

KEEP MOVING  
**FORWARD**



## Beginner : 1D Arrays 3

### TABLE OF CONTENTS

- Revision of previous class
- Swap 2 elements
- Reverse an array
- Drawbacks of Array
- ArrayList introduction
- Functions [add, get, set, remove, size, sort]
- ArrayList functions via code
- Write a function which takes arraylist and return all even numbers
- Unique Elements
- Return an arraylist with elements having at least 1 element greater than it
- toArray() method for converting arraylist to array



Aakash Trivedi  
Anand Ingaleshwar  
Balaji Naidu  
Chethan I B  
Diksha  
G Chittibabu  
Jagtar Singh  
Kamal Thirpola  
Kannan  
Karthik Srinivasan  
Madhu Sahithi  
**NALLABOTHULA AMARACHINNESWARI**  
Pranshi Chauhan  
Pritam  
Priyanka Kadam  
Shambhavee  
sukhleen  
**SUMANTH S**  
Vijay Kolar  
**Yashawanth P**



# Revision

## Revision Quiz 1

What happens to the original array after a function modifies the passed array?

- A Changes are not reflected in the original array
- B Changes are reflected in the original array
- C A runtime error occurs
- D The array is unchanged but copied

50 10 20 30 40

## Revision Quiz 2

What is the result of a right shift operation on the array {10, 20, 30, 40, 50}?

- A {50, 40, 30, 20, 10}
- B {50, 10, 20, 30, 40}
- C {20, 30, 40, 50, 10}
- D {40, 50, 10, 20, 30}

## Revision Quiz 3

Which of these is not a correct way to initialize an array in Java?

- A int[] array = new int[5];
- B int[] array = {1, 2, 3, 4, 5};
- C int[] array; array = new int[5];
- D int[] array = new int[];



0 1 2 3 4  
1 2 3 4 5

### Revision Quiz 4

How to access ith element of an array A ?

- |   |        |   |       |
|---|--------|---|-------|
| A | A[i]   | B | A[i]  |
| C | A[i+1] | D | A[2i] |

### Revision Quiz 5

increasing

Is the array [1, 22, 33, 44, 55] sorted in ascending order?

- |   |     |   |    |
|---|-----|---|----|
| A | Yes | B | No |
|---|-----|---|----|



## Swap 2 elements

### Problem Statement

Given 2 numbers  $a$  and  $b$ , swap their values.

Example :

Input:

$a = 3, b = 7$

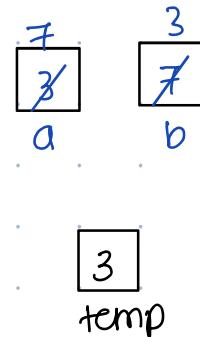
Output:

$a = 7, b = 3$

```
→ int a = 3;  
→ int b = 7;  
→ int temp = a;  
→ a = b;  
→ b = temp
```

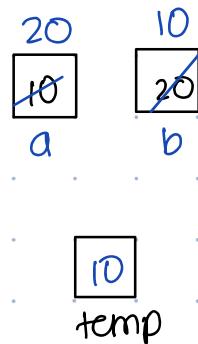


Approach



Swap

```
→ int temp = a;  
→ a = b;  
→ b = temp;
```



Java  
known limitations

```
1 public class MainClass {  
2     public static void main(String[] args) {  
3         int a = 11;  
4         int b = 22;  
5  
6         int temp = a;  
7         a = b;  
8         b = temp;  
9     }  
10 }
```

Frames

main:9
a 22
b 11
temp 11



## Reverse an array

Given an array 'arr' of size 'N'. Reverse the entire array.

