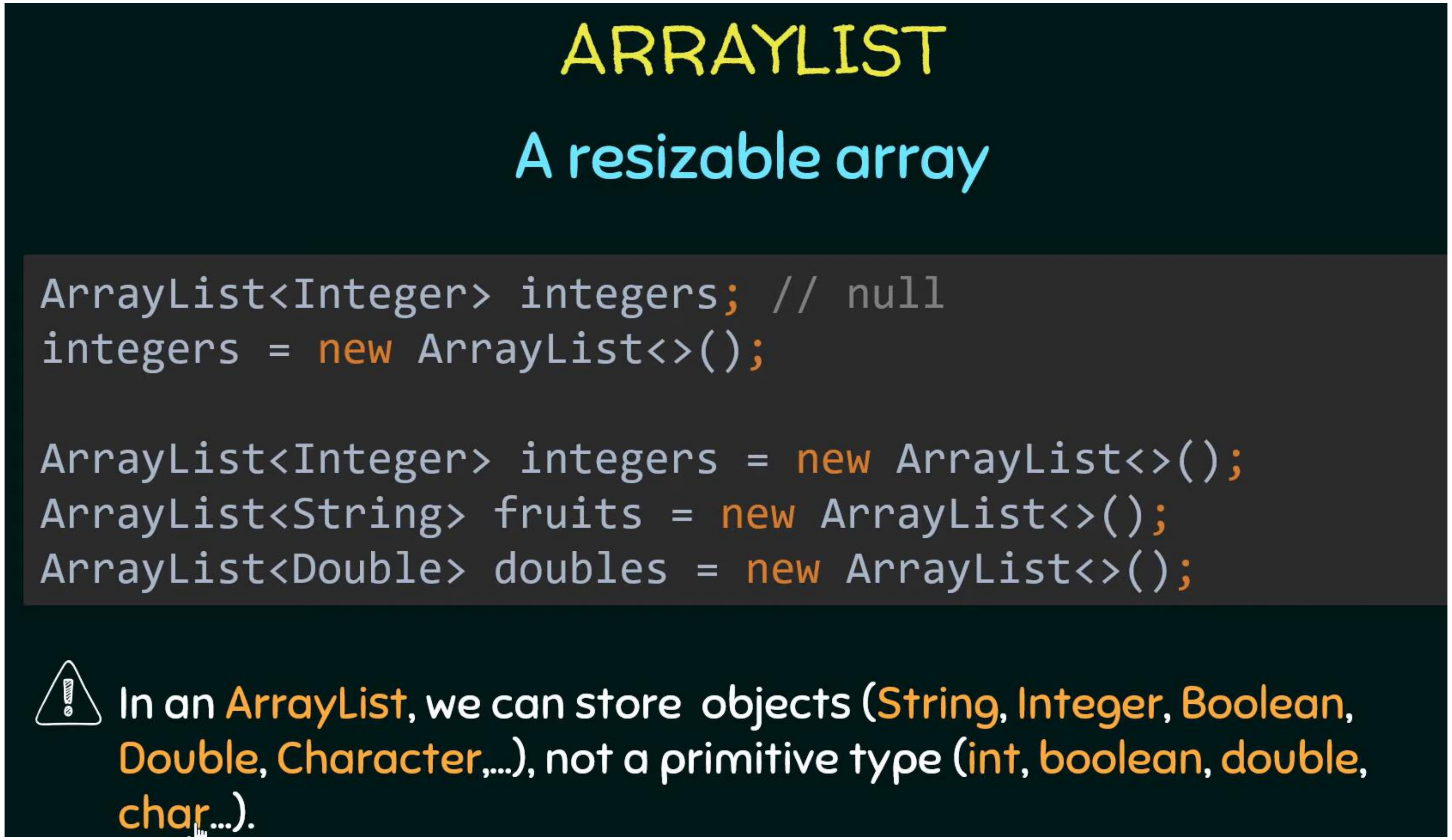
ArrayList



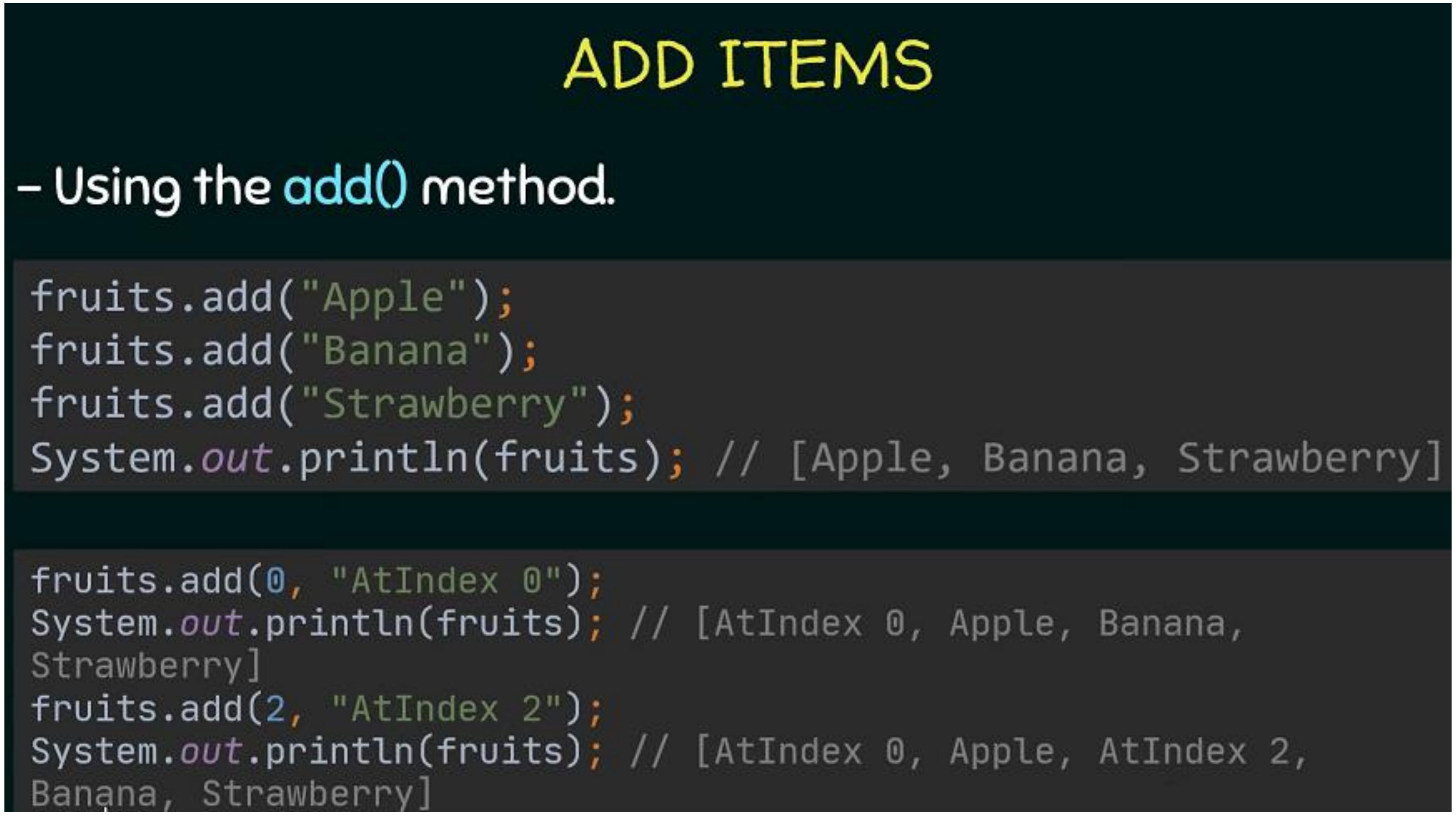
ARRAYLIST

A resizable array

```
ArrayList<Integer> integers; // null
integers = new ArrayList<>();

ArrayList<Integer> integers = new ArrayList<>();
ArrayList<String> fruits = new ArrayList<>();
ArrayList<Double> doubles = new ArrayList<>();
```

⚠ In an ArrayList, we can store objects (String, Integer, Boolean, Double, Character,...), not a primitive type (int, boolean, double, char...).

