1. **Internal Working of ArrayList, Set and hashmap in Java**

* **ArrayList** is a resizable array implementation in java. ArrayList grows dynamically and ensures that there is always a space to add elements. The backing data structure of ArrayList is an array of Object class.

Internally an ArrayList uses an Object[] Array which is an array of objects. All operation like deleting, adding and updating the elements happens in this Object[] array.

* **HashSet** uses HashMap internally to store it’s objects. Whenever you create a HashSet object, one **HashMap** object associated with it is also created. This HashMap object is used to store the elements you enter in the HashSet. The elements you add into HashSet are stored as **keys** of this HashMap object. The value associated with those keys will be a **constant**.
* Internally HashMap uses **a hashCode of the key Object** and this hashCode is further used by the hash function to find the index of the bucket where the new entry can be added. HashMap uses multiple buckets and each bucket points to a Singly Linked List where the entries (nodes) are stored.

1. **How to maintain order in set**

Use **HashSet** if you don't want to maintain any order of elements. Use LinkedHashSet if you want to maintain insertion order of elements. Use TreeSet if you want to sort the elements according to some Comparator.

1. **Difference between arraylist and linkedlist.**

|  |  |
| --- | --- |
| **ArrayList** | **LinkedList** |
| 1. ArrayList internally uses a **dynamic array** 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 bits are shifted in memory. 3. An ArrayList class can **act as a list** only because it implements List only. 4. ArrayList is **better for storing and accessing** data. | 1. LinkedList internally uses a **doubly linked list** to store the elements. 2. Manipulation with LinkedList is **faster** than ArrayList because it uses a doubly linked list, so no bit shifting is required in memory. 3. LinkedList class can **act as a list and queue** both because it implements List and Deque interfaces. 4. LinkedList is **better for manipulating** data. |

**4)What is fail fast and fail safe**

Fail-safe iterators means **they will not throw any exception even if the collection is modified while iterating over it**. Whereas Fail-fast iterators throw an exception(ConcurrentModificationException) if the collection is modified while iterating over it.

**5)What is hashcode and equals, why to use it for collections.**

When comparing two objects together, Java calls their **equals()** method which returns true if the two objects are equal, or false otherwise. Note that this comparison using **equals()**method is very different than using the == operator.

The **equals()**method is designed to compare two objects semantically (by comparing the data members of the class), whereas the == operator compares two objects technically (by comparing their references i.e. memory addresses).

A **hashtable** groups its elements by their hash code values. This arrangement helps the **hashtable** locates an element quickly and efficiently by searching on small parts of the collection instead the whole collection.