$N = 5$

Input

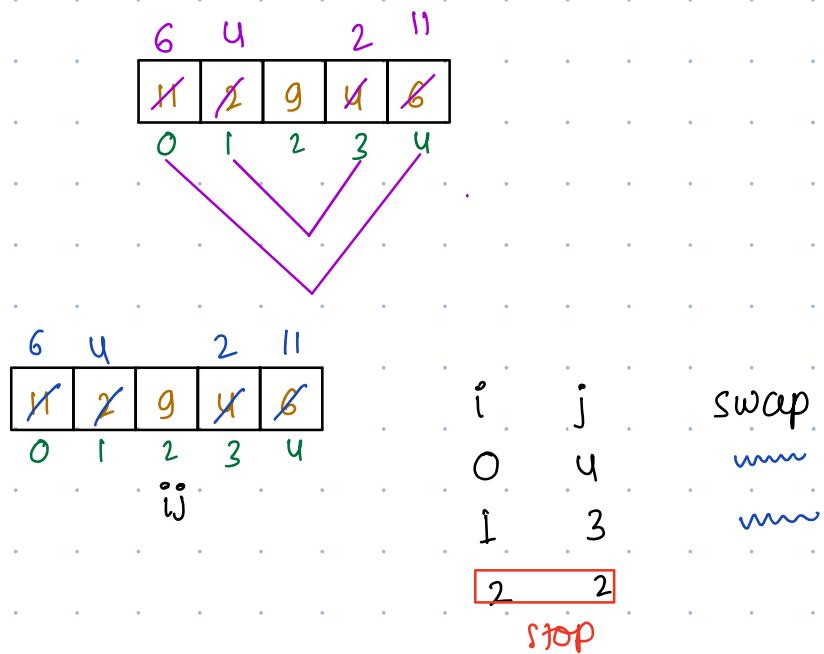
11	2	9	4	6
0	1	2	3	4

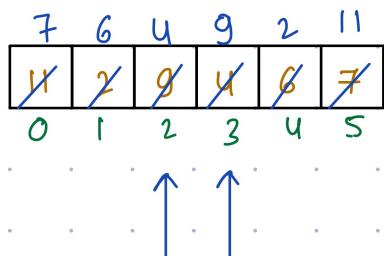
  

6	4	9	2	11
0	1	2	3	4

<Question> :

### 💡 Approach





i < j	swap	i++	j--
0	A <sub>0</sub> A <sub>5</sub>	1	4
1	A <sub>1</sub> A <sub>4</sub>	2	3
2	A <sub>2</sub> A <sub>3</sub>	3	2
3	2		

Java  
known limitations

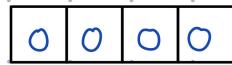
```
1 public class MainClass {  
2     public static void main(String[] args) {  
3         int[] A = new int[]{7, 1, 5, 9, 2, 4};  
4         int n = A.length;  
5  
6         int i = 0;  
7         int j = n-1;  
8  
9         while(i < j){  
10             // swap A[i] with A[j]  
11             int temp = A[i];  
12             A[i] = A[j];  
13             A[j] = temp;  
14             // update i, j  
15             i++;  
16             j--;  
17         }  
18  
19     }  
20 }
```

