## FAQ's on Collections

### 1. What the limitations are of Object arrays?

The limitations of Object arrays are that:

- They are fixed in size. Once an array object is created, there is no chance of increasing or decreasing the size based on the requirement. Hence, If we don't know size in advance, arrays are not recommended.

- They can hold only homogeneous elements.

- There is no underlying data structure for arrays and hence no readymade method support for arrays. So, for every requirement, programmer has to explicitly code the relevant method.

To overcome these problems collections are recommended.

### 2. State the differences between arrays and collections.

| Arrays | Collections |
|---|---|
| 1. Arrays are fixed in size. We are not allowed to increase or decrease the array size, once it is created. | 1. Collections are growable in nature and hence based on the requirements. We can increase or decrease the size. |
| 2. From the memory point of view, use of arrays is not recommended. | 2. From the memory point of view, collections are recommended. |
| 3. From the performance point of view, use of arrays is not recommended. | 3. From the performance point of view, use of collections is recommended. |
| 4. Arrays can hold only homogeneous elements. | 4. Collections can hold both homogeneous and heterogeneous elements. |
| 5. Arrays can hold both primitives as well as objects. | 5. Collections can hold only objects. |
| 6. For any requirement, there is no supported readymade method available. Programmer has to compulsory code it explicitly. | 6. For every requirement readymade method support is available. Being a programmer we have to know how to use those methods and we are not responsible to implement them. |

**3.    What are differences between arrays and ArrayList?**

Refer to the answer of  Q2

**4.    What are differences between arrays and Vector?**

Refer to the answer of  Q2

**5.    What is a Collection API?**

The Collection API defines a set of classes and interfaces which can be used for representing a group of objects as single entity.

**6.    What is a Collection framework?**

The Collection Framework defines a set of classes and interfaces which can be used for representing a group of objects as single entity.

**7.    State the difference between Collections and Collection?**

**Collection** is an interface which can be used for representing a group of individual objects as single entity and it acts as root interface of collection frame work.

**Collections** are a utility class to define several utility methods for Collection implemented class objects.

**8.    Explain about Collection interface.**

The **Collection Interface** can be used to represent a group of objects as a single entity:

- It acts as root interface for entire collection framework.
- It defines the most commonly used methods which can be applicable for any collection implemented class object

**9.    Explain about List interface.**

The List interface is a child interface of Collection interface. This can be used to represent group of individual objects as a single entity where:
- Duplicates are allowed and
- Insertion order is preserved.

### 10. Explain about Set interface.

Set interface is a child interface of Collection interface. It can be used to represent a group of individual objects as a single entity where:

- Duplicate objects are not allowed and
- Insertion order is not preserved.

### 11. Explain about SortedSet interface.

It is child interface of Set interface. It can be used to represent a group of individual objects into a single entity where:

- All the objects are arranged in some sorting order (Can be natural sorting order or customized) and
- Duplicates are not allowed.

### 12. Explain about NavigableSet.

It is child interface of SortedSet and provides several utility methods for navigation purposes:

- It doesn't allows duplicates,
- Insertion order is preserved and
- It is introduced in 1.6 version.

### 13. Explain about Queue interface.

If we want to represent a group of individual objects prior to processing, then we should go for Queue interface. It is child interface of Collection interface. It has been introduced in 1.5 version.

### 14. Explain about Map interface.

Remember it is not a child Interface of Collection Interface and hence Map and Collection Interfaces doesn't have any relationship.

- It can be used for representing a group of Objects as key, value pairs.
- Both keys and values should be objects.
- Keys can t be duplicated but values can be duplicated.
- It has been introduced in 1.2 version.

### 15. Explain about SortedMap.

If we want to represent a group of objects as key value pairs, where all the entries are arranged according some sorting order of keys then we should go for SortedMap. It is child interface of Map and it has been introduced in 1.2 version.

### 16. Explain about NavigableMap.

It is child interface of SortedMap and defines several methods for navigation purpose. It has been introduced in 1.6 version.

### 17. Explain about ArrayList class.

ArrayList is a Collection which can be used to represent a group of objects as a single entity.

- It is an implemented class for the List interface.

- Introduced in 1.2 version.

