Collection Framework

1. What is the Collection framework in Java?

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?

ArrayList	LinkedList
1) ArrayList internally uses a dynamic array to store the elements.	LinkedList internally uses a doubly linked list to store the elements.
2) Manipulation with ArrayList is slow because it internally uses an array. If any element is removed from the array, all the other elements are shifted in memory.	Manipulation with LinkedList is faster than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory.
3) An ArrayList class can act as a list only because it implements List only.	LinkedList class can act as a list and queue both because it implements List and Deque interfaces.
4) ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.
5) The memory location for the elements of an ArrayList is contiguous.	The location for the elements of a linked list is not contagious.

3. What is the difference between Iterator and ListIterator?

BASIS FOR COMPARISON	ITERATOR	LISTITERATOR
Basic	Iterator can traverse the elements in a collection only in forward direction.	ListIterator can traverse the elements in a collection in forward as well as the backwards direction.
Add	Iterator is unable to add elements to a collection.	ListIteror can add elements to a collection.
Modify	Iterator can not modify the elements in a collection.	ListIterator can modify the elements in a collection using set().
Traverse	Iterator can traverse Map, List and Set.	ListIterator can traverse List objects only.
Index	Iterator has no method to obtain an index of the element in a collection.	Using ListIterator, you can obtain an index of the element in a collection.

4. What is the difference between Iterator and Enumeration?

Sr. No.	Key	Iterator	Enumeration
1	Basic	In Iterator, we can read and remove element while traversing element in the collections.	Using Enumeration, we can only read element during traversing element in the collections.
2.	Access	It can be used with any class of the collection framework.	It can be used only with legacy class of the collection framework such as a Vector and HashTable.
3.	Fail-Fast and Fail -Safe	Any changes in the collection, such as removing element from the collection during a thread is iterating collection then it throw concurrent modification exception.	Enumeration is Fail safe in nature. It doesn’t throw concurrent modification exception
4.	Limitation	Only forward direction iterating is possible	Remove operations can not be performed using Enumeration.

5.	Methods	It has following methods −
		*hasNext()
		*next()
		*remove()
			It has following methods −
			*hasMoreElements()
			*nextElement()

5. What is the difference between List and Set?

S. No	List	Set
1.	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.
2.	The insertion order is maintained by the List.	It doesn't maintain the insertion order of elements.
3.	List allows us to add any number of null values.	Set allows us to add at least one null value in it.
4.	The List implementation classes are LinkedList and ArrayList.	The Set implementation classes are TreeSet, HashSet and LinkedHashSet.
5.	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().

6. What is the difference between HashSet and TreeSet?

Parameters	HashSet	TreeSet
Ordering or Sorting	It does not provide a guarantee to sort the data.	It provides a guarantee to sort the data. The sorting depends on the supplied Comparator.
Null Objects	In HashSet, only an element can be null.	It does not allow null elements.
Comparison	It uses hashCode() or equals() method for comparison.	It uses compare() or compareTo() method for comparison.
Performance	It is faster than TreeSet.	It is slower in comparison to HashSet.
Implementation	Internally it uses HashMap to store its elements.	Internally it uses TreeMap to store its elements.
Data Structure	HashSet is backed up by a hash table.	TreeSet is backed up by a Red-black Tree.
Values Stored	It allows only heterogeneous value.	It allows only homogeneous value.

7.What is the difference between Array and ArrayList?

Basis	Array	ArrayList
Definition	An array is a dynamically-created object. It serves as a container that holds the constant number of values of the same type. It has a contiguous memory location.	The ArrayList is a class of Java Collections framework. It contains popular classes like Vector, HashTable, and HashMap.
Static/ Dynamic	Array is static in size.	ArrayList is dynamic in size.
Resizable	An array is a fixed-length data structure.	ArrayList is a variable-length data structure. It can be resized itself when needed.
Initialization	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.
Performance	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.
Primitive/ Generic type	An array can store both objects and primitives type.	We cannot store primitive type in ArrayList. It automatically converts primitive type to object.
Iterating Values	We use for loop or for each loop to iterate over an array.	We use an iterator to iterate over ArrayList.
Type-Safety	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.

Length	Array provides a length variable which denotes the length of an array.	ArrayList provides the size() method to determine the size of ArrayList.
Adding Elements	We can add elements in an array by using the assignment operator.	Java provides the add() method to add elements in the ArrayList.
Single/ Multi-Dimensional	Array can be multi-dimensional.	ArrayList is always single-dimensional.