Frames      Objects

main:19	array
A	0 1 2 3 4 5
n	6
i	3
j	2



```
int[] A = new int[4];
```



## Drawback of Arrays

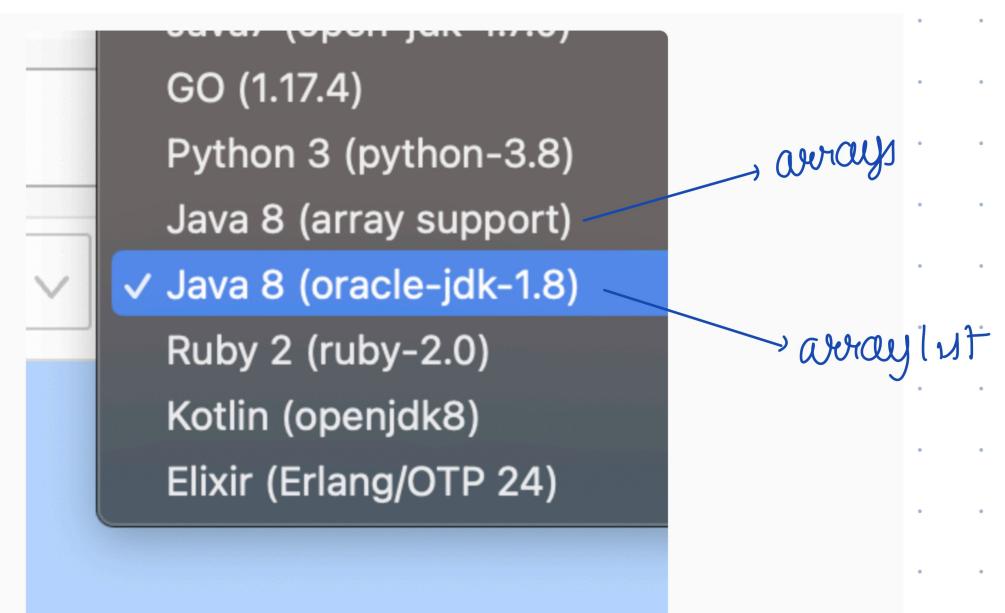
- Once array size is fixed, it cannot be changed.
- If we want to change, we need to create a new array.

`int[] ar=new int[4];`

This can only store 4 elements, if we want to store 1 more element we cannot update the size. We have to create a new array only.

## Introduction to **ArrayList** and its properties

Note : For solving questions on ArrayList, we have to select "Java 8 (oracle-jdk-1.8)" from dropdown while selecting language.





### Definition :

ArrayList provides a resizable array implementation that is similar to an ordinary array, but with the added benefit of being able to resize as elements are added or removed.

### Syntax :

`ArrayList<datatype> varName = new ArrayList<>();`

<code>int x</code>	<code>Integer</code>
<code>double x</code>	<code>Double</code>
<code>boolean x</code>	<code>Boolean</code>
<code>char x</code>	<code>Character</code>
<code>long x</code>	<code>Long</code>

```
class Main {
    public static void main(String[] args) {
        ArrayList<Integer> al = new ArrayList<>(); // empty ArrayList
        al.add(1);
        al.add(5);
        al.add(2);

        // To find the size
        System.out.println(al.size());
        // To get the value at index = 2
        System.out.println(al.get(2));
        // Set the value at index = 2
        al.set(2, 20);
        // To print the ArrayList
        System.out.println(al);
    }
}
```



## Some Methods in ArrayList

- Adding an element at the end

We can add an element at the end of ArrayList using the add(value) method :

```
ArrayList<Integer> al = new ArrayList<>();
al.add(1); // [1]
al.add(9); // [1, 9]
al.add(13); // [1, 9, 13]
```

- Access ith index element of an ArrayList

We can access ith index element of an ArrayList using the get(index) method :

```
al = [1, 9, 13]
SOP(al.get(1)); // 9
SOP(al.get(2)); // 13
```



## Quiz 1

Predict the output:

```
ArrayList<Integer> al = new ArrayList<>();  
al.add(10); [10]  
al.add(20); [10, 20]  
al.add(30); [10, 20, 30]  
al.add(40); [10, 20, 30, 40]  
System.out.print(al.get(2));
```

↑  
index

- |   |       |   |    |
|---|-------|---|----|
| A | 10    | B | 20 |
| C | 30    | D | 40 |
| E | Error |   |    |

## Quiz 2

Predict the output:

```
ArrayList<Integer> al = new ArrayList<>();  
al.add(10); [10]  
al.add(20); [10, 20]  
al.add(30); [10, 20, 30]  
al.add(40); [10, 20, 30, 40]  
System.out.print(al.get(4));
```

→ Index Out of Bounds

- |                                       |       |   |    |
|---------------------------------------|-------|---|----|
| A                                     | 40    | B | 20 |
| <input checked="" type="checkbox"/> C | Error | D | 10 |

