

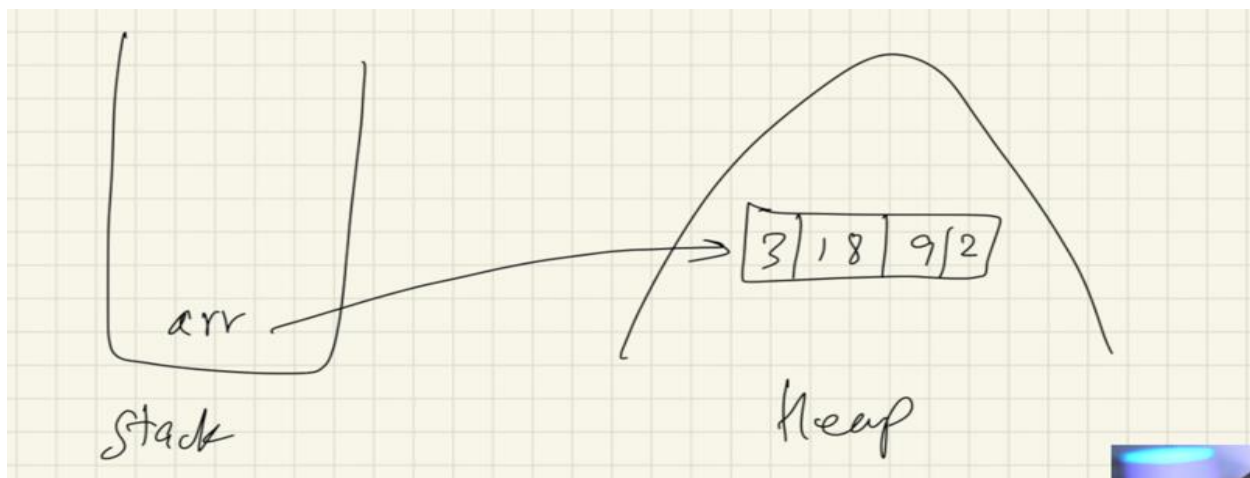
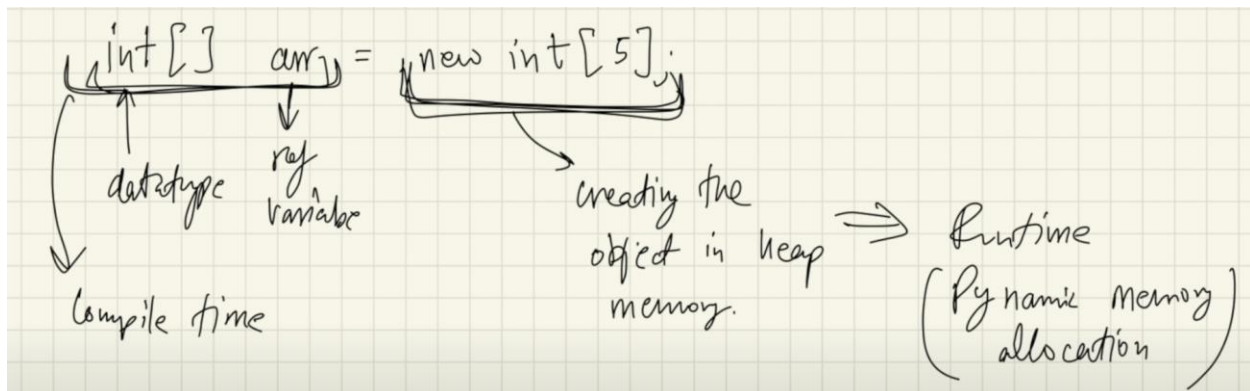
Introduction to Arrays and ArrayList in Java

Syntax

```
// Q: store 5 roll numbers
int rno1 = 23;
int rno2 = 55;
int rno3 = 18;

// syntax
// datatype[] variable_name = new datatype[size];
// store 5 roll numbers:
int[] rnos = new int[5];
// or directly
int[] rnos2 = {23, 12, 45, 32, 15};
```

```
int[] rnos; // declaration of array. rnos is getting defined in the stack
rnos = new int[5]; // actually here object is being created in the memory (heap)
```



