LINKEDHASHSET

(1) Characteristics of LinkedHashSet?

- ➢ Ordered: Elements are maintained in their insertion order.
- ➢ Unique Elements: Duplicates are not allowed, similar to HashSet.
- ➢ Performance: Slower than HashSet due to the additional cost of maintaining the linked list.
- ➢ Thread Safety: It is not thread-safe by default.

(2) INTERNAL WORKING OF HASHSET ?

LinkedHashSet is a subclass of HashSet and uses a combination of:

(a) HashMap:

- ➢ Internally, LinkedHashSet uses a LinkedHashMap to store its elements.
- ➢ The keys of the LinkedHashMap store the elements of the set, and the values are dummy objects.

(b) Linked List:

- ➢ The LinkedHashMap maintains a doubly-linked list of its entries to preserve the insertion order of elements.

Ex-

```java
public static void main(String[] args) {
    // Create a LinkedHashSet
    LinkedHashSet<Integer> set = new LinkedHashSet<>();

    // Add elements
    set.add(10);
    set.add(20);
    set.add(30);
    set.add(20); // Duplicate, will not be added

    // Display the LinkedHashSet
    System.out.println("LinkedHashSet: " + set);

    // Check if an element is present
    System.out.println("Contains 20? " + set.contains(20));

    // Remove an element
    set.remove(20);
    System.out.println("After removing 20: " + set);
    // Get the size of the LinkedHashSet
    System.out.println("Size: " + set.size());

    // Check if the LinkedHashSet is empty
    System.out.println("Is Empty? " + set.isEmpty());

    // Traverse the LinkedHashSet
    System.out.println("Traversing LinkedHashSet:");
    for (Integer i : set) {
        System.out.println(i);
    }

    // Clear the LinkedHashSet
    set.clear();
    System.out.println("After clear: " + set);
}
```