Week4-Session1 Interview Questions

1. What are Java collections?

Collections in Java is a framework that provides classes and interfaces to store, manage, and manipulate groups of objects. They offer various data structures and algorithms for working with groups of elements efficiently.

2. What is the difference between Array and Collection?

The following are the important differences between Arrays and Collection.

Arrays	Collection
Arrays are fixed in size i.e once the array with the specific size is declared then we can't change it's size.	The collection is dynamic in size i.e based on requirement size could be altered even after its declaration.
Arrays due to fast execution consumes more memory and has better performance.	Collections, on the other hand, consume less memory but also have low performance as compared to Arrays.
Arrays can hold both object and primitive type data.	Collection can hold only object types but not the primitive type of data.

3. Name some classes that implement the List interface.

Some classes that implement the List interface are ArrayList, LinkedList, and Vector.

4. How is Vector different from ArrayList?

Vector is synchronized by default, which makes it thread-safe for concurrent access. ArrayList is not synchronized by default.

5. What is the addAll() method in the List interface used for?

The **addAll(Collection<? extends E> c)** method is used to append all elements from the specified collection c to the end of the invoking list.

6. How do you check if an element exists in a List?

You can use the **contains(Object o)** method of the List interface to check if the specified element is present in the list.

7. Explain the indexOf() and lastIndexOf() methods in the List interface.

The **indexOf(Object o)** method returns the index of the first occurrence of the specified element in the list. The **lastIndexOf(Object o)** method returns the index of the last occurrence of the specified element in the list.

8. Explain the subList() method in the List interface.

The **subList(int fromIndex, int toIndex)** method returns a new List containing elements from the specified range [fromIndex, toIndex) of the original list. Changes made to the sub-list are reflected in the original list, and vice versa.

9. What is the difference between remove(int index) and clear() methods in ArrayList?

The **remove(int index)** method removes the element at the specified index, while the **clear()** method removes all elements from the list.

10. How can you remove elements from an ArrayList?

You can use methods like **remove(int index)** to remove an element at a specific index, or **remove(Object o)** to remove the first occurrence of a specified element.