- The underlying data structure is resizable or growable array.

- Insertion order is preserved.

- Duplicates are allowed.

- Heterogeneous objects are allowed.

- Null insertion is possible.

- This class implements RandomAccess , Serializable , Cloneable interfaces.

- Best choice for retrieval purpose and worst if our frequent operation is insertion or deletion in the middle.

### 18. What is RandomAccess Interface?

- RandomAccess interface is marker interface and it doesn't contain any methods.

- ArrayList and vector classes implement this interface.

- If a collection class implements RandomAccess interface then we can access any of its element with the same speed.

### 19. Explain about LinkedList class.

LinkedList is a Collection implemented class which can be used for representing a group of objects as a single entity.

- LinkedList is the implemetation class for List interface.

- Introduced in 1.2 version.

- Underlying data Structure is DoubleLinkedList.

- Allows duplicates.

- Insertion order is preserved.

- Allows heterogeneous objects.

- Null insertion is possible.

- LinkedList class implements Seriallizable and Cloneable interface but not RandomAccess interface.

- Best choice if frequent operation is insertion or deletion an objects in middle but worst choice if frequent operation is retrieval.

## 20. Explain about Vector class.

Vector is a legacy collection class which can be used to represent a group of objects.

- Introduced in 1.0 version. it is legacy class.

- The underlying data structure is resizable or growable array.

- Insertion order is preserved

- Duplicates are allowed

- Heterogeneous objects are allowed

- It is a implemented class for List interface

- null insertion is possible

- Vector class implements RandomAccess , Serializable, Cloneable interfaces.

- Best Choice if frequent operation is retrieval and worst choice if frequent operation is insertion or deletion in the middle.

All methods present in Vector class are synchronized hence Vector class object is thread safe.

## 21. What is difference between ArrayList and Vector?

| ArrayList | Vector |
|---|---|
| 1. No method is synchronized in the ArrayList class | 1. All methods in Vector are synchronized. |
| 2. ArrayList object is not thread safe. | 2. Vector is thread safe. |
| 3. Relatively performance is high | 3. Relatively performance is low |
| 4. Introduced in 1.2 version and it is non legacy | 4. Introduced in 1.0 version and it is legacy |

## 22. How we can get synchronized version of ArrayList?

Collections class contains `synchronizedList()` method for this

```
Public static List synchronizedList(List l)
```

Example:
```
ArrayList l= new  ArrayList();
List l2=Collections.synchronizedList(l);
```

Similarly we can get synchronized versions of Set and Map objects by the following methods.

```
Public static List synchronizedSet(Set s)
Public static List synchronizedMap(Map m)
```

## 23. What is difference between size and capacity of a Collection Object?

Size means number of objects present where as capacity means no of objects it can accommodate.

## 24. What is difference between ArrayList and Linked List?

| ArrayList | LinkedList |
|---|---|
| 1. The underlying data structure is resizable or growable array. | 1. The underlying data structure is Double Linked List. |
| 2. This is Best choice if frequent operation is retrieval and worst choice if frequent operation is insertion or deletion in the middle. | 2. This is Best choice  if frequent operation is insertion or deletion in the middle and worst choice if frequent operation is retrieval . |
| 3. This class implements Serializable , Cloneable and RandomAccess interfaces. | 3. This class implements Serializable , Cloneable but not  RandomAccess interface. |

## 25. What are legacy classes and interfaces present in Collections framework?

- Enumeration ---Interface
- Dictonary ------Abstract class
- Hashtable -----Concrete class
- Properties -----Concrete class
- Vector -----Concrete class
- Stack  -----Concrete class

## 26. What is difference Enumeration and Iterator?

| Enumeration | Iterator |
|---|---|
| 1. It is legacy interface and introduced in 1.0 version | 1. It is non-legacy and introduced in 1.2 version |
| 2. Applicable only for legacy classes and it is not universal cursor | 2. Applicable for any Collection implemented class object. |
| 3. While iterating the elements we are not allowed to remove the objects just we can perform only read operation | 3. While iterating we can perform removal also in addition to read operation. |
| 4. By using elements() method we can get Enumeration object | 4. By using iterator() method we can get Iterator object |

## 27. What are limitations of Enumeration?

- While iterating the elements we are not allowed to perform removal operation

- It is applicable only for legacy classes and it is not a universal cursor.

