**ArrayList**

**Duplicate allow and un order(depend on index)**

List<Integer> h = **new** ArrayList<>();

h.add(2);

h.add(4);

h.add(1);

h.add(3);

System.***out***.println(h);

Ans:

**[2, 4, 1, 3]**

**public** **class** Main {

**public** **static** **void** main(String[] args) {

List<Integer> list = **new** ArrayList<>();

list.add(3);

list.add(4);

list.add(2);

list.add(1);

System.***out***.println(list); // [3, 4, 2, 1]

Set<Integer> s = **new** HashSet<>(list);

System.***out***.println(s); // [1, 2, 3, 4]

String result = s.stream()

.map(temp -> String.*valueOf*(temp))

.collect(Collectors.*joining*(","));

System.***out***.println(result); // 1,2,3,4

Iterator<Integer> it = s.iterator();

**while** (it.hasNext()) {

Integer temp = it.next();

System.***out***.print(temp); // 1234

}

}

**import** java.util.ArrayList;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

ArrayList<String> list = **new** ArrayList<>();

list.add("apple");

list.add("banana");

list.add("watermelon");

list.add("pear");

list.add("apple");

**for**(String temp: list) {

System.***out***.print(temp + " "); // apple banana watermelon pear apple

}

System.***out***.println(); // new line

System.***out***.println(list.contains("banana")); // true

System.***out***.println(list.size()); // 5

System.***out***.println(list.get(1)); // banana

System.***out***.println(list.remove(0)); // apple

System.***out***.println(list); // [banana, watermelon, pear, apple]

list.remove("pear");

System.***out***.println(list); // [banana, watermelon, apple]

}

}

**Interview**

1. how to avoid duplicate in ArrayList
2. **import** java.util.ArrayList;
3. **public** **class** Main {
4. **public** **static** **void** main(String[] args) {

7. ArrayList<String> list = **new** ArrayList<>(); // list Allows Duplicate
9. list.add("apple");
10. list.add("banana");
11. list.add("watermelon");
12. list.add("pear");
13. list.add("apple");
15. **for**(String tempList : list) {
16. System.***out***.print(tempList + " "); // apple banana watermelon pear apple
17. }
19. System.***out***.println();
21. ArrayList<String> newList = **new** ArrayList<>(); //list Not Allows Duplicate
23. **for** (String temp : list) {
24. **if**(!newList.contains(temp)) {
25. newList.add(temp);
26. System.***out***.print(temp + " "); // apple banana watermelon pear
27. }
28. }

31. }
32. }

**1. How to remove duplicates from ArrayList in Java?**

List<Integer> list = **new** ArrayList<>();

list.add(2);

list.add(3);

list.add(4);

list.add(4);

list.add(5);

list.add(6);

list.add(6);

list.add(7);

System.***out***.println(list);

Set<Integer> listSet = **new** LinkedHashSet<>(list);

list.clear();

list.addAll(listSet);

System.***out***.println(list);

**2. How to reverse ArrayList in Java?**

List<Integer> list = **new** ArrayList<>();

list.add(1);

list.add(2);

list.add(3);

list.add(4);

list.add(5);

// Collections.reverse(list); // first way

// System.out.println(list);

**for**(**int** i=list.size()-1; i >= 0 ; i--) { // second way

System.***out***.print(list.get(i) + " ");

}

**3. Difference between an array and an ArrayList in Java?  
  
Array:**

**a-** fixed sized 🡪 int arr[] = new int[10]

b- you can store primitives in Array

**ArrayList:**

1. Dynamic sized 🡪 ArrayList<Type> arrL = new ArrayList<Type>();
2. you can not store primitives in ArrayList

**4. How to synchronize ArrayList in Java?**

List<String> listNonS = **new** ArrayList<String>();

List<String> list = Collections.*synchronizedList*(listNonS);

list.add("ninos");

list.add("nahrain");

list.add("matthew");

list.add("daniel");

**synchronized**(list)

{

// must be in synchronized block

Iterator it = list.iterator();

**while** (it.hasNext())

System.***out***.println(it.next());

}

**5. When to use ArrayList and LinkedList in Java?**  
  
 **What is common between ArrayList and LinkedList in Java**  
 1) Both ArrayList and LinkedList are an implementation of List interface.

2) Both ArrayList and LinkedList are not synchronized.

3) ArrayList and LinkedList are ordered collection.

4) ArrayList and LinkedList also allow [duplicates](http://javarevisited.blogspot.sg/2012/12/how-to-remove-duplicates-elements-from-ArrayList-Java.html) and null.

