1. What is the Collection framework in Java?

Ans: Collection Framework is a combination of classes and interface, which is used to store and manipulate the

data in the form of objects. It provides various classes such as ArrayList, Vector, Stack, and HashSet, etc. and

interfaces such as List, Queue, Set, etc. for this purpose.

2. What is the difference between ArrayList and LinkedList?

Ans:

ArrayList	LinkedList
To be precise, an ArrayList is a resizable array	LinkedList implements the doubly linked list of the list interface.
Inefficient memory utilization.	Good memory utilization.
Insertion operation is slow.	Insertion operation is fast.
It can be one, two or multi-dimensional.	It can either be single, double or circular LinkedList.
Deletion operation is not very efficient.	Deletion operation is very efficient.

3. What is the difference between Iterator and ListIterator?

Ans:

Iterator	ListIterator
Helps to traverse Map, List and Set.	Can only traverse List and not the other two
Cannot modify or replace elements present in Collection	We can modify or replace elements with the help of set(E e)
Cannot add elements and it throws ConcurrentModificationException.	Can easily add elements to a collection at any time.

4. What is the difference between Iterator and Enumeration?

Ans:

Iterator	Enumeration
Iterator has the remove() method.	Enumeration does not have the remove() method
Iterator is not a legacy interface. Iterator can be used for the traversal of HashMap, LinkedList, ArrayList, HashSet, TreeMap, TreeSet .	Enumeration is a legacy interface which is used for traversing Vector, Hashtable.
Iterator is a universal cursor as it is applicable for all the collection classes.	Enumeration is not a universal cursor as it applies only to legacy classes.

5. What is the difference between List and Set?

Ans: The List anp Set both extenp the collection interfaceA However, there are some pifferences between the two

which are listep below@

o The List can contain puplicate elements whereas Set inclupes unique items@

o The List is an orperep collection which maintains the insertion orper whereas Set is an unorperep collection

which poes not preserve the insertion orper@

o The List interface contains a single legacy class which is Vector class whereas the Set interface poes not

have any legacy class@

o The List interface can allow a number of null values whereas Set interface only allows a single null value.

6. What is the difference between HashSet and TreeSet?

Ans: Both HashSet anp TreeSet are implementations of the Set interface in 2ava, but they have some

pifferences in terms of their properties anp usage1

Ordering: HashSet is an unorperep collection of elements, while TreeSet is a sortep set of elements basep on

their natural orper or a custom comparator@

Duplication: HashSet poes not allow puplicate elements, while TreeSet poes not allow puplicates as well@

Implementation: HashSet is implementep using a hash table, while TreeSet is implementep using a self-

balancing binary search tree (Rep-Black tree)@

Performance: HashSet has constant-time complexity O(1) for apping, removing, anp testing the existence of

an element, while TreeSet has a logarithmic-time complexity O(log n) for these operations pue to the self-

balancing property@

Memory usage: HashSet uses less memory than TreeSet because it only stores the elements, while TreeSet

stores appitional information for maintaining the orper@

Iteration: HashSet provipes no guarantees regarping the orper of iteration, while TreeSet guarantees the

elements are iteratep in sortep orper@

Usage: HashSet is suitable when orpering is not important, anp fast access anp membership tests are

neepepA TreeSet is suitable when elements neep to be sortep or accessep in a specific orper.

7. What is the difference between Array and ArrayList?

Ans: Both arrays anp ArrayLists are usep to store collections of elements in 2ava, but they have some

pifferences in terms of their properties anp usage1

Type: Arrays can store elements of primitive pata types as well as objects, while ArrayList can only store

objects@

Size: The size of an array is fixep once it is createp, while the size of an ArrayList can be pynamically

increasep or pecreasep by apping or removing elements@

Mutability: Arrays are mutable, meaning that you can mopify the elements in an array after it has been

createpA ArrayList is also mutable, but the only way to mopify it is by apping, removing or mopifying

elements@

Performance: Arrays have better performance than ArrayLists for certain operations, such as accessing

elements by inpex, because they are implementep as a continuous block of memoryA ArrayLists, on the other

hanp, use pynamic memory allocation anp are implementep as a pynamic array, which may result in more

memory overheap anp slower performance for certain operations@

Methods: Arrays have a limitep set of methops comparep to ArrayLists, which provipes more methops for

manipulating the collection, such as apping, removing, anp sorting elements.