

JAVA ASSIGNMENT WEEK – 4

1. Write a Java program to join two array lists.

```
import java.util.ArrayList;
public class JoinTwoArraylist {
    public static void main(String[] args) {
        ArrayList al1 = new ArrayList();
        ArrayList al2 = new ArrayList();

        al1.add(1);
        al1.add(2);
        al1.add(3);
        al1.add(4);
        al1.add(5);

        al2.add(6);
        al2.add(7);
        al2.add(8);
        al2.add(9);
        al2.add(0);

        System.out.println("First ArrayList:"+al1);
        System.out.println("Second ArrayList:"+al2);

        al1.addAll(al2);
        System.out.println("Joint ArrayList:"+al1);

        //      System.out.println("Joint Arraylist:"+ (al1+al2));
    }
}
```

2. Write a Java program to retrieve but does not remove, the first element of a linked list.

```
import java.util.LinkedList;
public class FirstElementOfLinkedList {
    public static void main(String[] args) {
        LinkedList ll = new LinkedList();
        ll.add(10);
        ll.add(2);
        ll.add(3);
        ll.add(4);
        ll.add(5);
        ll.add(6);

        System.out.println("LinkedList:"+ll);

        System.out.println("First Element is:"+ll.getFirst());
    }
}
```

3. Write a Java program to check if a particular element exists in a linked list.

```
import java.util.LinkedList;

public class ElementInLinkedList {
    public static void main(String[] args) {
        LinkedList ll = new LinkedList();
        ll.add(10);
        ll.add(2);
        ll.add(3);
        ll.add(4);
        ll.add(5);
        ll.add(6);
        System.out.println("LinkedList:"+ll);

        if(ll.contains(5)){
            System.out.println("Element 5 exist in the LinkedList");
        }
        else {
            System.out.println("Given element does not exist in the
LinkedList");
        }
    }
}
```

4. Write a Java program to remove all of the elements from a hash set.

```
import java.util.HashSet;

public class RemoveAllFromHashset {
    public static void main(String[] args) {
        HashSet set = new HashSet();
        set.add(1);
        set.add(2);
        set.add(3);
        set.add(4);
        set.add(5);

        System.out.println("HashSet:"+set);
        set.removeAll(set);
        System.out.println("HashSet after removeAll operation:"+set);
    }
}
```

5. Write a Java program to compare two hash set.

```
import java.util.HashSet;

public class CompareTwoHashSets {
    public static void main(String[] args) {
        HashSet hs1 = new HashSet<>();
        hs1.add(10);
        hs1.add(20);
        hs1.add(90);
        hs1.add(40);
        hs1.add(70);

        HashSet hs2 = new HashSet<>();
        hs2.add(10);
        hs2.add(20);
        hs2.add(30);
        hs2.add(40);
        hs2.add(50);

        System.out.println("HashSet1:"+hs1);
        System.out.println("HashSet2:"+hs2);
        if (hs1.equals(hs2)) {
            System.out.println("Both the Hashsets are equal");
        }
        else{
            System.out.println("Both the Hashsets are not equal");
        }
    }
}
```

7. Write a Java program to retrieve and remove the last element of a tree set.

```
import java.util.TreeSet;

public class LastElementTreeSet {
    public static void lastElement(){
        TreeSet ts = new TreeSet<>();
        ts.add(1);
        ts.add(2);
        ts.add(3);
        ts.add(4);
        ts.add(5);
        ts.add(6);
        ts.add(7);

        System.out.println("Last element of the TreeSet is: "+ts.last());
    }

    public static void main(String[] args) {
        lastElement();
    }
}
```

8. Write a Java program to convert a Priority Queue elements to a string representation.

```
import java.util.PriorityQueue;

public class PriorityQueueToString {
    public static void main(String[] args) {
        PriorityQueue pq = new PriorityQueue<>();
        pq.add("Shah");
        pq.add("Vimal");
        pq.add("Arun");
        pq.add("Karthi");
        pq.add("Steve");

        System.out.println("Priority queue:"+pq);

        String str = pq.toString();
        System.out.println("String Representation:"+str);
    }
}
```

9. Write a Java program to get a collection view of the values contained in this map.

```
import java.util.HashMap;

public class CollectionViewOfMap {
    public static void main(String[] args) {
        HashMap map = new HashMap<>();
        map.put(1,"obj1");
        map.put(2,"obj2");
        map.put(3,"obj3");
        map.put(4,"obj4");
        map.put(5,"obj5");

        System.out.println("HashMap view:"+map);
        System.out.println("Collection view of HashMap:"+map.values());
    }
}
```

10. Write a Java program to get a set view of the keys contained in this map.

```
import java.util.HashMap;

public class SetViewOfMap {
    public static void main(String[] args) {
        HashMap map = new HashMap<>();
        map.put(1,"obj1");
        map.put(2,"obj2");
        map.put(3,"obj3");
        map.put(4,"obj4");
        map.put(5,"obj5");

        System.out.println("HashMap view:"+map);
        System.out.println("Collection view of HashMap:"+map.keySet());
    }
}
```

11. Write a Java program to get a key-value mapping associated with the least key greater than or equal to the given key. Return null if there is no such key.

```
import java.util.Scanner;
import java.util.TreeMap;

public class Q11 {
    public static void main(String[] args) {
        TreeMap map = new TreeMap();
        map.put(10, "violet");
        map.put(25, "indigo");
        map.put(37, "blue");
        map.put(48, "green");
        map.put(59, "yellow");
        map.put(61, "orange");
        map.put(73, "red");

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the key to search:");
        int n = sc.nextInt();

        System.out.println("The least key-value mapping greater than or
equal to the given key is "+map.ceilingEntry(n));
    }
}
```

12. Write a Java program to get the least key greater than or equal to the given key. Returns null if there is no such key.

```
import java.util.Scanner;
import java.util.TreeMap;

public class Q12 {
    public static void main(String[] args) {
        TreeMap map = new TreeMap();
        map.put(10, "violet");
        map.put(25, "indigo");
        map.put(37, "blue");
        map.put(48, "green");
        map.put(59, "yellow");
        map.put(61, "orange");
        map.put(73, "red");

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the key to search:");
        int n = sc.nextInt();

        System.out.println("The least key greater than or equal to the
given key is "+map.ceilingKey(n));
    }
}
```