DSA

```
import java.util.*;

Public class CollectionDemo {
        Public static void main (string[] args) {

    // Linklist
    Linkedlist <string> LinkedList = new LinkedList<>();
    LinkedList.add("A");
    LinkedList.add("c");
   LinkedList.add(1,"B");  // Insert a position
   System.out.printIn ("Linklist :" + linkedList);

   // PriorityQueue
  PriorityQueue<Integer> pq = new PriorityQueue<>();

  pq.add (5);
  pq.add (2);
  pq.add (8);
 System.out.printIn();
 // Deque
 Deque<string> deque = new ArrayDeque<>();

deque.add ("first");
deque.addFirst("Zero"); // Add at front
deque.addLast ("second"); // Add at end
System.out.printIn ("Deque : + deque);
system.out.printIn ("Removing from Deque:");
System.out.printIn("Remove First:" + deque.removeFirst()) ;
System.out.printIn ("Remove Last: " + deque.removeLast());
// HashMap
Map<string, Integerr> hashMap = new HashMap<>();
hashMap.put ("Alice", 30);
hashMap.put ("Bob", 25);
hashMap.put ("Charlie, 35);
System.out.println ("HashMap:" + hashMap); // No guaranteed
                                                orderr

// Tree Map
Map<string, Integerr> treeMap = new TreeMap <>();
treeMap.put ("Alice", 30);
treeMap.put("Bob", 25);
treeMap.put ("Charrlie", 35);
System.out.printIn ("TreeMap (sorted by key): "+ treeMap

// Hashset
set<string> hashSet = new HashSet<>();
hashset.add ("Apple");
hashset.add ("Banana"),
hashset.add ("Apple"); // Duplicate ignored
system.out.printIn (" Hashset (unique elements, no orderr):
                                          "+ hashset);
}

}
```