

ArrayLists

NOTE : Use Java8 oracle jdk

Akshay Goel

Amreshwar

Anil Kumar

Bhavesb Pandey

Bhushan Ganesh Shelar

Chandra Shekhar Bhatt

Deepika

Deepshikha Arora

Dev Raj Gaur

Dhanashree Sagane

Dhasthagiri Reddy

Dhruv

Farzana kauser

Harshad Marathe

Hemant Verma

Hemanth T

Induja

Iti

jeevanantham

Keshav Dutt Kaushik

M S Haseeb Khan

Mahesh

Manisha

Manohar A N

Mohona Paul

Naveen

Nikhil D

Nutan

Pranjul Kesharwani

Priyank

Pushpa

Rahul

Rishav Ghosh

Sai Kiran JNR

Saurabh

Shaik Lal Jan Basha

Shambhavi Belligatti

Shivam Shiv

Shivanand Patil

Srikant Kumar Pratihary

Ved Verma

Yoga Nand Sattvik

Q → QT

A → Private Chat

Hands always on keyboard

Q> You need to write a simple feature in instagram where you keep taking an integer input *rating* from the user until they enter a negative rating. Once a negative rating is entered, Print all the previous ratings in order.

————→ 1 5 10 2 4 6 9 -1
————→ 1 5 10 2 4 6 9

NOTE: The above cannot be solved via array ∵
The total no. of ratings is unknown.

```
public class Main {  
    public static void main(String[] args) {  
        // Keep taking input from the user till the user enters negative number  
        ArrayList<Integer> A = new ArrayList<>();  
        Scanner scanner = new Scanner(System.in);  
  
        while (true) {  
            int val = scanner.nextInt();  
            if (val < 0) {  
                break;  
            }  
            A.add(val);  
        }  
  
        for (int i = 0; i < A.size(); i++) {  
            System.out.print(A.get(i) + " ");  
        }  
    }  
}
```

Arrays

Disadvantages

- Fixed-size { cannot increase or decrease after creation }
- size should be known in advance.

ArrayList

ArrayList have all the advantages of arrays with some additional features.

- Dynamic size
- Doesn't require to know the size in advance.

Real world Examples

- Amazon Cart
- Tabs in browser
- Youtube Playlist

Syntax

ArrayList <Type> A = new ArrayList<>();

This type cannot be primitive data type.

int → Integer , long → Long , boolean → Boolean

Basic Operations

Add a **val** element to the end — `A.add(val)`

Get value at **ith** index — `A.get(i)`

Remove an element at **ith** index — `A.remove(i)`

Set a **val** at **ith** index — `A.set(i, val)`

```
public class Main {
    public static void main(String[] args) {
        // Create an ArrayList object
        ArrayList<Integer> A = new ArrayList<>();

        System.out.println("Before : " + A);

        // 10 20 30
        A.add(10);
        A.add(20);
        A.add(30);

        System.out.println("After : " + A);

        // Get item at 1st index
        // A[index] -> A.get(index)
        System.out.println("1st index has value : " + A.get(1));

        // set value at the first index to 40
        A.set(1, 40);
        System.out.println("Updating");
        System.out.println("1st index has value : " + A.get(1));
        System.out.println(A);

        // 0   1   2
        // [10, 40, 30]
        // Remove 1st index
        A.remove(index: 1);
        System.out.println("After removal : " + A);
    }
}
```

Taking ArrayList as input

NOTE: Donot forget to import java.util.ArrayList

```
public class Main {  
    public static void main(String[] args) {  
        // Create an ArrayList object  
        ArrayList<Integer> A = new ArrayList<>();  
        Scanner scanner = new Scanner(System.in);  
  
        int n = scanner.nextInt();  
  
        for (int i = 0; i < n; i++) {  
            int val = scanner.nextInt();  
            A.add(val);  
        }  
  
        // sout(A)  
        for (int i = 0; i < A.size(); i++) {  
            System.out.print(A.get(i) + " ");  
        }  
    }  
}
```

Q> Given **ArrayList** as input, return an **ArrayList** of the multiples of 5 or 7

I/p : [1, 5, 3, 0, 7]
O/p → [5, 0, 7]

```
public class Main {  
    static ArrayList<Integer> solve(ArrayList<Integer> A) {  
        ArrayList<Integer> B = new ArrayList<>();  
  
        for (int i = 0; i < A.size(); i++) {  
            int val = A.get(i);  
            if (val % 5 == 0 || val % 7 == 0) {  
                B.add(val);  
            }  
        }  
  
        return B;  
    }  
  
    public static void main(String[] args) {  
        ArrayList<Integer> A = new ArrayList<>();  
        // 1 5 3 0 7  
        A.add(1);  
        A.add(5);  
        A.add(3);  
        A.add(0);  
        A.add(7);  
  
        System.out.println(solve(A));  
    }  
}
```

Q> Given two integer A, B as input.
Return an `ArrayList` containing first B multiples of A .