- It can retrieve the elements only in forward direction

## 28. What is difference between enum and Enumeration?

An enum can be used to define a group of named constants .It has  introduced in 1.5 version.

Example:
```
Class Beer{
        KO,KF,RC,FO
        }
```

Enumeration is cursor to retrieve Objects one by one from Collection objects.

## 29.  What is difference between Iterator and ListIterator?

- ListIterator is the child interface of the Iterator

- Iterator is the single direction cursor where as ListIterator is bidirectional cursor.

- While iterating the elements by Iterator we can perform only read and remove operations. But by using ListIterator we can perform read,removal, replace and addition of new objects also.

- Iterator is applicable for every Collecton implemented class object but ListIterator  is applicable only for List implemented class objects.

- Iterator can be get by using iterator() of Collection interface where as ListIterator can be get by using listIterator() method of List interface
- both are introduced in 1.2 version

## 30. What is relation between ListIterator and Iterator?

ListIterator is child interface of Iterator

## 31. Explain about HashSet class?

- The underlying data structure is Hashtable
- null values are accepted
- duplicates are not allowed
- insertion order is based on hashcode of the object hence insertion order is not preserved
- best  suitable if frequent operation is  search operations
- HashSet  class implements Serializable and Cloneable
- it is implementation class for Set interface
- heterogeneous objects are allowed
- it is introduced in 1.2 version

## 32. If we are trying to insert duplicate values in Set what will happen?

If we are trying to insert duplicate objects to the HashSet , we won't get any compile time or run time errors just the add(Object o) returns false and it doesn't add that object.

## 33. What is LinkedHashSet?

It is the child class of HashSet. The main difference between HashSet and LinkedHashSet is: In the case of HashSet insertion order is not preserved, but in the case of LinkedHashSet insertion will be preserved.

## 34. Differences between HashSet and LinkedHashSet?

| HashSet | LinkedHashSet |
|---|---|
| 1. The Underlying datastructure is Hashtable | 1. The underlying datastructure is combination of LinkedList and |

| | Hashtable |
|---|---|
| 2. Insertion Order is not preserved | 2. Insertion order is preserved. |
| 3. Introduced in 1.2 version | 3. Introduced in 1.4 version |

**35. What are major enhancements in 1.4 version of collection frame work?**

- LinkedHashSet
- LinkedHashMap
- IdentityHashMap

**36. Explain about TreeSet?**

It is Collection object which can be used to represent a group of objects according to some sorting order.

- The underlying datastructure is Balanced tree

- Duplicates are not allowed

- All objects are stored according to some sorting order hence insertion order is not preserved

- Heterogeneous objects are not allowed violation leads to ClassCastException

- For an Empty TreeSet as firs element null value can be inserted but after inserting that first value if we are trying to insert any other objects then we will get NullPointerException

- For an non empty TreeSet if we are trying to inser null value at run time u will get NullPointerException

**37. What are differences between List and Set interfaces?**

| List | Set |
|---|---|
| 1. Insertion Order is preserved | 1. Insertion Order is not preserved |
| 2. Duplicate Objects are allowed | 2. Duplicate Objects are not allowed |
| 3. The implemented classes are ArrayList,LinkedList, Vector and Stack classes | 3. The implemented classes are HashSet, LinkedHashSet and Tree |

**38. What is Comparable interface?**

- This interface can be used for defining natural sorting order of the objects.
- It is present in java.lang package
- It contains a method public int compareTo(Object obj1)

### 39. What is Comparator interface?

- This interface can be used for implementing customized sorting order.

