Treeset pgms

Write a program to remove duplicate entries from an array.

The easiest way to remove duplicate entries from the given array is, create TreeSet object and add array entries to the TreeSet. Since the set doesnot support duplicate entries, you will get only unique elements left with TreeSet.

```
public static void main(String a[]){
        String[] strArr = {"one","two","three","four","four","five"};
        //convert string array to list
        List<String> tmpList = Arrays.asList(strArr);
        //create a treeset with the list, which eliminates duplicates
        TreeSet<String> unique = new TreeSet<String>(tmpList);
        System.out.println(unique);
    }
```

Output:

```
[five, four, one, three, two]
```

How to create a TreeSet with comparator?

```
public class MySetWithCompr {

    public static void main(String a[]){

        TreeSet<String> ts = new TreeSet<String>(new MyComp());
        ts.add("RED");
```

```
        ts.add("ORANGE");
        ts.add("BLUE");
        ts.add("GREEN");
        System.out.println(ts);
    }
}

class MyComp implements Comparator<String>{

    @Override
    public int compare(String str1, String str2) {
        return str1.compareTo(str2);
    }

}
```

Create TreeSet with comparator by user define objects.

```
public class MyCompUserDefine {

    public static void main(String a[]){
        //By using name comparator (String comparison)
        TreeSet<Empl> nameComp = new TreeSet<Empl>(new MyNameComp());
        nameComp.add(new Empl("Ram",3000));
        nameComp.add(new Empl("John",6000));
        nameComp.add(new Empl("Crish",2000));
        nameComp.add(new Empl("Tom",2400));
        for(Empl e:nameComp){
            System.out.println(e);
        }
```

```
        System.out.println("============================");
        //By using salary comparator (int comparison)
        TreeSet<Empl> salComp = new TreeSet<Empl>(new MySalaryComp());
        salComp.add(new Empl("Ram",3000));
        salComp.add(new Empl("John",6000));
        salComp.add(new Empl("Crish",2000));
        salComp.add(new Empl("Tom",2400));
        for(Empl e:salComp){
            System.out.println(e);
        }
    }
}

class MyNameComp implements Comparator<Empl>{

    @Override
    public int compare(Empl e1, Empl e2) {
        return e1.getName().compareTo(e2.getName());
    }
}

class MySalaryComp implements Comparator<Empl>{

    @Override
    public int compare(Empl e1, Empl e2) {
        if(e1.getSalary() > e2.getSalary()){
            return 1;
        } else {
            return -1;
        }
```

```
    }
}

class Empl{

    private String name;
    private int salary;

    public Empl(String n, int s){
        this.name = n;
        this.salary = s;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getSalary() {
        return salary;
    }
    public void setSalary(int salary) {
        this.salary = salary;
    }
    public String toString(){
        return "Name: "+this.name+"-- Salary: "+this.salary;
    }
}
```

Output:

```
Name: Crish-- Salary: 2000
Name: John-- Salary: 6000
Name: Ram-- Salary: 3000
Name: Tom-- Salary: 2400
==========================
Name: Crish-- Salary: 2000
Name: Tom-- Salary: 2400
Name: Ram-- Salary: 3000
Name: John-- Salary: 6000
```

How to get subset from sorted set?

```
public class MySetSublist {

    public static void main(String a[]){

        TreeSet<String> ts = new TreeSet<String>(new MyStrComp());
        ts.add("RED");
        ts.add("ORANGE");
        ts.add("BLUE");
        ts.add("GREEN");
        ts.add("WHITE");
        ts.add("BROWN");
        ts.add("YELLOW");
        ts.add("BLACK");
        System.out.println(ts);
```

```
        Set<String> subSet = ts.subSet("GREEN", "WHITE");
        System.out.println("sub set: "+subSet);
        subSet = ts.subSet("GREEN", true, "WHITE", true);
        System.out.println("sub set: "+subSet);
        subSet = ts.subSet("GREEN", false, "WHITE", true);
        System.out.println("sub set: "+subSet);
    }
}

class MyStrComp implements Comparator<String>{

    @Override
    public int compare(String str1, String str2) {
        return str1.compareTo(str2);
    }

}
```

Output:

```
[BLACK, BLUE, BROWN, GREEN, ORANGE, RED, WHITE, YELLOW]
sub set: [GREEN, ORANGE, RED]
sub set: [GREEN, ORANGE, RED, WHITE]
sub set: [ORANGE, RED, WHITE]
```

How to get least value element from a set?

```
public class MyLeastElementInSet {

    public static void main(String a[]){

        TreeSet<Empl1> salComp = new TreeSet<Empl1>(new MySalCompr());
        salComp.add(new Empl1("Ram",3000));
        salComp.add(new Empl1("John",6000));
        salComp.add(new Empl1("Crish",2000));
        salComp.add(new Empl1("Tom",2400));
        System.out.println("Least salary emp: "+salComp.first());
    }
}
```

How to get highest value element from a set?

```
 System.out.println("Highest salary emp: "+salComp.last());
```

How to avoid duplicate user defined objects in TreeSet?

```
public class MyUserDuplicates {

    public static void main(String a[]){

        Set<Emp> ts = new TreeSet<Emp>(new EmpComp());
        ts.add(new Emp(201,"John",40000));
        ts.add(new Emp(302,"Krish",44500));
        ts.add(new Emp(146,"Tom",20000));
        ts.add(new Emp(543,"Abdul",10000));
        ts.add(new Emp(12,"Dinesh",50000));
        //adding duplicate entry
        ts.add(new Emp(146,"Tom",20000));
```

```
        //check duplicate entry is there or not
        for(Emp e:ts){
            System.out.println(e);
        }
    }
}

class EmpComp implements Comparator<Emp>{

    @Override
    public int compare(Emp e1, Emp e2) {
        if(e1.getEmpId() == e2.getEmpId()){
            return 0;
        } if(e1.getEmpId() < e2.getEmpId()){
            return 1;
        } else {
            return -1;
        }
    }
}

class Emp {

    private int empId;
    private String empName;
    private int empSal;

    public Emp(int id, String name, int sal){
        this.empId = id;
        this.empName = name;
```

```
        this.empSal = sal;
    }

    public int getEmpId() {
        return empId;
    }

    public void setEmpId(int empId) {
        this.empId = empId;
    }

    public String getEmpName() {
        return empName;
    }

    public void setEmpName(String empName) {
        this.empName = empName;
    }

    public int getEmpSal() {
        return empSal;
    }
    public void setEmpSal(int empSal) {
        this.empSal = empSal;
    }

    public String toString(){
        return empId+" : "+empName+" : "+empSal;
    }
}
```