10:17



## Properties of ArrayList

- Update existing element

We can update the existing element of an ArrayList using the `set(index, value)` method :

al = 0 1 2 3 4  
10, 20, 30, 40, 50  
al.set(4, 100); // 10 20 30 40 100

- Remove an element

We can remove an element from the ArrayList using the `remove(index)` method:

al = 10 20 30 40  
0 1 2 3  
al.remove(2); // 10 20 40  
0 1 2

- Sort the arraylist

We can sort the ArrayList using the `Collections.sort(arraylist_name)` method :

// al 5 9 2 14 3  
Collections.sort(al); // ascending order  
// al 1 2 3 4 5 9

HW find how to sort in descending order



### Quiz 3

Predict the output:

```
public static void main(String[] args) {  
    ArrayList<Integer> ar = new ArrayList<>();  
    ar.add(1); 1  
    ar.add(2); 1 2  
    ar.add(3); 1 2 3  
    ar.set(1, 5); 1 5  
    ar.set(2, ar.get(0) + ar.get(1)); 1 5 6  
  
    System.out.println(ar);  
}
```

A [1, 5, 3]

B [1, 7, 3]

C [1, 5, 6]

D [1, 6, 3]

### Quiz 4

Predict the output:

```
public static void main(String[] args) {  
    ArrayList<Integer> ar = new ArrayList<>();  
    ar.add(-5); -5  
    ar.add(20); -5 20  
    ar.add(19); -5 20 19  
    ar.add(50) -5 20 19 50  
  
    ar.remove(1);  
  
    System.out.println(ar);  
}
```

A [-5, 19, 50]

B [20, 19, 50]

C [-5, 20, 50]

D [-5, 20, 19, 50]

### Quiz 5

Predict the output:

```
public static void main(String[] args) {  
    ArrayList<Integer> ar = new ArrayList<>();  
    ar.add(5); 5  
    ar.add(2); 5 2  
    ar.add(9); 5 2 9  
    ar.add(1); 5 2 9 1  
  
    Collections.sort(ar); // 1 2 5 9  
  
    System.out.println(ar);  
}
```

A [5, 2, 9, 1]

B [9, 5, 2, 1]

C [1, 2, 5, 9]

D [2, 1, 5, 9]



# Update all values of arrayList by 1

## Problem Statement

Write a function which takes arrayList as input and update all values by 1

### Example :

```
temp : [20, 15, 8, 25, 21]
ans : [21, 16, 9, 26, 22]
```

```
ArrayList<Integer> al = new ArrayList<>();
al.add(5);
al.add(9);
al.add(7);

int n = al.size();
// 6 10 8
// 0 1 2
// ^
for(int i = 0; i < n; i++){
    int val = al.get(i);
    al.set(i, val+1);
}
```

[Success] Your code was executed successfully

[6, 10, 8]

Custom Input



# Return all the even numbers

## Problem Statement

Given an `ArrayList` of integers, return all the even numbers in the `ArrayList`.

*new ArrayList*

Example 1 :

input      `al = { 1, 9, 5, 2, 4, 10, 15 }`

output     `{ 2, 4, 10 }`

Example 2 :

input :    `al = { -1, -2, -3, -4, 2, 9, 10 }`

output :    `{ -2, -4, 2, 10 }`

Java 8 (oracle-jdk-1.8) ▾

Run ▾

More ▾

Run ▾

```
index-1 • +  
1 import java.util.*;  
2 public class Main {  
3     static ArrayList<Integer> findEven(ArrayList<Integer> al){  
4         ArrayList<Integer> ans = new ArrayList<>();  
5  
6         for(int i = 0; i < al.size(); i++){  
7             int val = al.get(i);  
8             if(val % 2 == 0){ // even  
9                 ans.add(val);  
10            }  
11        }  
12  
13        return ans;  
14    }  
15    public static void main(String[] args) {
```