- It is present in java.util package

- It contains two methods

    - public int compare(Object ,Object)

    - public boolean equals(Object)

### 40. What are differences between Comparable and Comparator?

| Comparable | Comparator |
|---|---|
| 1. This can be used for natural sorting order | 1. This can be used for implementing customized sorting |
| 2. This interface present in java.lang package | 2. This is present in java.util package |
| 3. Contains only one method: public int compareTo(Object obj1) | 3. It contains two methods. public int compare(Object,Object) public Boolean equals(Object) |
| 4. It is marker interface | 4. It is not a marker interface. |

### 41. What is difference between HashSet and TreeSet?

| HashSet | TreeSet |
|---|---|
| 1. The underlying data structure is Hashtable | 1. The underlying data structure is balanced tree |
| 2. Heterogeneous objects are allowed | 2. Heterogeneous objects are not allowed bydefalut |
| 3. Insertion order is not preserved and it is based on hashcode of the objects | 3. Insertion order is not preserved and all the objects are inserted according to some sorting order. |
| 4. null insertion is possible | 4. As the first element only null insertion is possible and in all other cases we will get *NullPointerException* |

### 42. What is Entry interface?

The Entry Interface is the inner interface of Map. In the Map each key value pair is considered as Entry object.

```
interface Map{
      //more code here
      interface Entry{
            Object  getKey()
            Object  getValue()
            Object  setValue(Object new)
      }
}
```

## 43.  Explain about HashMap?

It is a Map Object which can be used used to represent a group of objects as key-value pairs.

- The underlying data structure is Hashtable

- Duplicaes keys are not allowed duplicate values are allowed

- Insertion order is not preserved because insertion is based on hashcode of keys.

- Heterogeneous objects are allowed for  both keys and values

- null key is allowed  only once

- null values  are allowed multiple times

- Introduced in 1.2 version

## 44.  Explain about LinkedHashMap?

It is child class of HashMap. It is exactly same as HashMap except the following difference.

In the case of HashMap the insertion order is not preserved but in the case of LinkedHashMap insertion order is preserved. Introduced in 1.4 version.

## 45.  Differences between HashMap and LinkedHashMap?

| HashMap | LinkedHashMap |
|---|---|
| 1.  The underlying data structure is Hashtable | 1.  The underlying data structure is a combination of Hashtable and linkedlist |
| 2.  Insertion order is not preserved and it is based on hashcode of keys | 2.  Insertion order is preserved |
| 3.  Introduced in 1.2 version | 3.  Introduced in 1.4 version. |

### 46. Differences between HashMap and Hashtable?

| HashMap | Hashtable |
|---|---|
| 1.  The underlying data structure is Hashtable | 1.  The underlying data structure of Hashtable |
| 2.  No method is synchronized and hence HashMap object is not thread safe | 2.  All methods are synchronized and hence it is   thread safe |
| 3.  Performance is high | 3.  Performance is low |
| 4.  null insertion is possible for both keys and values | 4.  null insertion is not possible for both key and value violation leads to NullPointerException |
| 5.  Introduced in 1.2 version and it is non legacy | 5.  Introduced in 1.0 version and it is legacy |

### 47. What is IdentityHashMap?

It is exactly same as HashMap except the following difference. In the HashMap JVM uses equals() method to identify duplicate keys  but in the  case of IdentityHashMap JVM uses == operator for this.

### 48. What is difference between HashMap and IdentityHashMap?

Refer Q47 for the answer.

### 49.  What is WeakHashMap?

It is exactly same as HashMap except the following difference.  In case of HashMap an Object is not eligible for garbage collection if it is associated with HashMap even though it dosen't have any external references.  ie HashMap dominates garbage collector.

But in case of WeakHashMap , if an Object is not having any external references then it is always eligible for garabage collectoion even though it is associated with weakHashMap.  i.e. garbage collector dominates WeakHashMap

### 50.  What is difference between HashMap and WeakHashMap?

Refer Q49 for the answer.

### 51.  What is TreeMap?

TreeMap can be used to store a group of objects as key-value pairs where all the entries are arranged according to some sorting order of keys.
- The underlying data structure is RED-BLACK Tree

- Duplicates keys are not allowed but values can be duplicated.

- Insertion order is not preserved because insertion is based on some sorting order

- If we are depending on Natural sorting order then keys should be homogeneous(violation leads to ClassCastException)  but values need not homogeneous

- In case of customized sorting order we can insert  heterogeneous keys and values

- For empty TreeMap as first entry with null values are allowed but after inserting that entry if we are trying to insert any other entry we will get NullPointerException

- For non empty TreeMap if we are trying to insert null keys we will get NullPointerException

- There are no restrictions for null values.