- ① array objects are in heap
 - ② heap objects are not continuous
 - ③ DMA
- Hence! may not be continuous → Depends on JVM

```
23
24     int[] ros; // declaration of array. ros is getting d
25     ros = new int[5]; // initialisation: actually here o
26     System.out.println(ros[1]);
27
28
29
30
Run: Main x
/Users/kunalkushwaha/Library/Java/JavaVirtualMachines/openjdk-
```

Internally all the array elements will be assigned to zero.

```
Main.java x
1 package com.rahu1;
2
3 import java.util.Arrays;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         int [] arr = new int [5];
9         System.out.println(Arrays.toString(arr));
10
11     }
12 }
```

```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe"
[0, 0, 0, 0, 0]

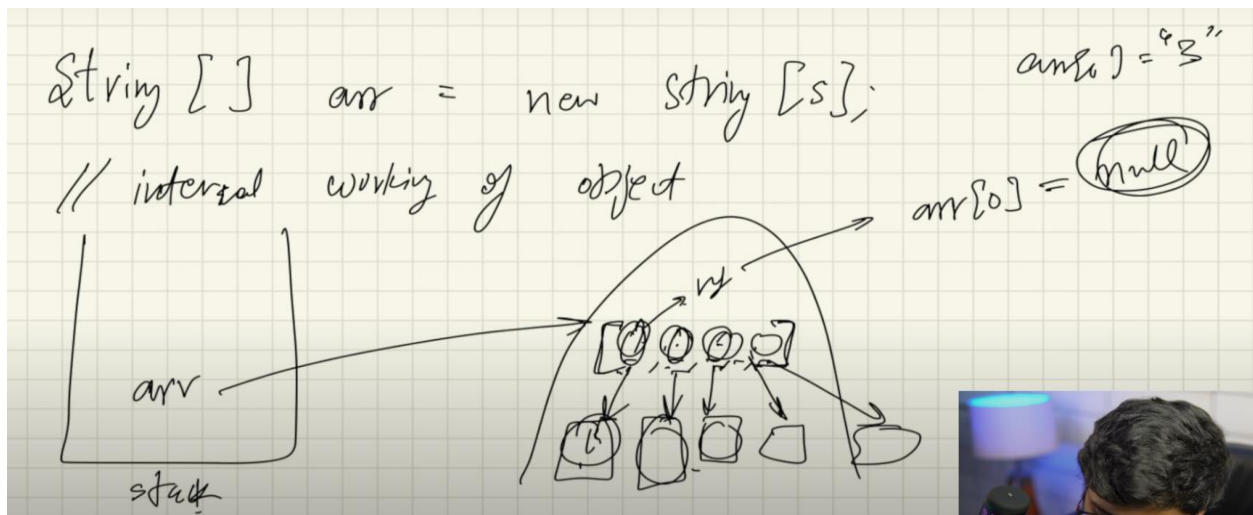
Process finished with exit code 0
```

```
Main.java x
1 package com.rahu1;
2
3 import java.util.Arrays;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         int [] arr = new int [5];
9         //System.out.println(Arrays.toString(arr));
10        String [] a = new String [5];
11        System.out.println(a[0]);
12        System.out.println(Arrays.toString(a));
13    }
14 }
```

```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe"
null
[null, null, null, null, null]

Process finished with exit code 0
```

In java, array primitive are stored in stack and object are stored in heap. In heap of array each element in index is an object which refers to another location where data is located. By default when array is created each element will be stored as zero.



```
1 package com.kunal;
2
3 public class Input {
4     public static void main(String[] args) {
5         int[] arr = new int[5];
6         arr[0] = 23;
7         arr[1] = 45;
8         arr[2] = 233;
9         arr[3] = 543;
10        arr[4] = 3;
11        // [23, 45, 233, 543, 3]
12        System.out.println(arr[3]);
13    }
14 }
```