$A = 2$ $B = 4$

Output $\longrightarrow [2, 4, 6, 8]$

first 4 multiples in the table of 2

```
public class Main {  
    static ArrayList<Integer> solve(int A, int B) {  
        ArrayList<Integer> multiples = new ArrayList<>();  
        for (int i = 0; i < B; i++) { // B times  
            multiples.add(A * (i + 1));  
        }  
        return multiples;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(solve(A: 2, B: 4));  
    }  
}
```

2D ArrayList

2D Array \longrightarrow Array of Arrays

2D ArrayList \longrightarrow ArrayList of ArrayLists

Syntax

`ArrayList<ArrayList<Type>> A = new ArrayList<>();`

Let's create 2D ArrayList as follows

```
-1  4  3
  5  2
  9  2  4  1  6
```

```
public class Main {
    public static void main(String[] args) {
        // ArrayList containing -1 4 3
        // ArrayList of ArrayList A1 A2 A3
        /*
        -1 4 3 // A1
        5 2 // A2
        9 2 4 1 6 // A3
        */
        ArrayList<ArrayList<Integer>> list2D = new ArrayList<>();
        ArrayList<Integer> A1 = new ArrayList<>();
        A1.add(-1);
        A1.add(4);
        A1.add(3);

        ArrayList<Integer> A2 = new ArrayList<>(Arrays.asList(5, 2));

        ArrayList<Integer> A3 = new ArrayList<>();
        A3.add(9);
        A3.add(2);
        A3.add(4);
        A3.add(1);
        A3.add(6);

        list2D.add(A1);
        list2D.add(A2);
        list2D.add(A3);

        System.out.println(list2D);
    }
}
```



```
public class Main {  
    public static void main(String[] args) {  
        // ArrayList containing -1 4 3  
        // ArrayList of ArrayList A1 A2 A3  
        /*  
        -1 4 3 // A1  
        5 2 // A2  
        9 2 4 1 6 // A3  
        */  
        Scanner scanner = new Scanner(System.in);  
        ArrayList<ArrayList<Integer>> list2D = new ArrayList<>();  
        System.out.print("No. of rows -> ");  
        int R = scanner.nextInt();  
        // R is the number of rows to be added in the list2D  
        for (int r = 0; r < R; r++) {  
            System.out.print("No. of cols in " + r + " row -> ");  
            int C = scanner.nextInt();  
            ArrayList<Integer> row = new ArrayList<>();  
            for (int c = 0; c < C; c++) {  
                int val = scanner.nextInt();  
                row.add(val);  
            }  
            list2D.add(row);  
        }  
        System.out.println(list2D);  
    }  
}
```

Q> Given 2D ArrayList as input, print it line by line

I/p : $[[1, 4], [0], [10, -5, 1]]$

Output

```
1 4
0
10 -5 1
```

```
static void print2DArrayList(ArrayList<ArrayList<Integer>> list2D) {
    int R = list2D.size();
    // int C = list2D.size(); Do not do this

    for (int r = 0; r < R; r++) {
        // for(int c = 0; c < A[r].length; c++)
        for (int c = 0; c < list2D.get(r).size(); c++) {
            // print(A[r][c])
            System.out.print(list2D.get(r).get(c) + " ");
        }
        System.out.println();
    }
}
```

Q> Given N as input , return numeric staircase as AL

I/p : 3

O/p : [[1] ,
 [1,2] ,
 [1,2,3]]

Try till 23:20

[[1] , → row 1 create array list [1...1]
 [1,2] , → row 2 [1...2]
 [1,2,3]] → row 3 [1...3]

```
public class Main {  
    public static void main(String[] args) {  
        int N = 3;  
        ArrayList<ArrayList<Integer>> list2D = new ArrayList<>();  
        // [1] -> r = 1  
        // [1, 2] -> r = 2  
        // [1, 2, 3] -> r = 3  
        for (int r = 1; r <= N; r++) { // N rows for the AL  
            ArrayList<Integer> row = new ArrayList<>();  
            for (int val = 1; val <= r; val++) {  
                row.add(val);  
            }  
            list2D.add(row);  
        }  
  
        System.out.println(list2D);  
    }  
}
```

Reminder

we language — Java 8 Oracle JDK