## 52.  What is Hashtable?

Hashtable is a legacy Map and can be used to store objects as key value pairs.
- The underlying data sturucture is Hashtabe

- Duplicates keys are not allowed but duplicate values are allowed

- null insertion is not possible for both keys and values

- All methods are synchronized

- Insertion order is not preserved because it is  based on hashcode  of keys

- Heterogeneous Objects are allowed for both keys and values

- Introduced in 1.0 version it is legacy class

## 53.  What is PriorityQueue?

It represents a data structure to hold group of individual objects prior to processing based on some priority .it can be natural sorting order and it can be customized sorting order described by Comparator. It is the implementation class of Queue interface.
- Insertion order is not preserved because here insertion is done based on some sorting order

- Duplicates are not allowed

- null insertion is not possible even as first element also

- If we are depending on natural sorting order Objects should be homogeneous  violation leads to ClassCastException

- If we are depending on customized sorting order Objects can be heterogeneous also.

## 54. What is Arrays class?

- It is utility class for arrays.
- It defines several utility methods for arrays like sorting an array or searching an element in array
- present in java.util package

## 55. We are planning to do an indexed search in a list of objects. Which of the two Java collections should you use: ArrayList or LinkedList?

ArrayList

## 56. Why ArrayList is faster than Vector?

All methods present in the Vector are synchronized and hence any method can be executed by only one thread at a time. It slows down the execution. But in ArrayList, no method is synchronized and hence multiple threads are allowed execute simultaneously which speed up the execution.

## 57. What is an Iterator?

Some of the collection classes provide traversal of their contents via a java.util.Iterator interface. This interface allows you to walk through a collection of objects, operating on each object in turn.

Remember when using Iterators that they contain a snapshot of the collection at the time the Iterator was obtained; generally it is not advisable to modify the collection itself while traversing an Iterator.

## 58. What is HashMap and Map?

Map is an Interface and Hashmap is the class that implements Map.

## 59. Difference between HashMap and HashTable?

The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls. (HashMap allows null values as key and value whereas Hashtable doesnt allow).

HashMap does not guarantee that the order of the map will remain constant over time. HashMap is unsynchronized and Hashtable is synchronized.

## 60. Difference between Vector and ArrayList?

Vector is synchronized whereas arraylist is not.

### 61. What is the Collections API?

The Collections API is a set of classes and interfaces that support operations on collections of objects.

### 62. Which characters may be used as the second character of an identifier, but not as the first character of an identifier?

The digits 0 through 9 may not be used as the first character of an identifier but they may be used after the first character of an identifier.

### 63. What is the List interface?

The List interface provides support for ordered collections of objects.

### 64. What is the Vector class?

The Vector class provides the capability to implement a growable array of objects.

### 65. What is an Iterator interface?

The Iterator interface is used to step through the elements of a Collection.

### 66. What is the Map interface?

The Map interface replaces the JDK 1.1 Dictionary class and is used associate keys with values.

### 67. What is the Collection interface?

The Collection interface provides support for the implementation of a mathematical bag - an unordered collection of objects that may contain duplicates.

### 68. What is the Set interface?

The Set interface provides methods for accessing the elements of a finite mathematical set. Sets do not allow duplicate elements.

### 69.  What is the difference between Array and vector?

Array is a set of related data type and static whereas vector is a growable array of objects and dynamic.

### 70.  What are Vector, Hashtable, LinkedList and Enumeration?

Vector: The Vector class provides the capability to implement a growable array of objects. Hashtable : The Hashtable class implements a Hashtable data structure. A Hashtable indexes and stores objects in a dictionary using hash codes as the object's keys. Hash codes are integer values that identify objects. LinkedList: Removing or inserting elements in the middle of an array can be done using LinkedList. A LinkedList stores each object in a separate link whereas an array stores object references in consecutive locations. Enumeration: An object that implements the Enumeration interface generates a series of elements, one at a time. It has two methods, namely hasMoreElements() and nextElement(). HasMoreElemnts() tests if this enumeration has more elements and nextElement method returns successive elements of the series.

### 71.  What is the difference between set and list?-

Set stores elements in an unordered way but does not contain duplicate elements, whereas list stores elements in an ordered way but may contain duplicate elements.