NAME: A.S.MOHAMED MUBEEN

CODINC CHALLENGE IV

1.Write a Java Program To Insert An Element at The Specified Position At The LinkedList

```
package com.codingchallenge4;

import java.util.LinkedList;

public class Inserlinkedlist {

    public static void main(String[] args) {

        LinkedList<String> linkedList = new LinkedList<>();


        linkedList.add("Apple");
        linkedList.add("Banana");
        linkedList.add("Orange");
        linkedList.add("Mango");

        System.out.println("Original LinkedList: " + linkedList);


        int indexToInsert = 2;
        String elementToInsert = "Grapes";

        if (indexToInsert >= 0 && indexToInsert <= linkedList.size()) {
            linkedList.add(indexToInsert, elementToInsert);
            System.out.println("Element '" + elementToInsert + "' inserted
at position " + indexToInsert + ": " + linkedList);
        } else {
            System.out.println("Invalid index. Insertion failed.");
        }

    }
}
```

Output:

```
Original LinkedList: [Apple, Banana, Orange, Mango]
Element 'Grapes' inserted at position 2: [Apple, Banana, Grapes, Orange,
Mango]
```

2.Write A Java Program To Test An ArrayList Is Empty Or Not

```
package com.codingchallenge4;
import java.util.ArrayList;

public class ArrayListIsEmpty {

    public static void main(String[] args) {

        ArrayList<String> arrayList = new ArrayList<>();


        if (arrayList.isEmpty()) {
            System.out.println("ArrayList is empty");
        } else {
            System.out.println("ArrayList is not empty");
        }


        arrayList.add("Apple");
        arrayList.add("Banana");
        arrayList.add("Orange");


        if (arrayList.isEmpty()) {
            System.out.println("ArrayList is empty");
        } else {
            System.out.println("ArrayList is not empty");
        }
    }
}
```

Output:

```
ArrayList is empty
ArrayList is not empty
```

3.Write a Java Program To Convert A HashSet To An ArrayList

```
package com.codingchallenge4;
import java.util.HashSet;
import java.util.ArrayList;
import java.util.List;

public class HashSetToArrayList {

    public static void main(String[] args) {

        HashSet<String> hashSet = new HashSet<>();

        hashSet.add("Apple");
        hashSet.add("Banana");
        hashSet.add("Orange");
        hashSet.add("Mango");

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

        List<String> arrayList = new ArrayList<>(hashSet);

        System.out.println("ArrayList: " + arrayList);
    }
}
```

Output:

```
HashSet: [Apple, Mango, Orange, Banana]
ArrayList: [Apple, Mango, Orange, Banana]
```

4.Write A Program To Sort HashMap by keys

```
package com.codingchallenge4;
import java.util.*;

public class SortHashMapByKeys {

    public static void main(String[] args) {

        HashMap<Integer, String> hashMap = new HashMap<>();

        hashMap.put(3, "Three");
        hashMap.put(1, "One");
        hashMap.put(4, "Four");
        hashMap.put(2, "Two");

        System.out.println("Original HashMap: " + hashMap);

        List<Integer> sortedKeys = new ArrayList<>(hashMap.keySet());
        Collections.sort(sortedKeys);

        LinkedHashMap<Integer, String> sortedHashMap = new
LinkedHashMap<>();

        for (Integer key : sortedKeys) {
            sortedHashMap.put(key, hashMap.get(key));
        }

        System.out.println("Sorted HashMap by Keys: " + sortedHashMap);
    }
}
```

Output:

```
Original HashMap: {1=One, 2=Two, 3=Three, 4=Four}
Sorted HashMap by Keys: {1=One, 2=Two, 3=Three, 4=Four}
```

