Q1. What are limitations of object Arrays?

The main limitations of Object arrays are

- These are fixed in size ie once we created an array object there is no chance of increasing or decreasing size based on our requirement. Hence If we don't know size in advance , arrays are not recommended to use
- Arrays can hold only homogeneous elements.
- There is no underlying data structure for arrays and hence no readymade method support for arrays. Hence for every requirement programmer has to code explicitly

To over come these problems collections are recommended to use

Q2. What are differences between arrays and collections?

Arrays	Collections
1. Arrays r fixed in size and hence once we created an array we are not allowed to increase or decrease the size based on our requirement.	1. Collections are grow able in nature and hence based on our requirement we can increase or decrease the size.
2. Memory point of view arrays are not recommended to use	2. Memory point of view collections are recommended to use.
3. Performance point of view arrays are recommended to use	3. Performance point of view collections are not recommended to use.
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 ready method support compulsory programmer has to code explicitly.	6. For every requirement ready made method support is available. Being a programmer we have to know how to use those methods and we are not responsible to implement those.

Q3. What are differences between arrays and ArrayList?

Refer the answer of Q2

Q4. What are differences between arrays and Vector?

Refer the answer of Q2

Q5. What is Collection API ?

It defines set of classes and interfaces which can be used for representing a group of objects as single entity

Q6. What is Collection framework?

It defines set of classes and inter faces which can be used for representing a group of objects as single entity

Q7. What is 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" is an utility class to define several utility methods for Collection implemented class objects.

Q8. Explain about Collection interface?

- This 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

Q9. Explain about List interface?

List interface is a child interface of Collection interface. This can be used to represent group of individual objects in as a single entity where

- Duplicates are allowed
- Insertion order is preserved

Q10. Explain about Set interface?

Set 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.
- Insertion order is not preserved

Q11. Explain about SortedSet interface?

it **is child** interface of Set interface. it can be used to represent a group of individual objects in to a single entity where

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

Q12. 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
- It is introduced in 1.6 version

Q13. 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 introduced in 1.5 version.

Q14. 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 introduced in 1.2 version

Q15. 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.
- It has introduced in 1.2 version

Q16. Explain about NavigableMap?

- It is child interface of SortedMap and defines several method for navigation purpose
- It is introduced in 1.6 version

Q17. Explain about ArrayList class?

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

- it is a implemented class for 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

Q18. What is RandomAccess Interface?

- If a collection class implements RandomAccess interface then we can access any of its element with the same speed.
- RandomAccess interface is marker interface and it dosent contains any methods.
- ArrayList and vector classes implements this interface.

Q19. 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.

Q20. 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.

Q21. 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

Q22. How we can get synchronized version of ArrayList?

Collections class contains synchronizedList() method for this

```
Public static List synchronizedList(List l)

EX
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)

Q23. 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.

Q24. 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.

Q25. 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

Q26. 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
2Applicable only for legacy classes and it is not universal	2Applicable for any Collection implemented class object.

cursor	
3While iterating the elements we are not allowed to remove the objects just we can perform only read operation	3While iterating we can perform removal also in addition to read operation.
4By using elements() method we can get Enumeration object	4. By using iterator() method we can get Iterator object

Q27. 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

Q28. 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

Ex

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

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

Q29. 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

Q30. What is relation between ListIterator and Iterator?

ListIterator is child interface of Iterator

Q31. 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

Q32. 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 wont get any compile time or run time errors just the add(Object o) returns false and it doesn't add that object.

Q33. 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.

Q34. Differences between HashSet and LinkedHashSet?

HashSet	LinkedHashSet
1The Underlying datastructure is Hashtable	1The underlying datastructure is combination of LinkedList and Hashtable
2Insertion Order is not preserved	2 Insertion order is preserved.
3Introduced in 1.2 version	3 Introduced in 1.4 version

Q35. What are major enhancements in 1.4 version of collection frame work?

LinkedHashSet
LinkedHashMap
IdentityHashMap

Q36. 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

Q37. What are differences between List and Set interfaces?

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

Q38. 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)**

Q39. 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)

Q40. What are differences between Comparable and Comparator?

Comparable	Comparator
1This can be used for natural sorting order	1This can be used for implementing customized sorting
2This interface present in java.lang package	2 This is present in java.util package
3Contains 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.

Q41. What is difference between HashSet and TreeSet?

HashSet	TreeSet
1The underlying data structure is Hashtable	1The underlying data structure is balanced tree
2Heterogeneous objects are allowed	2 Heterogeneous objects are not allowed bydefalut
3Insertion 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.
4null insertion is possible	4 As the first element only null insertion is possible and in all other cases we will get NullPointerException

Q42. What is Entry interface?

It is 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)
        }
}
```

Q43. 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

Q44. 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

Q45. 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.

Q46. 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

Q47. 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.

Q48. What is difference between HashMap and IdentityHashMap?

Refer Q47 for the answer.

Q49. 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 dosent 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. ie garbage collector dominates WeakHashMap

Q50. What is difference between HashMap and WeakHashMap?

Refer Q49 for the answer.

Q51. 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.

Q52. 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

Q53. 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.

Q54. 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

Q55. 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
Q56. 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 thread are allowed execute simultaneously which speed up the execution.