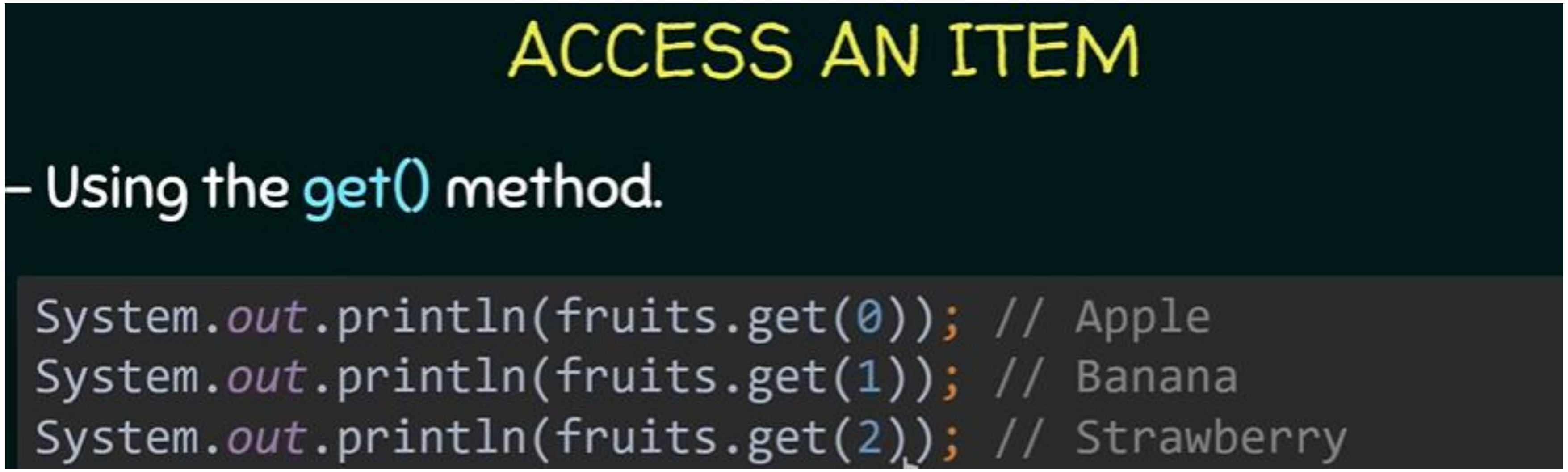


ADD ITEMS

– Using the add() method.

```
fruits.add("Apple");
fruits.add("Banana");
fruits.add("Strawberry");
System.out.println(fruits); // [Apple, Banana, Strawberry]
```

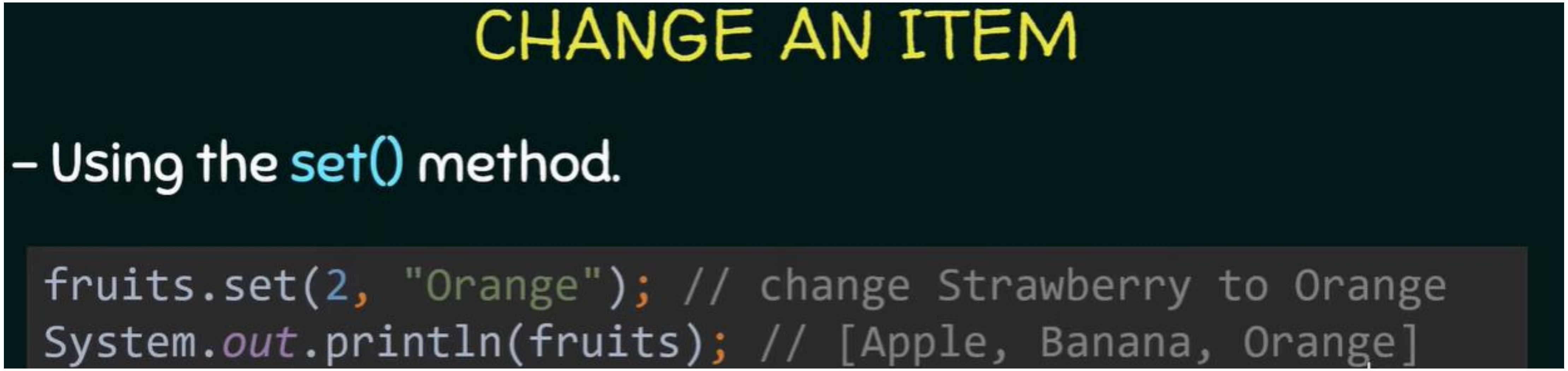
```
fruits.add(0, "AtIndex 0");
System.out.println(fruits); // [AtIndex 0, Apple, Banana,
Strawberry]
fruits.add(2, "AtIndex 2");
System.out.println(fruits); // [AtIndex 0, Apple, AtIndex 2,
Banana, Strawberry]
```



ACCESS AN ITEM

- Using the get() method.

