

Q) What are the main difference b/w array and collection?

Ans)

### Array

a) Array itself a data structure and has some restrictions for entering values.

b) Arrays are not growable

c) Arrays elements cannot be removed.

d) Arrays doesn't allow null values

### Collection

a) Collections has various data structure available providing freedom to users for manipulation of object.

b) Collections are growable

c) In collections elements can be removed and modified.

d) Collection allow null values.

Q) Explain various interfaces used in Collection framework?

Ans) Interfaces present in collection frameworks are :-

a) List

b) Queue

c) Set.

List :-

- The child of List Interface are
- \* ArrayList
  - \* Vector
  - \* LinkedList
- List is adopted when
- a) Insertion order is of higher priority
  - b) when inserting or deleting the elements wanted to be feasible.
  - c) when sorting is of less priority.

Set :-

- The child of Set are
- \* HashSet
  - \* LinkedHashSet
  - \* TreeSet
- Set is adopted when
- a) Duplicate value is to remove
  - b) when we don't want insertion order.
  - c) when we don't to do sorting.

Queue :-

- The child of Queue Interface
- \* priority Queue.

Q3.) What is the Difference b/w ArrayList & Vector?

Ans.) ArrayList

Vector

- |   |   |
|---|---|
| 1.) ArrayList comes from non-legacy class i.e., from version 1.2.   | 1.) Vector comes from legacy class i.e., from version 1.0   |
| 2.) ArrayList is not synchronized   | 2.) Vector is synchronized                                  |
| 3.) It is thread safe.  | 3.) It is thread safe, but not fast.                        |
| 4.) Initially the new capacity is meant to be $\rightarrow (\frac{1}{2} * \text{old capacity}) + \text{later changed to } 50\% \text{ (i.e., } 0.5\% + \text{load factor})$ | 4.) The capacity is $(2 * \text{old capacity})$ i.e., 200%. |

Q4.) What is difference b/w ArrayList & LinkedList.

Ans.)

ArrayList

LinkedList

- a.) Underline D.S. is growable or resizable array.

- a.) Underline D.S. is Doubly linked list.

- |   |  |
|---|--|
| a) Best suited operation if for data retrieval and insertion order given higher priority. | b) Best suited operation is for insertion & deletion feasibility.    |
| c) It implements the interfaces of serializable & cloneable & random access.              | d) The secondary interface of linkedlist is serializable, cloneable. |

Q5.) What is the diff b/w Iterator & ListIterator?

Ans.)

### Iterator

a) Iterator is the universal class that helps to fetch the data (among the otherthing)

b) Iterator can only point forward

c) It points from first to last.

### ListIterator

a) ListIterator is the interface that extends Iterator.

b) ListIterator can point backward & forward.

c) ListIterator can point for any specified location.

Q.1) What is the difference b/w List & Set?

Ans.)

### List

- The List implementation allows us to add the same or duplicate elements.
- The insertion order is maintained by the list.
- List allows us to add any number of null values.
- The List implementation classes are LinkedList and arrayList.
- We can get the elements of a specified index from the list using the get() method.

### Set

- The set implementation doesn't allow us to add the same or duplicate elements.
- It doesn't maintain the insertion order of elements.
- The set allows us to add at least one null value in it.
- The set implementation classes are TreeSet & linkedHashSet.
- We cannot find the element from the set based on the index because it doesn't provide any get method().

Q7) What is the Difference b/w HashSet & TreeSet.

### HashSet

1) It does not provide a guarantee to sort the data.

2) In HashSet only one element can be null.

3) It uses hashCode() or equals() method for comparison.

4) It is faster than TreeSet.

5) Underline D.S. is hashtable

6) It allows only heterogeneous values.

### TreeSet

1) It provides a guarantee to sort the data. Sorting depends on the supplied comparator.

2) It does not allow null elements.

3) It uses compare() or compareTo() method for comparison.

4) It delegates to HashSet for comparison to TreeSet.

5) Underline D.S. is binary tree.

6) It allows only homogeneous values.

Q8.) What is the difference b/w HashSet & HashMap.

Ans)

### HashSet

### HashMap

- |   |   |
|---|---|
| a) HashSet is a set.<br>It creates a collection that uses a hash table for storage. | a.) HashMap is a hash table based implementation of map interface.      |
| b.) Secondary Interface is Serializable, cloneable, Iterable, collection.           | b.) Secondary Interface is Serializable and cloneable.                  |
| c.) It doesn't allow duplicate null values.   | c.) It does not allow duplicate keys, but duplicate values are allowed. |
| d.) It can contain a single null value.   | d.) It allows a single null value and multiple null values.             |
| e.) It uses add() method to add elements in the HashSet.                            | e.) HashMap uses the put() method to add the elements in the HashMap.   |

Q2) Difference b/w HashMap & HashTable ?

### HashMap

- 1) HashMap is non-synchronized. It is not thread-safe and can't be shared between many threads without proper synchronization code.

### HashTable

- 1) HashTable is synchronized. It is thread-safe and can be shared with many threads.

- 2) HashMap allows one null key and multiple null values.
- 2) HashTable doesn't allow any null key or value.

- 3) HashMap is a new class introduced in JDK 1.2.

- 3) HashTable is legacy class.

- 4) HashMap is fast

- 4) HashTable is slow

- 5) Iterator in HashMap is fail-fast.

- 5) Enumerator in HashTable is not fail-fast.

Q1) Difference b/w Comparable & Comparator?

Ans)

### Comparable

- i) Comparable provides a single sorting sequence. In other words we can sort the collection on the basis of single elements such as ID, name and price.

### Comparator

- ii) The comparator provides multiple sorting sequences. In other words. We can sort the collection on the basis of multiple elements such as ID, name & price, etc.

- iii) Comparable affects the original class . i.e. the actual class is modified.

- iii) Comparator doesn't affect the original class . i.e., the actual class is not modified.

- iv) Comparable provide compareTo() method to sort elements.

- iv) Comparator provide compare() method to sort elements.

- v) Comparable is present in java.lang package.

- v) A comparator is present in java.util package.

Q2) How to synchronize list, Set, Map elements.

- Ans)
- i) Inorder to get a synchronized list from ArrayList. we use synchronized List<T> method.
  - ii) Collections.synchronizedSet() method is used to synchronize Set elements.

iii) Collection synchronized map (HashMap) method `ReentrantMap`  
used to synchronize map elements.

Ques) What do you understand by fail-fast & fail-safe?

Ans) In Java, collections supports two types of iterators they are.

- i) Fail fast } These are very useful in exception handling.
- ii) fail-safe }

Fail-fast :-

The fail-fast iterator aborts the operation as soon it exposes failure and stops the entire operation comparatively.

Fail-safe :-

The fail-safe iterator does not abort the operation in case of failure. Instead it tries to avoid failures as much as possible.

Ques) Diff. b/w Array & ArrayList?

Ans) Array

- i) Array is dynamically created
- ii) Array is static in size
- iii) Array is fixed length data structure.
- iv) It's mandatory to provide size of an array why it's follows it directly or indirectly.

ArrayList

- i) The ArrayList is a class of Java collection framework.
- ii) ArrayList is dynamic in size
- iii) ArrayList is variable length
- iv) We can create an instance of ArrayList without specifying its size.