Custom Output  
[Success] Your code was executed successfully  
[2, 4, 6]  
Custom Input

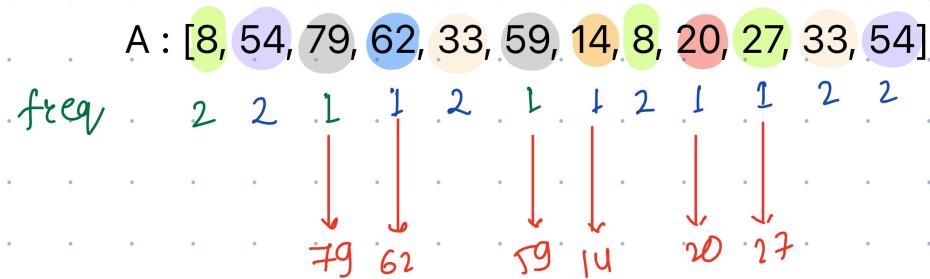


## Return an arraylist having only the unique elements from given array

Given a dynamic Array A. // ArrayList

Return a dynamic Array which has All Unique Elements from the given Array.

Example Input



Idea

For each element in arraylist find its freq  
if freq == 1 add element in answer .



```
import java.util.*;  
public class Main {  
    static int getFreq(ArrayList<Integer> al, int k){  
        int count = 0;  
        for(int i = 0; i < al.size(); i++){  
            int val = al.get(i);  
            if(val == k){  
                count++;  
            }  
        }  
  
        return count;  
    }  
    public static void main(String[] args) {  
        ArrayList<Integer> al = new ArrayList<>();  
        al.add(1);  
        al.add(2);  
        al.add(3);  
        al.add(1);  
        al.add(2);  
        al.add(5);  
  
        ArrayList<Integer> ans = new ArrayList<>();  
  
        for(int i = 0; i < al.size(); i++){  
            int ele = al.get(i);  
            int freq = getFreq(al, ele);  
            if(freq == 1){ // unique  
                ans.add(ele);  
            }  
        }  
    }  
}
```



## Return an **ArrayList** with elements having at least 1 element greater than itself

### Problem Statement

Given an *ArrayList*

Return *ArrayList* of elements which has at least 1 element greater than itself.

[1, 3, 10, 6, 3, 2, 10, 9]  
  X

AL = [1 2 3 4 5]  
  X

ans = [1 3 6 3 2 9]

ans = [1 2 3 4]

AL = [5, 1, 5, 2, 5, 3]  
ans = [1 2 3]

### Quiz 6

How many elements have at least one element greater than themselves in this array?  
[2, 3, 10, 7, 3, 2, 10, 8, 6]

A 4

B 5

C 6

D 7

✓ ✓ 10 ✓ ✓ 2 ✓ 10 ✓ 6



$$AL = [5 \ 5 \ 5 \ 10 \ 10 \ 11]$$

Idea : {Best}

Find the max value

Add all the values except for max value in  
the answer arraylist.

Other Ideas

> For every element find if there is any  
other element which is greater



Beginner | Difficulty 3

SCALER ↗

```
import java.util.*;

public class Main {
    // 1 2 5 1 2 5
    static ArrayList<Integer> greaterAL(ArrayList<Integer> al){
        // Find the max element
        int max = Integer.MIN_VALUE;
        for(int i = 0; i < al.size(); i++){
            int ele = al.get(i);
            if(ele > max){
                max = ele;
            }
        }

        ArrayList<Integer> ans = new ArrayList<>();
        for(int i = 0; i < al.size(); i++){
            int ele = al.get(i);
            if(ele != max){
                ans.add(ele);
            }
        }
        return ans;
    }

    public static void main(String[] args) {
        ArrayList<Integer> al = new ArrayList<>();
        al.add(1);
        al.add(2);
        al.add(5);
        al.add(1);
        al.add(2);
        al.add(5);
        System.out.println(greaterAL(al));
    }
}
```

Doubt tension



