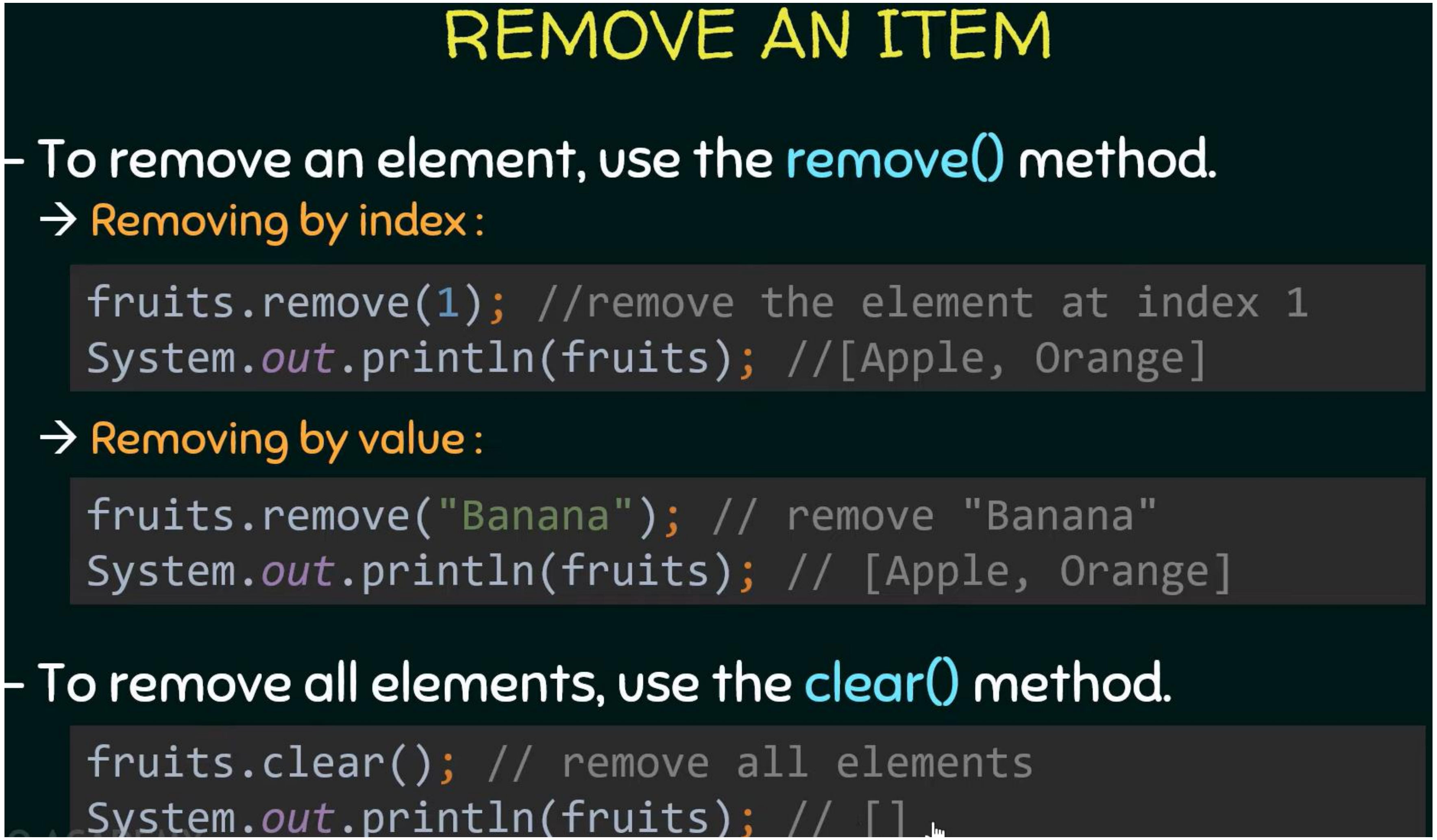
```
System.out.println(fruits.get(0)); // Apple
System.out.println(fruits.get(1)); // Banana
System.out.println(fruits.get(2)); // Strawberry
```



CHANGE AN ITEM

– Using the set() method.

```
fruits.set(2, "Orange"); // change Strawberry to Orange
System.out.println(fruits); // [Apple, Banana, Orange]
```



REMOVE AN ITEM

- To remove an element, use the remove() method.
  → Removing by index :

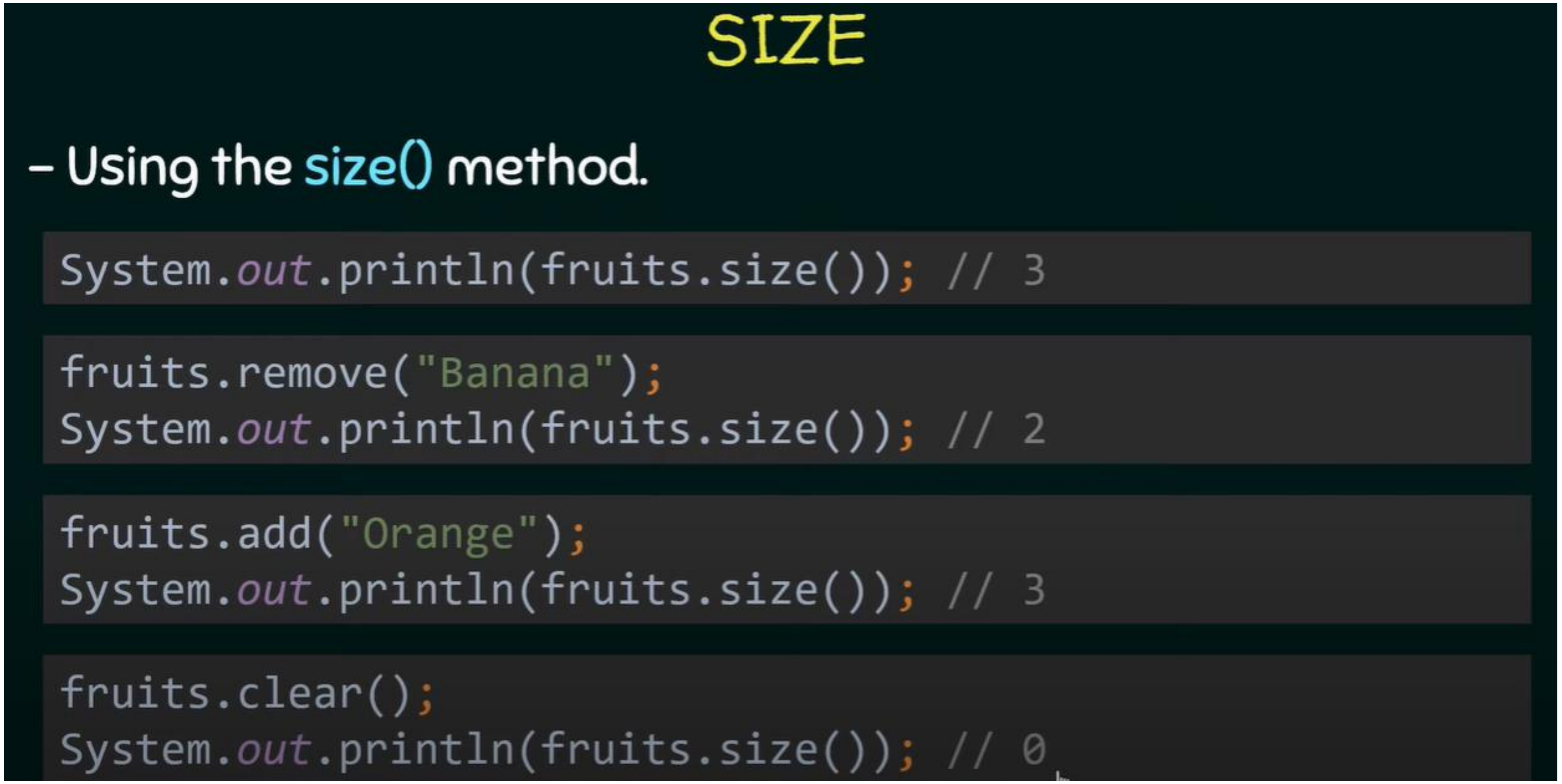
```
fruits.remove(1); //remove the element at index 1
System.out.println(fruits); //[Apple, Orange]
```