Array input

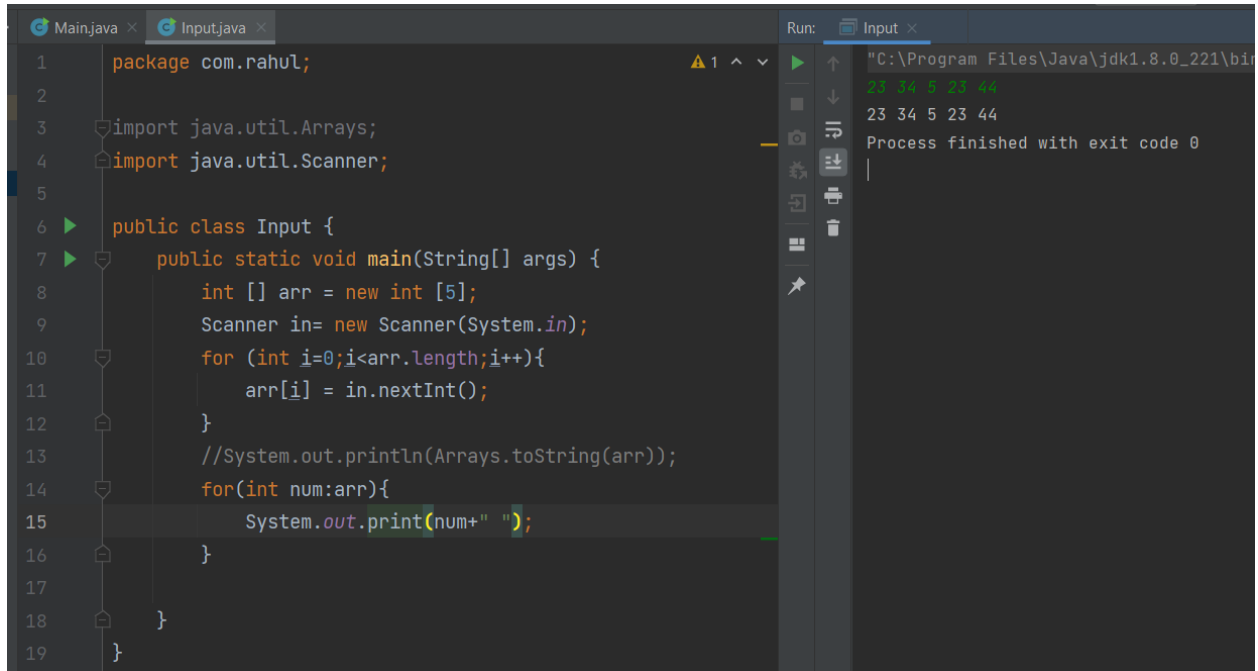
The screenshot shows an IDE window titled "Main.java x Input.java x". The code defines a class named "Input" with a static method "main". Inside "main", it creates an integer array "arr" of size 5, initializes it with values [3, 4, 5, 2, 5] using a loop and Scanner, and prints the array contents. The output console on the right displays the array "[3, 4, 5, 2, 5]" and confirms "Process finished with exit code 0".

```
package com.rahul;  
  
import java.util.Arrays;  
import java.util.Scanner;  
  
public class Input {  
    public static void main(String[] args) {  
        int [] arr = new int [5];  
        Scanner in= new Scanner(System.in);  
        for (int i=0;i<arr.length;i++){  
            arr[i] = in.nextInt();  
        }  
        System.out.println(Arrays.toString(arr));  
    }  
}
```

"C:\Program Files\Java\jdk1.8.0_221\bin\n..."
[3, 4, 5, 2, 5]
Process finished with exit code 0

FOR EACH LOOP

```
for(int num : arr) { // for every element in array, print the element
    System.out.print(num + " "); // here num represents element of the array
}
```

