ArrayList

```
import java.util.ArrayList;

public class ArrayListExample {
    public static void main(String[] args) {
        // Create
        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("Element1");
        arrayList.add("Element2");
        arrayList.add("Element3");

        // Read
        System.out.println("ArrayList elements: " + arrayList);

        // Update
        arrayList.set(1, "UpdatedElement2");
        System.out.println("ArrayList after update: " + arrayList);

        // Delete
        arrayList.remove(2);
        System.out.println("ArrayList after deletion: " + arrayList);
    }
}
```

LinkedList

```
import java.util.LinkedList;

public class LinkedListExample {
    public static void main(String[] args) {
        // Create
        LinkedList<String> linkedList = new LinkedList<>();
        linkedList.add("Element1");
        linkedList.add("Element2");
        linkedList.add("Element3");

        // Read
        System.out.println("LinkedList elements: " + linkedList);

        // Update
        linkedList.set(1, "UpdatedElement2");
        System.out.println("LinkedList after update: " + linkedList);

        // Delete
        linkedList.remove(2);
        System.out.println("LinkedList after deletion: " + linkedList);
    }
}
```

Vector

```
import java.util.Vector;

public class VectorExample {
    public static void main(String[] args) {
        // Create
        Vector<String> vector = new Vector<>();
        vector.add("Element1");
        vector.add("Element2");
        vector.add("Element3");

        // Read
        System.out.println("Vector elements: " + vector);

        // Update
        vector.set(1, "UpdatedElement2");
        System.out.println("Vector after update: " + vector);

        // Delete
        vector.remove(2);
        System.out.println("Vector after deletion: " + vector);
    }
}
```

Stack

```
import java.util.Stack;

public class StackExample {
    public static void main(String[] args) {
        // Create
        Stack<String> stack = new Stack<>();
        stack.push("Element1");
        stack.push("Element2");
        stack.push("Element3");

        // Read
        System.out.println("Stack elements: " + stack);

        // Update
        stack.set(1, "UpdatedElement2");
        System.out.println("Stack after update: " + stack);

        // Delete
        stack.pop();
        System.out.println("Stack after deletion: " + stack);
    }
}
```

PriorityQueue

```
import java.util.PriorityQueue;


public class PriorityQueueExample {
    public static void main(String[] args) {
        // Create
        PriorityQueue<String> priorityQueue = new PriorityQueue<>();
        priorityQueue.add("Element1");
        priorityQueue.add("Element2");
        priorityQueue.add("Element3");

        // Read
        System.out.println("PriorityQueue elements: " + priorityQueue);

        // Update (PriorityQueue doesn't support direct update)
        priorityQueue.remove("Element2");
        priorityQueue.add("UpdatedElement2");
        System.out.println("PriorityQueue after update: " + priorityQueue);

        // Delete
        priorityQueue.poll();
        System.out.println("PriorityQueue after deletion: " + priorityQueue);
    }
}
```

ArrayDeque

```
import java.util.ArrayDeque;

public class ArrayDequeExample {
    public static void main(String[] args) {
        // Create
        ArrayDeque<String> arrayDeque = new ArrayDeque<>();
        arrayDeque.add("Element1");
        arrayDeque.add("Element2");
        arrayDeque.add("Element3");

        // Read
        System.out.println("ArrayDeque elements: " + arrayDeque);

        // Update (ArrayDeque doesn't support direct update)
        arrayDeque.remove("Element2");
        arrayDeque.add("UpdatedElement2");
        System.out.println("ArrayDeque after update: " + arrayDeque);

        // Delete
        arrayDeque.remove();
        System.out.println("ArrayDeque after deletion: " + arrayDeque);
    }
}
```

HashSet

```
import java.util.HashSet;

public class HashSetExample {
    public static void main(String[] args) {
        // Create
        HashSet<String> hashSet = new HashSet<>();
        hashSet.add("Element1");
        hashSet.add("Element2");
        hashSet.add("Element3");
        hashSet.add("Element3");

        // Read
        System.out.println("HashSet elements: " + hashSet);

        // Update (HashSet doesn't support direct update)
        hashSet.remove("Element2");
        hashSet.add("UpdatedElement2");
        System.out.println("HashSet after update: " + hashSet);

        // Delete
        hashSet.remove("Element3");
        System.out.println("HashSet after deletion: " + hashSet);
    }
}
```