  → Removing by value :

```
fruits.remove("Banana"); // remove "Banana"
System.out.println(fruits); // [Apple, Orange]
```

- To remove all elements, use the clear() method.

```
fruits.clear(); // remove all elements
System.out.println(fruits); // []
```



SIZE

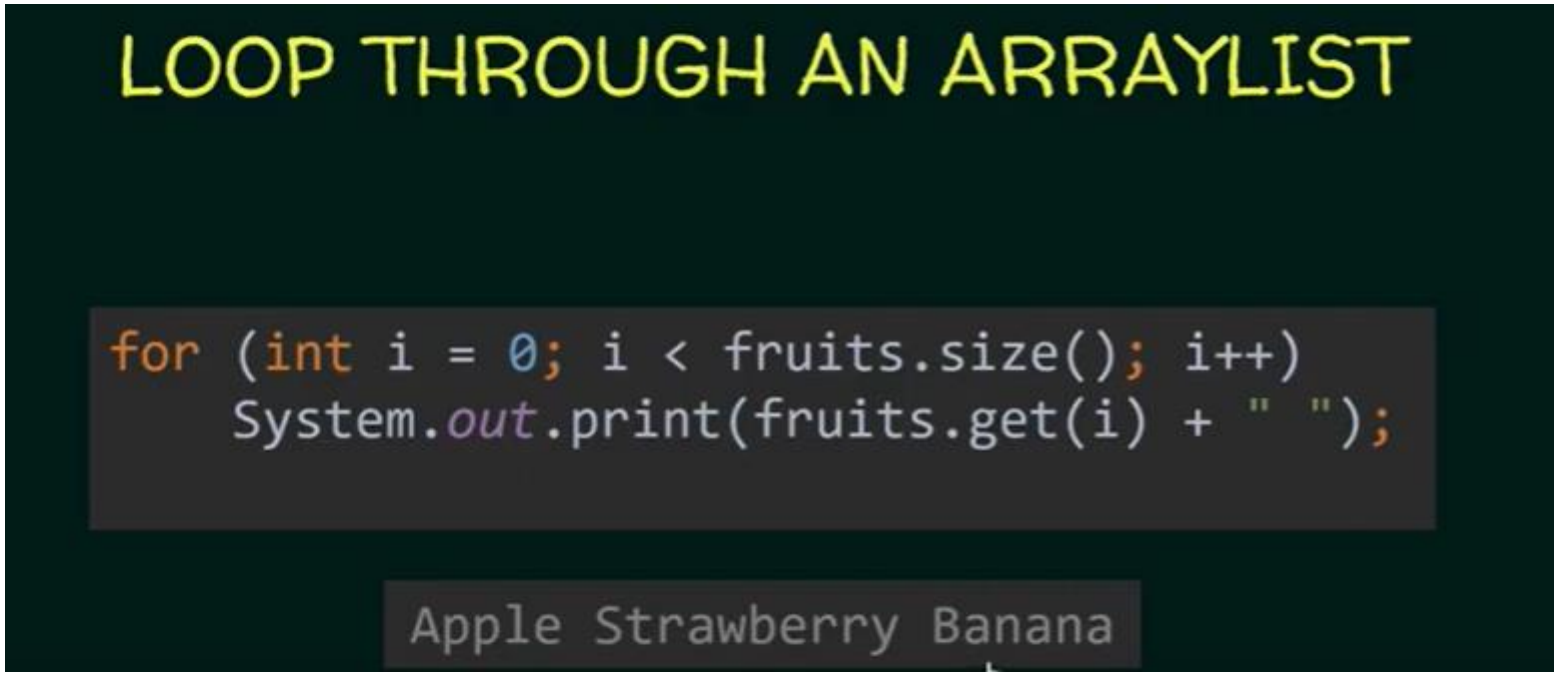
- Using the size() method.

```
System.out.println(fruits.size()); // 3
```

```
fruits.remove("Banana");
System.out.println(fruits.size()); // 2
```

```
fruits.add("Orange");
System.out.println(fruits.size()); // 3
```