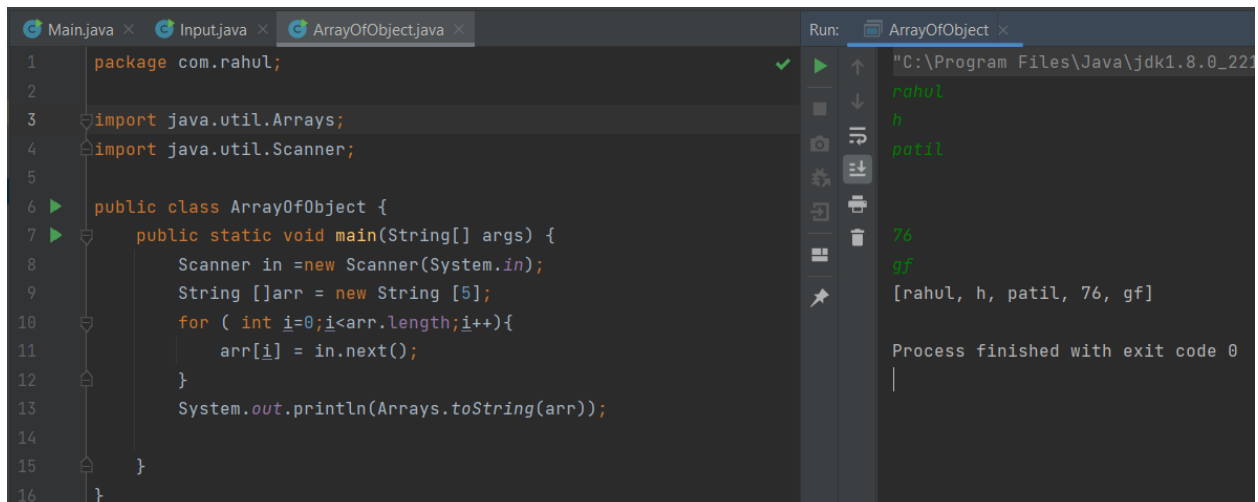


The screenshot shows an IDE with a file named `Input.java` open. The code defines a `public class Input` with a `main` method. Inside `main`, an integer array `arr` of size 5 is created. A `Scanner` object `in` is used to read five integers from the user. These integers are stored in `arr`. A `for` loop iterates over each element `num` in `arr`, printing it followed by a space. The output window shows the input sequence `23 34 5 23 44` and the resulting output `23 34 5 23 44`. The process finished with exit code 0.

```
1 package com.rahul;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class Input {
7     public static void main(String[] args) {
8         int [] arr = new int [5];
9         Scanner in= new Scanner(System.in);
10        for (int i=0;i<arr.length;i++){
11            arr[i] = in.nextInt();
12        }
13        //System.out.println(Arrays.toString(arr));
14        for(int num:arr){
15            System.out.print(num+" ");
16        }
17    }
18 }
19 }
```

Run: Input ×
"C:\Program Files\Java\jdk1.8.0_221\bin
23 34 5 23 44
23 34 5 23 44
Process finished with exit code 0