**Difference between LinkedList and ArrayList in Java**

1)ArrayList is backed by Array while LinkedList is backed by LinkedList.  
2) LinkedList also implements the Deque interface, which provides first in first out operations for add() and poll().

## 6. Difference between ArrayList and HashSet in Java?

## Similarities ArrayList and HashSet

## Both ArrayList and HashSet are non synchronized.

## 2) Both ArrayList and HashSet can be traversed using Iterator.

## Difference between ArrayList vs HashSet in Java

## 1) ArrayList implements List interface while HashSet implements Set interface. 2) ArrayList allow duplicate HashSet doesn’t allow duplicate.

## 3) ArrayList is an ordered collection while HashSet is an unordered collection

**7. How to loop over ArrayList in Java?**

**ArrayList**<**String**> games = **new** **ArrayList**<**String**>();  
        games.add("Cricket");  
        games.add("Soccer");  
        games.add("Hockey");  
        games.add("Chess");  
**1- for**(**String** game: games){   
            **System**.out.println(game);  
        }

2- **for**(**int** i =0; i<games.size(); i++){  
            **String** game = games.get(i);  
        }

3- **Iterator**<**String**> itr = games.iterator();  
        **while**(itr.hasNext()){  
            **System**.out.println( itr.next() );

        }

**8. How to create and initialize ArrayList in one line?**

List<String> coolStringList = Arrays.asList("Java", "Scala", "Groovy");

**9. How to sort ArrayList in Java?**

### **Sorting ArrayList in Ascending Order in Java**

List<Integer> listofYears = **new** ArrayList<Integer>();

listofYears.add(2021);

listofYears.add(2019);

listofYears.add(2018);

listofYears.add(2020);

Collections.*sort*(listofYears);

System.***out***.println(listofYears);

### **Sorting ArrayList in Descending Order in Java**

List<Integer> listofYears = **new** ArrayList<Integer>();

listofYears.add(2021);

listofYears.add(2019);

listofYears.add(2018);

listofYears.add(2020);

Collections.*sort*(listofYears, Collections.*reverseOrder*());

System.***out***.println(listofYears);

**10. Difference between HashMap and ArrayList in Java?**

## Similarity between ArrayList and HashMap in Java

1. Both ArrayList and HashMap are not synchronized, you should not use them in the multi-threading environment without external  synchronization.
2. Both ArrayList and HashMap allows null.
3. ArrayList allows duplicate elements and HashMap allow duplicate values.

## Difference between ArrayList vs HashMap in Java

1) ArrayList implements List interface while HashMap implements Map interface in Java.

2) ArrayList only stores one object while HashMap stores two objects key and value.  
3) ArrayList maintains the order of object, in which they are inserted while HashMap doesn't provide any ordering guarantee.

4) ArrayList allows duplicates but HashMap doesn't allow duplicates key though it allows [duplicate values](http://java67.blogspot.com/2015/03/how-to-remove-duplicates-from-arraylist.html).

**11. How to use ArrayList in Java?**

1) When you need to maintain insertion order of elements  
2) You want fastest access of element by index. get(index) return object from ArrayList  
3) You don't mind duplicates.

ArrayList<String> list = **new** ArrayList<>();

list.add("ninos");

list.add("nahrain");

list.add("matthew");

list.add("daniel");

System.***out***.println("size of arrayList : "+ list.size()); // size of arrayList : 4

System.***out***.println("get index by name : "+ list.indexOf("matthew")); // get index by name : 2

System.***out***.println("get name by index : "+ list.get(2)); // get name by index : matthew

System.***out***.println("remove last name : "+ list.remove(list.size() - 1)); // remove last name: daniel

System.***out***.println("return boolean contain : "+list.contains("ninos") ); // return boolean contain : true

list.clear();

System.***out***.println("arrayList it's clear now : "+ list); // arrayList it's clear now : []

12. How to get a sublist from ArrayList in Java?  
  
ArrayList<String> arrayList = **new** ArrayList<String>();

//Add elements to Arraylist

arrayList.add("ninos");

arrayList.add("nahrain");

arrayList.add("matthew");

arrayList.add("daniel");

arrayList.add("wilson");

arrayList.add("marlin");

/\*

**subList Method returns sublist from list with starting index to end index-1**

\*/

List list = arrayList.subList(1,3);

//display elements of sub list.

System.***out***.println("Sub list contains : ");

**for**(**int** i=0; i < list.size(); i++) {

System.***out***.println(list.get(i));

}

Answer:

Sub list contains :

Nahrain // index 1

Matthew // index 3 - 1