LinkedHashSet

```
import java.util.LinkedHashSet;

public class LinkedHashSetExample {
  public static void main(String[] args) {
    // Create
    LinkedHashSet<String> linkedHashSet = new LinkedHashSet<>();
    linkedHashSet.add("Element1");
    linkedHashSet.add("Element2");
    linkedHashSet.add("Element3");
    linkedHashSet.add("Element3");

    // Read
    System.out.println("LinkedHashSet elements: " + linkedHashSet);

    // Update (LinkedHashSet doesn't support direct update)
    linkedHashSet.remove("Element2");
    linkedHashSet.add("UpdatedElement2");
    System.out.println("LinkedHashSet after update: " + linkedHashSet);

    // Delete
    linkedHashSet.remove("Element3");
    System.out.println("LinkedHashSet after deletion: " + linkedHashSet);
  }
}
```

TreeSet

```
import java.util.TreeSet;

public class TreeSetExample {
    public static void main(String[] args) {
        // Create
        TreeSet<String> treeSet = new TreeSet<>();
        treeSet.add("Element1");
        treeSet.add("Element2");
        treeSet.add("Element3");
        treeSet.add("Element3");

        // Read
        System.out.println("TreeSet elements: " + treeSet);

        // Update (TreeSet doesn't support direct update)
        treeSet.remove("Element2");
        treeSet.add("UpdatedElement2");
        System.out.println("TreeSet after update: " + treeSet);

        // Delete
        treeSet.remove("Element3");
        System.out.println("TreeSet after deletion: " + treeSet);
    }
}
```

HashMap

```
import java.util.HashMap;


public class HashMapExample {
    public static void main(String[] args) {
        // Create
        HashMap<Integer, String> hashMap = new HashMap<>();
        hashMap.put(1, "Element1");
        hashMap.put(2, "Element2");
        hashMap.put(3, "Element3");
        hashMap.put(3, "Element3");

        // Read
        System.out.println("HashMap elements: " + hashMap);

        // Update
        hashMap.put(2, "UpdatedElement2");
        System.out.println("HashMap after update: " + hashMap);

        // Delete
        hashMap.remove(3);
        System.out.println("HashMap after deletion: " + hashMap);
    }
}
```

LinkedHashMap

```
import java.util.LinkedHashMap;


public class LinkedHashMapExample {
    public static void main(String[] args) {
        // Create
        LinkedHashMap<Integer, String> linkedHashMap = new
LinkedHashMap<>();
        linkedHashMap.put(1, "Element1");
        linkedHashMap.put(2, "Element2");
        linkedHashMap.put(3, "Element3");
        linkedHashMap.put(3, "Element3");

        // Read
        System.out.println("LinkedHashMap elements: " + linkedHashMap);

        // Update
        linkedHashMap.put(2, "UpdatedElement2");
        System.out.println("LinkedHashMap after update: " + linkedHashMap);

        // Delete
        linkedHashMap.remove(3);
        System.out.println("LinkedHashMap after deletion: " + linkedHashMap);
    }
}
```

TreeMap

```
import java.util.TreeMap;


public class TreeMapExample {
    public static void main(String[] args) {
        // Create
        TreeMap<Integer, String> treeMap = new TreeMap<>();
        treeMap.put(1, "Element1");
        treeMap.put(2, "Element2");
        treeMap.put(3, "Element3");
        treeMap.put(3, "Element3");

        // Read
        System.out.println("TreeMap elements: " + treeMap);

        // Update
        treeMap.put(2, "UpdatedElement2");
        System.out.println("TreeMap after update: " + treeMap);

        // Delete
        treeMap.remove(3);
        System.out.println("TreeMap after deletion: " + treeMap);
    }
}
```