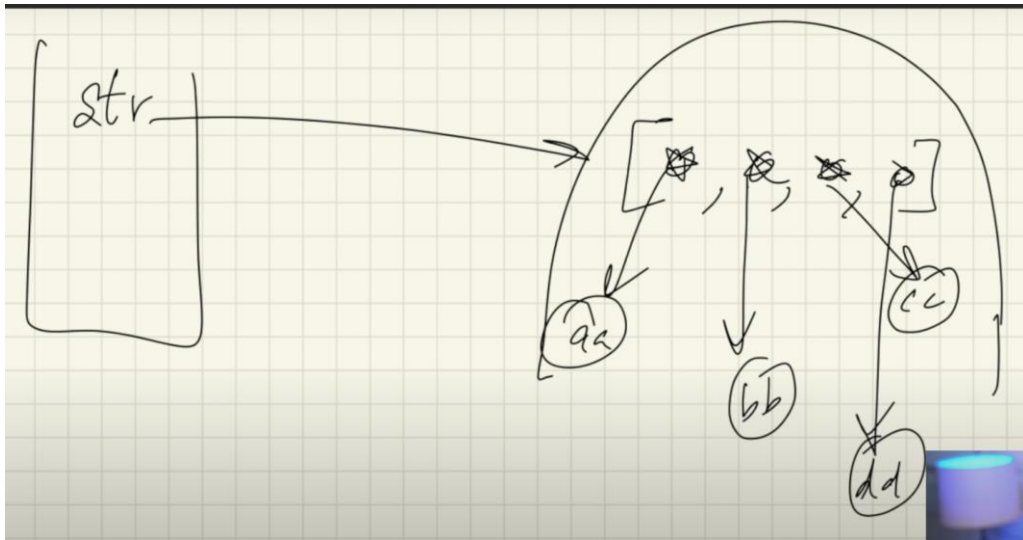
Array of objects



The screenshot shows an IDE with a file named `ArrayOfObject.java` open. The code defines a `public class ArrayOfObject` with a `main` method. Inside `main`, a `Scanner` object `in` is used to read five strings from the user. These strings are stored in a `String` array `arr`. The `Arrays.toString(arr)` method is used to print the entire array. The output window shows the input sequence `rahul h patil 76 gf` and the resulting output `[rahul, h, patil, 76, gf]`. The process finished with exit code 0.

```
1 package com.rahul;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class ArrayOfObject {
7     public static void main(String[] args) {
8         Scanner in =new Scanner(System.in);
9         String []arr = new String [5];
10        for ( int i=0;i<arr.length;i++){
11            arr[i] = in.next();
12        }
13        System.out.println(Arrays.toString(arr));
14    }
15 }
16 }
```

Run: ArrayOfObject ×
"C:\Program Files\Java\jdk1.8.0_221
rahul
h
patil
76
gf
[rahul, h, patil, 76, gf]
Process finished with exit code 0



