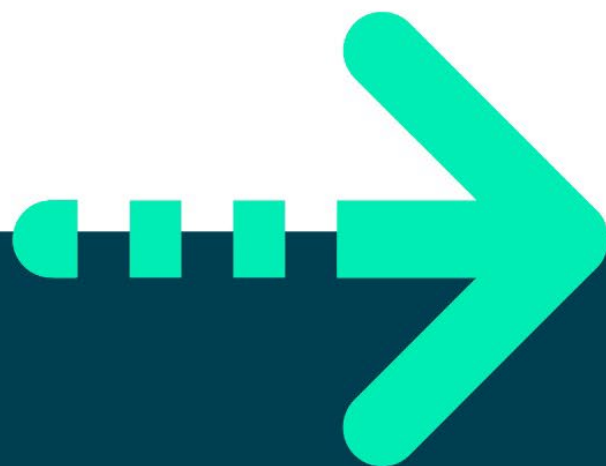




JAVA COLLECTIONS AND GENERIC





Objective

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

Part 1 – Using 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>. Use the ArrayList's add method in the constructor of Game to add the three shapes.

Part 2 – Using a Queue and a Stack

1. Back in the **labs** project add a new class called **Program** with a main() method and package called **exercise5**.
2. Add another class in the **exercise5** package called **ShoppingBasket** with following fields:

```
String productName;  
int quantity;  
double price;
```

3. Create a constructor for the ShoppingBasket class and then create a method called displayDetails() to display the fields' values.
4. In the Program class (above the main() method), create a Queue of ShoppingBasket called **baskets**.
Since we are going to use this queue in main(), it needs to be static.
5. Create a static method in the Program class called **buy()**.
Call the buy() method from within main().
6. Write code in the buy() method to add a few **ShoppingBaskets** to the **baskets** Queue. We will process these in another method.
7. 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.
8. Call the processBaskets() method in main().
9. Run and test your code.

Part 3 – 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. Create a class called **Zoo**.
2. Create an instance of `Zoo` in `main()`.
3. Create two methods in the `Zoo` class called **`open()`** and **`displayAnimalData()`**, both with no parameters or return value.
4. Declare and initialise the following fields in the `Zoo` class:

The following `String` arrays contain the names of the existing animals and the new animals we wish to add to our zoo.

The **`HashMap`** will keep track of the animals and their count.

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

5. Instantiate the **`animalMap`** (new it) in the class constructor or in the `open()` method.
6. Create a method in `Zoo` as **`void addAnimal(String animal)`**
7. In the `Open()` method, for each name in the **`originalAnimals`**, Call **`addAnimal()`**. Do the same with the **`newAnimals`**.
8. The **`void addAnimal(String animal)`** method should detect if the animal already exists in the **`animals`** `HashMap`. If it does, it must increase its count by **1**, otherwise its count must 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. Call the **`displayAnimalData()`** method from within `main()` to show all the animal names and their count.

Tip: Best use an enhanced for loop.

This method is going to display key/value pairs in two columns like this:

```
3, Zebra  
1, Gazelle  
1, Lion  
2, Buffalo
```

10. Run and test your code.