5.Write A Program To Iterate TreeMap in java

```
package com.codingchallenge4;
import java.util.*;

public class IterateTreeMap {

    public static void main(String[] args) {

        TreeMap<Integer, String> treeMap = new TreeMap<>();


        treeMap.put(3, "Three");
        treeMap.put(1, "One");
        treeMap.put(4, "Four");
        treeMap.put(2, "Two");

        // Method 1: Iterate using entrySet() and enhanced for loop
        System.out.println("Iterating TreeMap using entrySet() and enhanced
for loop:");
```

```
        for (Map.Entry<Integer, String> entry : treeMap.entrySet()) {
            System.out.println("Key: " + entry.getKey() + ", Value: " +
entry.getValue());
        }

        // Method 2: Iterate using keySet() and enhanced for loop
        System.out.println("\nIterating TreeMap using keySet() and enhanced
for loop:");
        for (Integer key : treeMap.keySet()) {
            System.out.println("Key: " + key + ", Value: " +
treeMap.get(key));
        }

        // Method 3: Iterate using forEach() method and lambda expression
        System.out.println("\nIterating TreeMap using forEach() method and
lambda expression:");
        treeMap.forEach((key, value) -> {
            System.out.println("Key: " + key + ", Value: " + value);
        });
    }
}
```

Output:

```
Iterating TreeMap using entrySet() and enhanced for loop:
Key: 1, Value: One
Key: 2, Value: Two
Key: 3, Value: Three
Key: 4, Value: Four

Iterating TreeMap using keySet() and enhanced for loop:
Key: 1, Value: One
Key: 2, Value: Two
Key: 3, Value: Three
Key: 4, Value: Four

Iterating TreeMap using forEach() method and lambda expression:
Key: 1, Value: One
Key: 2, Value: Two
Key: 3, Value: Three
Key: 4, Value: Four
```

6 Write A Program To sort ArrayList Using Comparable And Comparator Interface

Sortedarraylist using comparator interface

```
package com.codingchallenge4;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

class Employee1 {
    private int id;
    private String name;

    public Employee1(int id, String name) {
        this.id = id;
        this.name = name;
    }
    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    @Override
    public String toString() {
        return "Employee{id=" + id + ", name='" + name + "'}";
    }
}

class Employee1IdComparator implements Comparator<Employee1> {
    @Override
    public int compare(Employee1 emp1, Employee1 emp2) {
        return Integer.compare(emp1.getId(), emp2.getId());
    }
}

      public class SortArrayListUsingComparator {
          public static void main(String[] args) {

              ArrayList<Employee1> employees = new ArrayList<>();
              employees.add(new Employee1(101, "John"));
              employees.add(new Employee1(103, "Alice"));
              employees.add(new Employee1(102, "Bob"));

              System.out.println("Unsorted ArrayList:");
              System.out.println(employees);

              Collections.sort(employees, new Employee1IdComparator());

              System.out.println("\nSorted ArrayList by ID:");
              System.out.println(employees);
          }
}
```

Output:

```
Unsorted ArrayList:
[Employee{id=101, name='John'}, Employee{id=103, name='Alice'},
Employee{id=102, name='Bob'}]

Sorted ArrayList by ID:
[Employee{id=101, name='John'}, Employee{id=102, name='Bob'},
Employee{id=103, name='Alice'}]
```

Sortedarraylist using comparable interface

```
package com.codingchallenge4;
import java.util.ArrayList;
import java.util.Collections;


class Employee implements Comparable<Employee> {
    private int id;
```

```java
    private String name;

    public Employee(int id, String name) {
        this.id = id;
        this.name = name;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    @Override
    public int compareTo(Employee other) {
        return Integer.compare(this.id, other.id);
    }

    @Override
    public String toString() {
        return "Employee{id=" + id + ", name='" + name + "'}";
    }
}

      public class SortArrayListUsingComparable {
          public static void main(String[] args) {

              ArrayList<Employee> employees = new ArrayList<>();
              employees.add(new Employee(101, "John"));
              employees.add(new Employee(103, "Alice"));
              employees.add(new Employee(102, "Bob"));

              System.out.println("Unsorted ArrayList:");
              System.out.println(employees);

              Collections.sort(employees);

              System.out.println("\nSorted ArrayList by ID:");
              System.out.println(employees);
          }
}
```

Output:

```
Unsorted ArrayList:
[Employee{id=101, name='John'}, Employee{id=103, name='Alice'},
Employee{id=102, name='Bob'}]

Sorted ArrayList by ID:
[Employee{id=101, name='John'}, Employee{id=102, name='Bob'},
Employee{id=103, name='Alice'}]
```