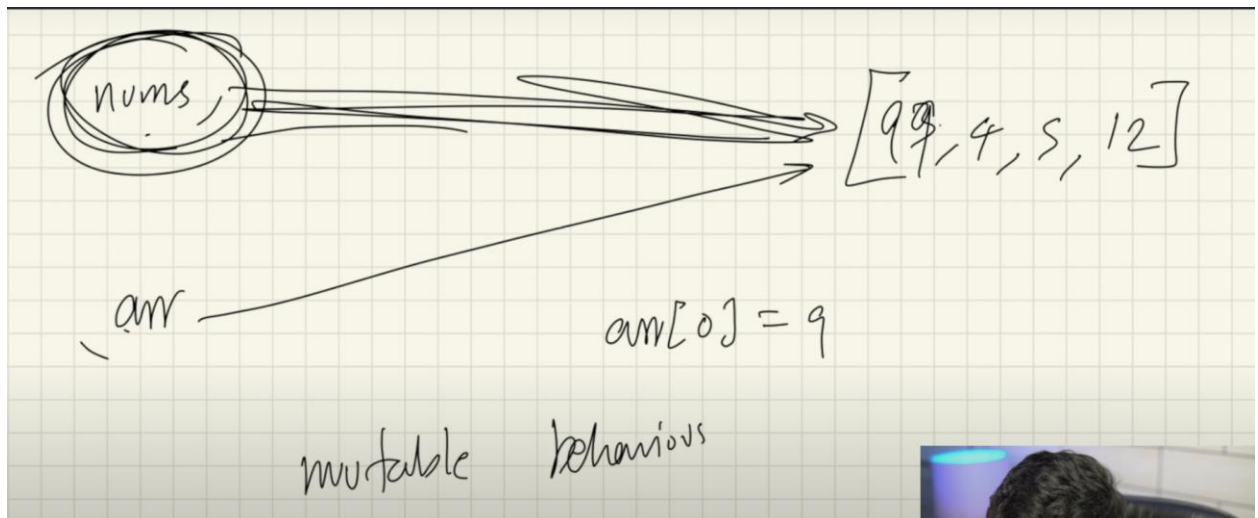
Array passing in function

```

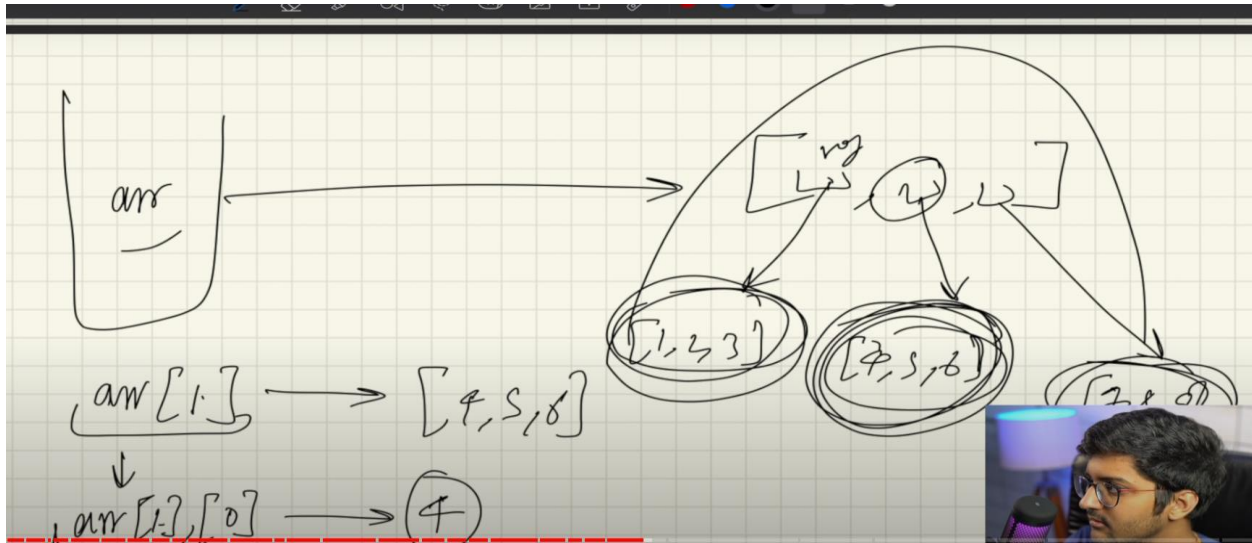
1  package com.rahul;
2  import java.util.Arrays;
3
4  public class ArrayPassing {
5      public static void main(String[] args) {
6          int [] arr = {1,2,3,4,5};
7          change(arr);
8          System.out.println(Arrays.toString(arr));
9      }
10     @ static void change(int[] arr){
11         arr[0] = 99;
12     }
13 }

```

Run: ArrayPassing
 "C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" -cp . ArrayPassing
 [99, 2, 3, 4, 5]
 Process finished with exit code 0



2 - Dimensional Array



```
int[][] arr = {  
    {1, 2, 3},  
    {4, 5},  
    {6, 7, 8, 9}}
```

```
main.java x Input.java x ArrayOfObject.java x ArrayPassing.java x D2Array.java x Run: D2Array x  
4 public class D2Array {  
5     public static void main(String[] args) {  
6         Scanner in = new Scanner(System.in);  
7         int [][] arr = new int [3][3];  
8         for(int row=0;row<arr.length;row++){  
9             for(int col = 0; col< arr[row].length; col++){  
10                arr[row][col]= in.nextInt();  
11            }  
12        }  
13        /*  
14        for(int row=0;row<arr.length;row++){  
15            for(int col = 0; col< arr[row].length; col++){  
16                System.out.println(arr[row][col]);  
17            }  
18        }  
19        for (int[] ints : arr) {  
20            for (int col = 0; col < ints.length; col++) {  
21                System.out.println(ints[col]);  
22            }  
23        }  
24        /*  
25        for (int[] ints : arr) {  
26            for (int anInt : ints) {  
27                System.out.print(anInt+" ");  
28            }  
29            System.out.println();  
30        }  
31    }  
32 }  
33 }  
34 }  
Run: D2Array x  
1 1 1  
2 2 2  
3 3 3  
Process finished with exit code 0
```

```
1 package com.rahul;
2
3 public class CollNoFixed {
4     public static void main(String[] args) {
5         int [][] arr={
6             {1,2,3,4},
7             {1,2,3},
8             {1,2,3,4,5}
9         };
10        for (int i=0;i<arr.length;i++){
11            for(int j=0;j<arr[i].length;j++){
12                System.out.print(arr[i][j]+" ");
13            }
14            System.out.println();
15        }
16    }
17 }
```