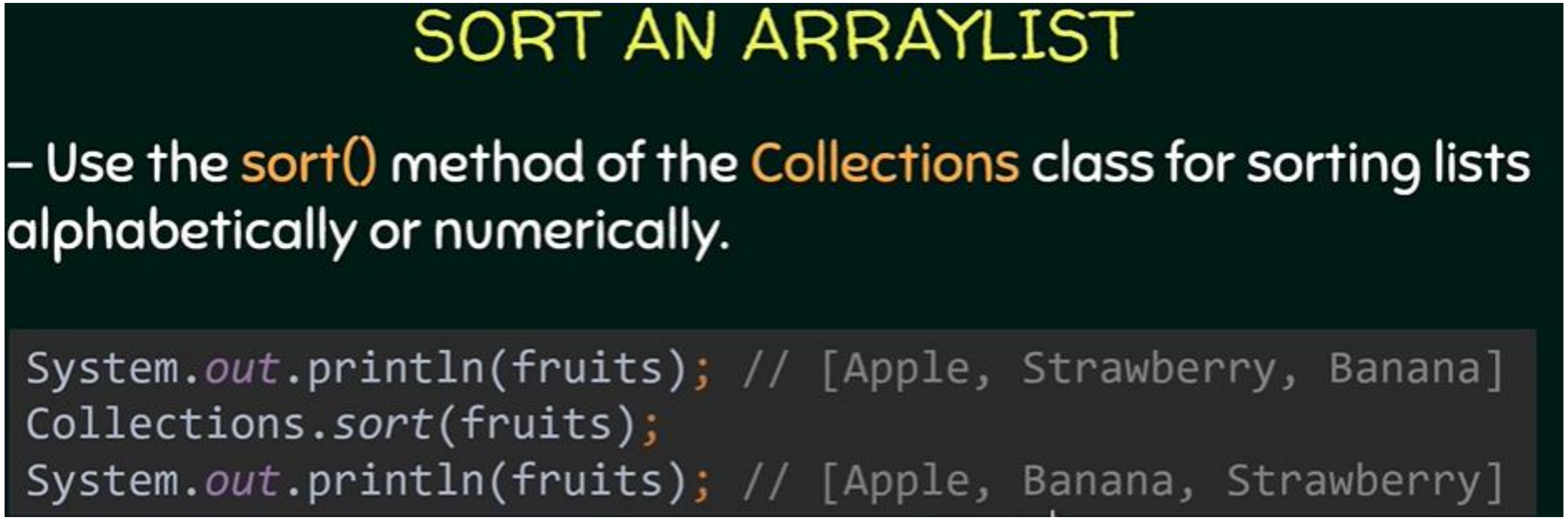
```
fruits.clear();
System.out.println(fruits.size()); // 0
```



LOOP THROUGH AN ARRAYLIST

```
for (int i = 0; i < fruits.size(); i++)
    System.out.print(fruits.get(i) + " ");
```

