

## 1.What are the main differences between array and collection?

ARRAY	COLLECTION
Arrays are fixed in size.	Collection are growable in nature.
Arrays can hold homogenous data.	Collection can hold both homogeneous and heterogeneous data.
There is no underlying data structure.	Every collection class is implemented based on some standard data structure.
Because of no underlying data structure there is no predefined methods	Collection has lots of predefined methods
It can hold both object and primitive.	It holds object types only.

## 2.Explain various interfaces used in Collection framework?

Java Collection framework provides many interfaces **Set**, **List**, **Queue**, **Deque**.

## 3.What is the difference between ArrayList and Vector?

ARRAY LIST	VECTOR
ArrayList's methods are not synchronized	Vector's methods are synchronized
It increases size by 50%	It increases size by 100
It is introduced in version 1.2	It is legacy class ,introduced in version 1.0
Underlying data structure is an array.	Underlying data structure is an array.

#### 4.What is the difference between ArrayList and LinkedList?

ARRAYLIST	LINKEDLIST
Underlying data structure is a growable array.	Underlying data structure is a doubly linked list.
It acts as a list only.	It acts as a list and queue.
It is better for sorting and accessing data.	It is better for insertion and deletion of data.

#### 5.What is the difference between Iterator and ListIterator?

ITERATOR	LIST ITERATOR
Iterator can traverse only in forward direction	ListIterator traverses both in forward and backward directions.
Iterator cannot help to replace an element	ListIterator can help to replace an element
It is used throughout the collection interface.	It is used only in the list interface.

#### 6.What is the difference between List and Set?

LIST	SET
The list implementation allows us to add the same or duplicate elements.	The set implementation doesn't allow us to add the same or duplicate elements.

The insertion order is maintained by the List.	It doesn't maintain the insertion order of elements.
List allows us to add any number of null values.	Set allows us to add at least one null value in it.
The List implementation classes are LinkedList and ArrayList.	The Set implementation classes are TreeSet, HashSet and LinkedHashSet.
We can get the element of a specified index from the list using the get() method.	We cannot find the element from the Set based on the index because it doesn't provide any get method().
It is used when we want to frequently access the elements by using the index	It is used when we want to design a collection of distinct elements.
The method of List interface listiterator() is used to iterate the List elements	The iterator is used when we need to iterate the Set elements.

## 7. What is the difference between HashSet and TreeSet?

<b>HASH SET</b>	<b>TREE SET</b>
Hash set is implemented using HashTable.	The tree set is implemented using a tree structure.
HashSet allows a null object .	The tree set does not allow the null object. It throws the null pointer exception.
Hash set uses an equals method to compare two objects.	Tree sets use a compare method for comparing two objects.
Hash set doesn't now allow a heterogeneous object .	Tree set allows a heterogeneous object .
HashSet does not maintain any order .	TreeSet maintains an object in sorted order .

## 8. What is the difference between HashSet and HashMap?

HASH MAP	HASH SET
Java HashMap is a hash table based implementation of Map interface.	HashSet is a Set. It creates a collection that uses a hash table for storage.
HashMap implements <b>Map, Cloneable, and Serializable</b> interfaces.	HashSet implements <b>Set, Cloneable, Serializable, Iterable</b> and <b>Collection</b> interfaces.
In HashMap we store a <b>key-value pair</b> . It maintains the mapping of key and value.	In HashSet, we store <b>objects</b> .
It does not allow <b>duplicate keys</b> , but <b>duplicate values</b> are <b>allowed</b> .	It does not allow <b>duplicate values</b> .
It can contain a <b>single null key</b> and <b>multiple null values</b> .	It can contain a <b>single null value</b>
HashMap uses the <b>put()</b> method to add the elements in the HashMap	HashSet uses the <b>add()</b> method to add elements in the HashSet.
HashMap is <b>faster/</b> than HashSet <b>because values are associated with a unique key</b> .	HashSet is <b>slower</b> than HashMap because the member object is used for calculating hashcode value, which can be same for two objects.
Only <b>one</b> object is created during the add operation.	There are <b>two</b> objects created during put operation, one for <b>key</b> and one for <b>value</b> .
Always prefer when we do not maintain the <b>uniqueness</b> .	It is used when we need to maintain the <b>uniqueness</b> of data.

## 9.What is the difference between HashMap andHashtable?

HASH MAP	HASH TABLE
HashMap is unsynchronized	Hashtable is synchronized
It is not thread-safe and can't be shared between many threads without proper synchronization code	It is thread-safe and can be shared with many threads.
HashMap allows one null key and multiple null values	Hashtable doesn't allow any null key or value.
HashMap is generally preferred over Hashtable if thread synchronization is not needed	Hashtable is generally preferred over Hashmap if thread synchronization is needed

## 10. What is the difference between Comparable and Comparator?

COMPARABLE	COMPARATOR
Comparable provides a single sorting sequence. In other words, we can sort the collection on the basis of a single element such as id, name, and price.	The Comparator provides multiple sorting sequences. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc.
Comparable affects the original class, i.e., the actual class is modified	Comparator doesn't affect the original class, i.e., the actual class is not modified.
Comparable provides compareTo() method to sort elements.	Comparator provides compare() method to sort elements.

Comparable is present in java.lang package.	A Comparator is present in the java.util package.
We can sort the list elements of Comparable type by Collections.sort(List) method.	We can sort the list elements of Comparator type by Collections.sort(List, Comparator) method.

## 11. How to synchronize List, Set and Map elements?

ArrayList, HashSet and HashMap are three most frequently used data structures in java. As they are most used, they are not synchronized for the sake of performance. But, java provides the methods to make them synchronized as and when the need arises.

These methods are introduced in java.util.Collections class. Collections class is an utility class which has some useful methods helpful for operations on collection types. In this post, we will see how to synchronize ArrayList, HashSet and HashMap in java.

## 12. What do you understand by fail-fast?

Fail fast is a philosophy that values extensive testing and incremental development to determine whether an idea has value. ... Failing fast seeks to take the stigma out of the word "failure" by emphasizing that the knowledge gained from a failed attempt actually increases the probability of an eventual success.

## 13.What is the difference between Array and ArrayList?

ARRAY	ARRAY LIST
Array is static in size.	ArrayList is dynamic in size.
An array is a fixed-length data structure.	ArrayList is a variable-length data structure. It can be resized itself when

	needed.
It is mandatory to provide the size of an array while initializing it directly or indirectly.	We can create an instance of ArrayList without specifying its size. Java creates ArrayList of default size.
It performs fast in comparison to ArrayList because of fixed size.	ArrayList is internally backed by the array in Java. The resize operation in ArrayList slows down the performance.
An array can store both objects and primitives type.	We cannot store primitive type in ArrayList. It automatically converts primitive type to object.
We use for loop or for each loop to iterate over an array.	We use an iterator to iterate over ArrayList.
We cannot use generics along with array because it is not a convertible type of array.	ArrayList allows us to store only generic/ type, that's why it is type-safe.
Array provides a length variable which denotes the length of an array.	ArrayList provides the size() method to determine the size of ArrayList.
We can add elements in an array by using the assignment operator.	Java provides the add() method to add elements in the ArrayList.
Array can be multi-dimensional.	ArrayList is always single-dimensional.

## 14. How to remove duplicates from ArrayList?

To remove the duplicates from a list, you can make use of the built-in function `set()`. The specialty of the `set()` method is that it returns distinct elements. You can remove duplicates from the given list by importing `OrderedDict` from `collections`.