Run: CollNoFixed

```
"C:\Program Files\Java\jdk1.8.0_221\bin\java.
1 2 3 4
1 2 3
1 2 3 4 5

Process finished with exit code 0
```

Array List Syntax

```
3 import java.util.ArrayList;
4
5 public class ArrayListExample {
6     public static void main(String[] args) {
7         // Syntax
8         ArrayList<Integer> list = new ArrayList<>(initialCapacity:10);
9         |
10
11     }
12 }
```

```
1 package com.rahul;
2
3 import java.util.ArrayList;
4
5 public class ArrayListExample {
6     public static void main(String[] args) {
7         ArrayList<Integer> list = new ArrayList <>();
8         list.add(23);
9         list.add(234);
10        list.add(43);
11        list.add(143);
12        list.add(423);
13        System.out.println(list.contains(230)); //Check weather that element exist in the array
14        System.out.println(list);
15        list.set(0,99); // Sets the particular index with the given value
16        System.out.println(list);
17        list.remove(index: 2); //Removes the element at particular index
18        System.out.println(list);
19        for (int i=0;i< list.size();i++){
20            System.out.print(list.get(i)+" ", " "); //gives the element at particular index
21        }
22    }
23 }
24 }
```

Run: ArrayListExample

```
"C:\Program Files\Java\jdk1.8.0_221\b
false
[23, 234, 43, 143, 423]
[99, 234, 43, 143, 423]
[99, 234, 143, 423]
99, 234, 143, 423,
Process finished with exit code 0
```


How Array List size is maintained.

① Size is fixed Internally
② Say arraylist fills by some amount
⇒ It will a new arraylist of say, double the size
⇒ old elements are copied in new one
⇒ old one is deleted

$[1, 2, 9] \Rightarrow [1, 2, 9, 18, \dots]$
amortised $O(1)$

Array List follows amortized time complexity.

```
1 package com.raahul;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 public class ArrayListMultiD {
7     public static void main(String[] args) {
8         ArrayList<ArrayList<Integer>> list= new ArrayList<>();
9         Scanner in= new Scanner(System.in);
10        // initialisation
11        for(int i=0;i<3;i++){
12            list.add(new ArrayList<>());
13        }
14
15        //add elements
16        for(int i=0;i<3;i++){
17            for(int j=0;j<3;j++){
18                list.get(i).add(in.nextInt());
19            }
20        }
21        System.out.println(list);
22    }
23 }
24
```

Run: ArrayListMultiD
C:\Program Files\Java\jdk1.8.0_221\bin\java.exe
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Process finished with exit code 0

Q Swap

```
ArrayPassing.java x D2Array.java x CollNoFixed.java x ArrayListExample.java x ArrayListMultiD.java x QSwap.java x Run: QSwap x
1 package com.rahul;
2
3 import java.util.Arrays;
4
5 public class QSwap {
6     public static void main(String[] args) {
7         int [] arr = {1, 2, 3, 4, 5, 6, 7};
8         swap(arr, index1: 1, index2: 3);
9         System.out.println(Arrays.toString(arr));
10
11     }
12     @ static void swap(int [] arr, int index1, int index2){
13         int temp = arr[index1];
14         arr[index1] = arr[index2];
15         arr[index2] = temp;
16     }
17 }
```

"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" -cp "C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" com.rahul.QSwap

[1, 4, 3, 2, 5, 6, 7]