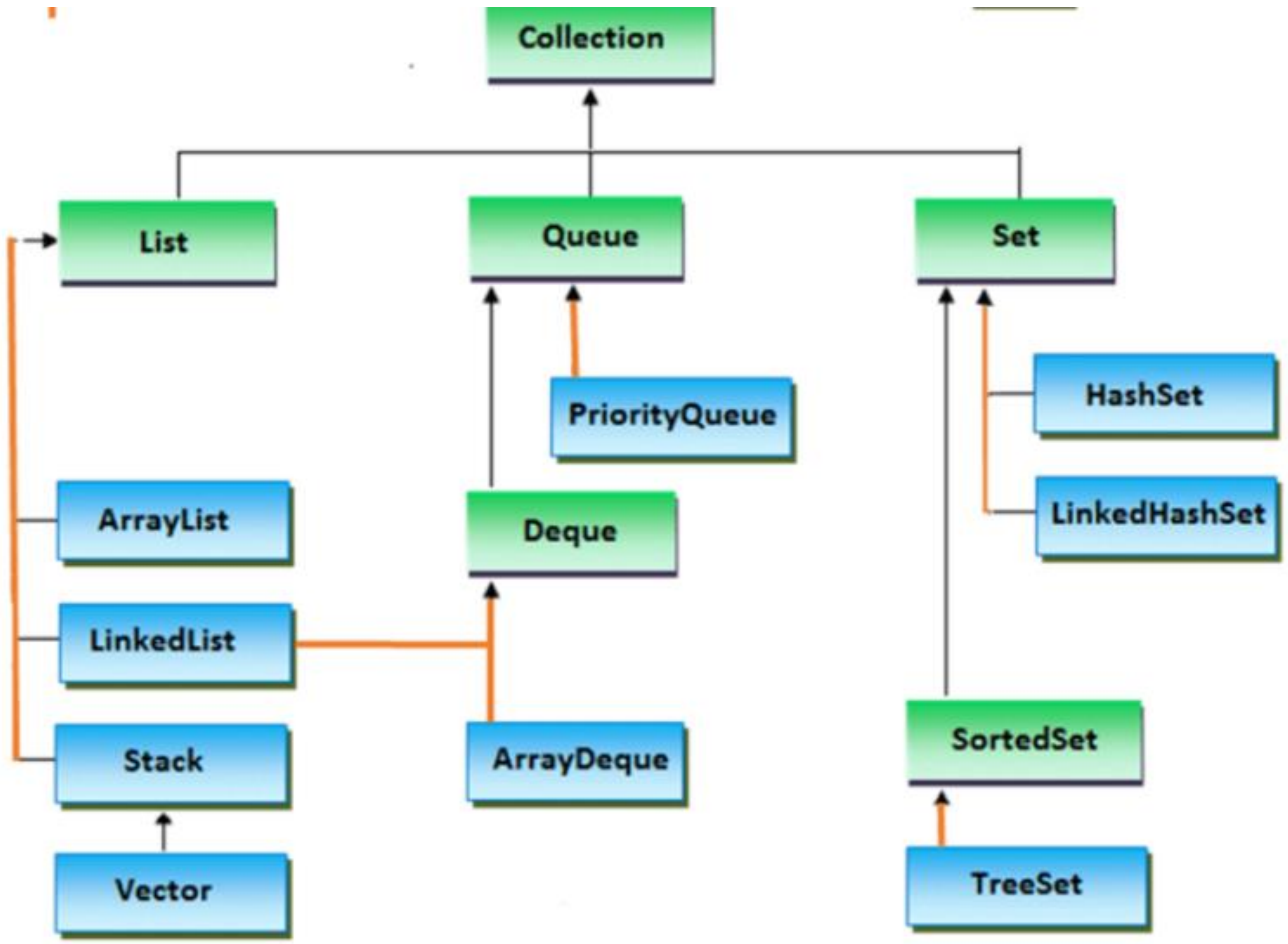
```
Apple Strawberry Banana
```

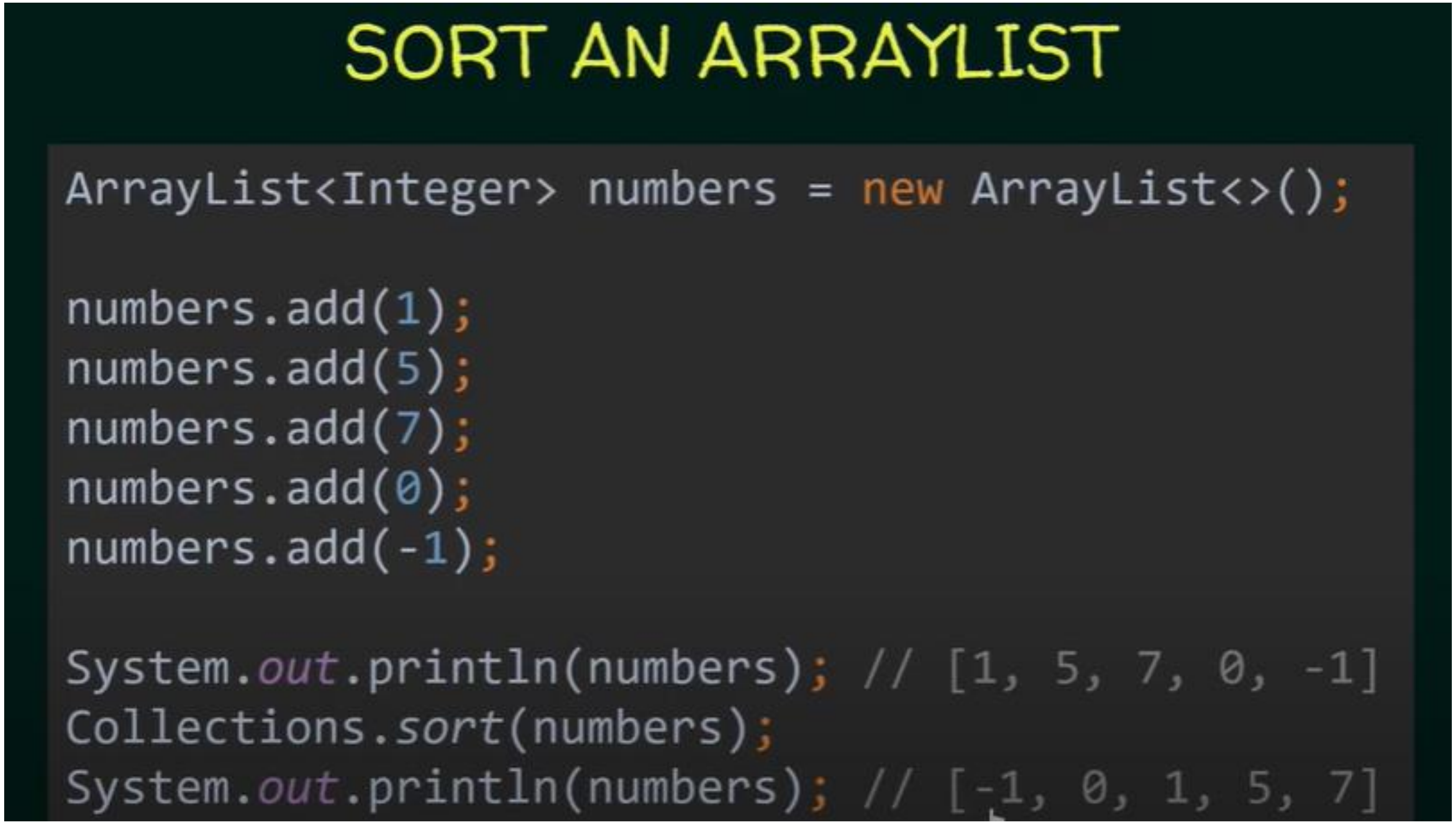


SORT AN ARRAYLIST

– Use the sort() method of the Collections class for sorting lists alphabetically or numerically.

```
System.out.println(fruits); // [Apple, Strawberry, Banana]
Collections.sort(fruits);
System.out.println(fruits); // [Apple, Banana, Strawberry]
```


Collection
List
Queue
Set
PriorityQueue
HashSet
LinkedHashSet
ArrayList
Deque
LinkedList
Stack
ArrayDeque
SortedSet
Vector
TreeSet



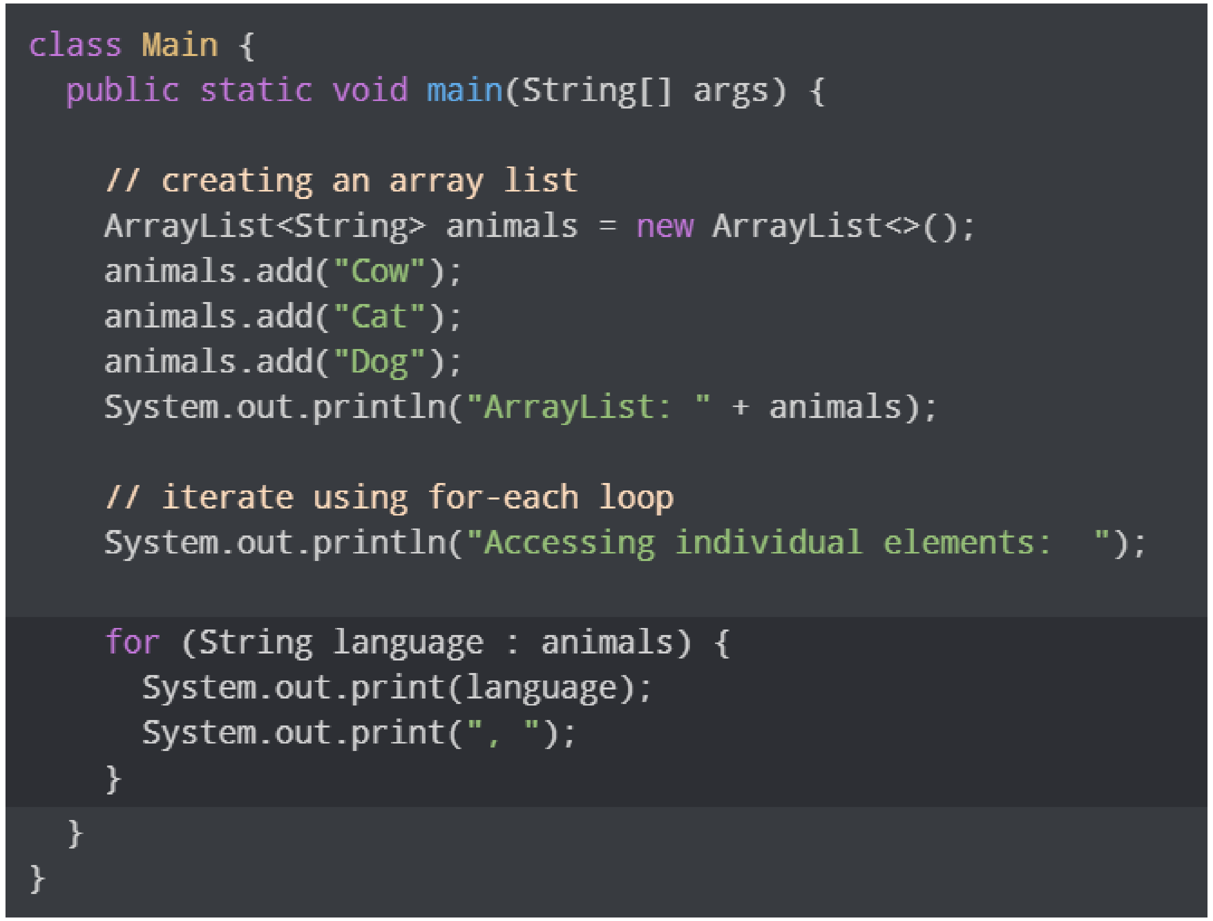
SORT AN ARRAYLIST

```java
ArrayList<Integer> numbers = new ArrayList<>();

numbers.add(1);
numbers.add(5);
numbers.add(7);
numbers.add(0);
numbers.add(-1);

System.out.println(numbers); // [1, 5, 7, 0, -1]
Collections.sort(numbers);
System.out.println(numbers); // [-1, 0, 1, 5, 7]
```

