(Naima
IT-22001)

1 ⊞ kth smallest element java

```
import java.util.*;
public class nthSmallestElement {
  public static int findnthSmallest(List<Integere >list,
                                                  int k)
  {
   collection.sort(list);
   returen list.get(k-1);
  }
  public static void main(String[]args){
  List <Integere> neumbere = Arreay.asList(2,4,1,3,9);
  Int k= 1;
  System.out.preintln ("kth Smallest:"+find nthSmallest
                                           (number,k));

  }
}
```

2 ⊞ WordFreqeuency Map.java

```
import java.util.*;
public class wordFrequencyMap{
 public static void main(String[]args){
   String text = "naima rafatul kulfip ilma";
   TreeMap<String,Integere> freqMap = new TreeMap<>();
   for(string word: text split(" ")){
   freqMap.put (word, freqMap.get OrDefault(word,0)
                                               +1);
   }
 System.out.preintln(freqMap);
}}
```

3 QueueStackUsingPiorityQueue.java

```
import java.util.*;
public class QueueStackUsingPriorityQueue{
  static class PQueue {
  private PriorityQueue<int[]> queue=new PriorityQu
     <>(Comparator.comparingInt(a->a[1]));
  private int time=0;
  public void offer(int val){
   queue.offer(new int[]{val,time++});
  }
  public int poll(){
    return queue.poll()[0];
  }
 }
 static class PQStack{
  private PriorityQueue<int[]> stack=new
        priorityQueue<>((a,b)-> b[1]-a[1]);
     private int time=0;
  public void push(int val){
   stack.offer(new int[]{val,time++}); }
 public int pop(){
  return stack.poll()[0];
 }
 }
 public static void main(String[]args){
      PQQueue q= new PQQueue();
q. offer (1); q. offer(2); q. offer (3);
System.out.println("Queue poll: "+q.poll());
PQStack s = new PQStack();
s.push (1); s.push(2); s.push (3);
System.out.println(" Stack pop: "+s.pop());
}
}
```

4] StudentTreeMap.Java

```
import java.util.*;
class student{
  string name;
  int age;
  public student(string name, int age){
    this.name =name;
    this.age = age;
  }
  public String toString(){
    return name+" ("+age+")";
  }
}
public class StudentTreeMap{
 public static void main(String []args){
 TreeMap <Integer, Student> students = new TreeMap<>();
   students.put(1110, new student(" Naima", 29));
   students.put(1111, new student(" Kuldip", 28));
```

```
    fore(Map. Entry <Integerr, Student >entry:
                 students . entrySet ()) {
     system.out. println (entry. getkey ()+ " => "+
                            entry. get value());
    }
  }
}
```

5] Linked_list Equality. java

```
import java . util .*;
public class LinkedListEquality {
  public staticvoid main (String [] args) {
   Linkedlist <Integere> List1 = new linkedList<>;
              (Arrays. as List (1,2,3));
  Linkedlist <Integere>List 2 = new Linkedlist <>
              (Arrays. asList (4,2,3));
   System.out. println ("lists equal ?" + list1.equa
                                          (list 2));
  }
}
```

6/ EmployeeDepartmentMap.java

```
import java.util.*;
public class EmployeeDepartmentMap{
  public static void main(String [] args){
   HashMap<Integer, String> employeeMap = new
                                    HashMap<>();
    employeeMap.put(22001, "Naima");
     employeeMap.put(22002, "Rafatul");
     employeeMap.put(22018," Kuldip");
  for(Map.Entry<Integere, String>entry: employeeMap.
                              entrySet()){

     system.out.println("ID:"+ entry.getKey()+ "=> Dept.
                             +entry.getValue());

     }
   }
}
```