Process finished with exit code 0

QMax

```
main
Fixed.java x ArrayListExample.java x ArrayListMultiD.java x QSwap.java x QMaxNum.java x Run: QMaxNum x
1 package com.rahul;
2
3 public class QMaxNum {
4     public static void main(String[] args) {
5         int []arr = {23,1,44,5,23,55,34};
6         System.out.println(maxVal(arr));
7     }
8
9     @ static int maxVal(int[] arr) {
10         int maxNo = arr[0];
11         for ( int i = 0; i < arr.length; i++)
12         {
13             if (arr[i] > maxNo) {
14                 maxNo = arr[i];
15             }
16         }
17         return maxNo;
18     }
19 }
```

"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" -cp "C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" com.rahul.QMaxNum

55

Process finished with exit code 0

```
Fixed.java x ArrayListExample.java x ArrayListMultiD.java x QSwap.java x QMaxNum.java x Run: QMaxNum x
1 package com.rahul;
2
3 public class QMaxNum {
4     public static void main(String[] args) {
5         int []arr = {23,1,44,5,23,55,34};
6         System.out.println(maxVal(arr));
7         System.out.println(maxValRange(arr, start: 1, end: 4));
8     }
9
10     @ private static int maxValRange(int[] arr, int start, int end) {
11         int maxNo = arr[0];
12         for ( int i = start; i < end; i++)
13         {
14             if (arr[i] > maxNo) {
15                 maxNo = arr[i];
16             }
17         }
18         return maxNo;
19     }
20
21     @ static int maxVal(int[] arr) {
22         int maxNo = arr[0];
23         for ( int i = 0; i < arr.length; i++)
24         {
25             if (arr[i] > maxNo) {
26                 maxNo = arr[i];
27             }
28         }
29         return maxNo;
30     }
31 }
```

"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" -cp "C:\Program Files\Java\jdk1.8.0_221\bin\java.exe" com.rahul.QMaxNum

55

44

Process finished with exit code 0

Q Reverse

```
1 package com.rahu1;
2
3 import java.util.Arrays;
4
5 public class QReverse {
6     public static void main(String[] args) {
7         int [] arr ={1, 2,3,4,5,6,7};
8         reverseArray(arr);
9         System.out.println(Arrays.toString(arr));
10    }
11    @
12    private static void reverseArray(int[] arr) {
13        int start =0;
14        int end = arr.length-1;
15        while(start<end){
16            swap(arr,start,end);
17            start++;
18            end--;
19        }
20    @
21    static void swap(int [] arr, int index1, int index2){
22        int temp = arr[index1];
23        arr[index1]= arr[index2];
24        arr[index2]= temp;
25    }
26 }
```

Run: QReverse x

```
"C:\Program Files\Java\jdk1.8.0_221\bin\java.exe
[7, 6, 5, 4, 3, 2, 1]
Process finished with exit code 0
```

Leetcode Problems

Q Number of Good pairs

Description

Solution

Discuss (999+)

Submissions

Success Details >

Runtime: 1 ms, faster than 88.51% of Java online submissions for Number of Good Pairs.

Memory Usage: 41.6 MB, less than 40.04% of Java online submissions for Number of Good Pairs.

Next challenges:

Number of Pairs of Interchangeable Rectangles

Substrings That Begin and End With the Same Letter

Show off your acceptance:

f

t

in

Time Submitted	Status	Runtime	Memory	Language
06/30/2022 21:19	Accepted	1 ms	41.6 MB	java

Java

Autocomplete

```
1 class Solution {
2     public int numIdenticalPairs(int[] nums) {
3         Arrays.sort(nums);
4         int count =0;
5         int i=0;
6         for(int j=0;j<nums.length;j++){
7             if(nums[i]==nums[j]){
8                 count+=j-i;
9             }
10            else{
11                i=j;
12            }
13        }
14        return count;
15    }
16 }
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debugger

Accepted Runtime: 0 ms

Time Complexity : $O(n \log n)$

<https://leetcode.com/problems/number-of-good-pairs/submissions/>