Iterate using for loop



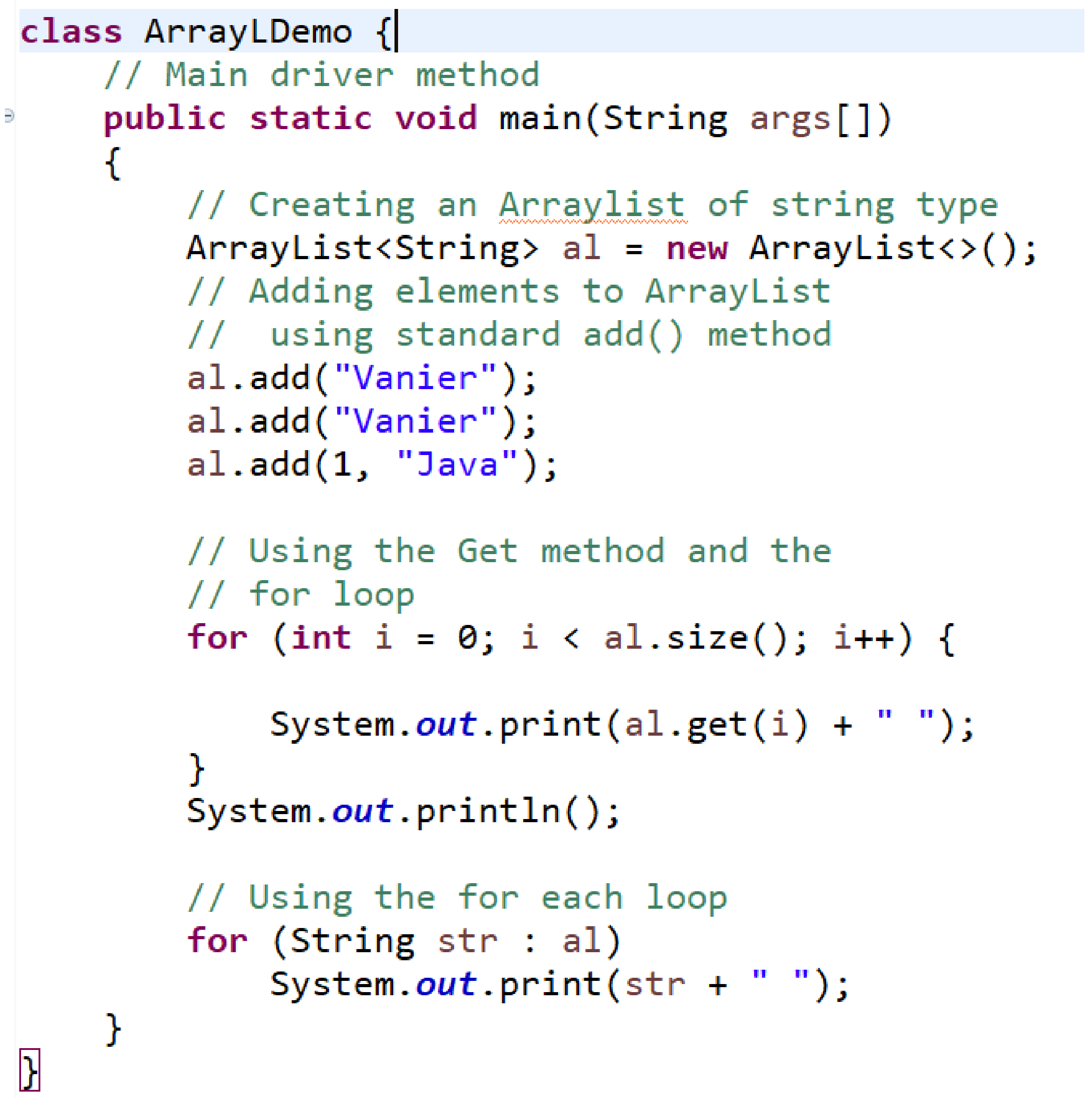
```
class Main {
  public static void main(String[] args) {

    // creating an array list
    ArrayList<String> animals = new ArrayList<>();
    animals.add("Cow");
    animals.add("Cat");
    animals.add("Dog");
    System.out.println("ArrayList: " + animals);

    // iterate using for-each loop
    System.out.println("Accessing individual elements:  ");

    for (String language : animals) {
      System.out.print(language);
      System.out.print(", ");
    }
  }
}
```

String

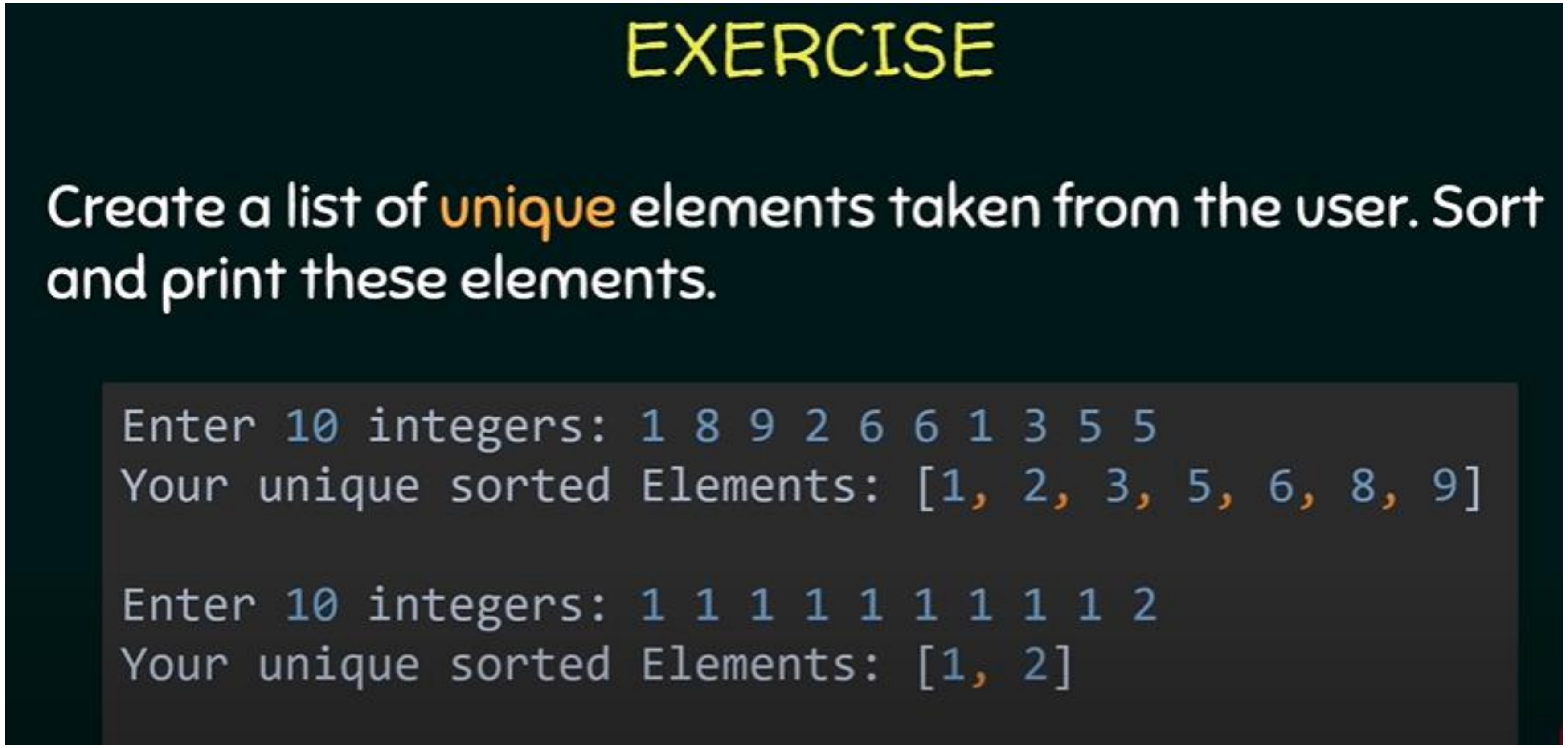


```java
class ArrayLDemo {
    // Main driver method
    public static void main(String args[])
    {
        // Creating an Arraylist of string type
        ArrayList<String> al = new ArrayList<>();
        // Adding elements to ArrayList
        //  using standard add() method
        al.add("Vanier");
        al.add("Vanier");
        al.add(1, "Java");

        // Using the Get method and the
        // for loop
        for (int i = 0; i < al.size(); i++) {

            System.out.print(al.get(i) + " ");
        }
        System.out.println();

        // Using the for each loop
        for (String str : al)
            System.out.print(str + " ");
    }
}
```



EXERCISE

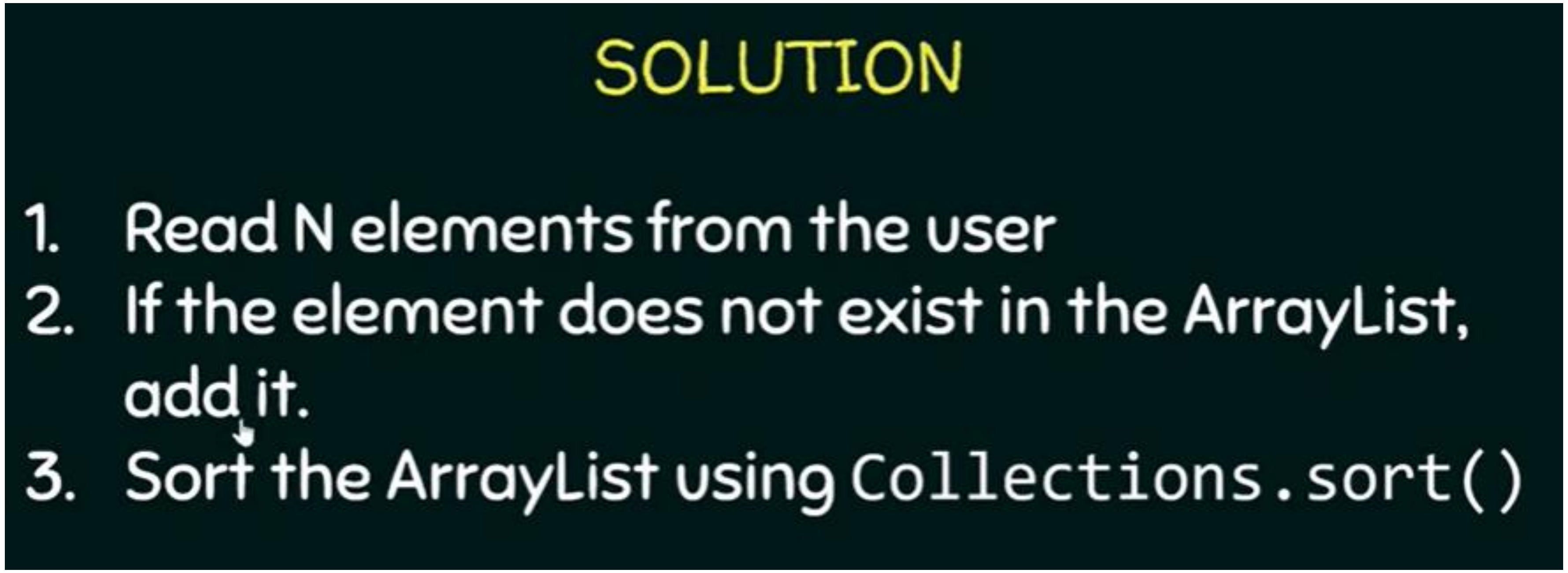
Create a list of unique elements taken from the user. Sort and print these elements.

```
Enter 10 integers: 1 8 9 2 6 6 1 3 5 5
Your unique sorted Elements: [1, 2, 3, 5, 6, 8, 9]

Enter 10 integers: 1 1 1 1 1 1 1 1 1 2
Your unique sorted Elements: [1, 2]
```

Use the exercise for strings too

- Input : “Vanier” “Vanier” “Java” “Java”
- Output: Vanier, Java


SOLUTION

1. Read N elements from the user
2. If the element does not exist in the ArrayList, add it.
3. Sort the ArrayList using `Collections.sort()`