Comparing Collection Based Classes.

They are 7 collection based classes:-

- ArrayList
- ArrayDeque
- LinkedList
- PriorityQueue
- TreeSet
- HashSet
- LinkedHashSet

ArrayList

1. It preserve the order of insertion.

2. It allow duplicate element.

Code:-

```
import java.util.*;
public class CollectionsDemo {
    public static void main(String[] args) {
        ArrayList coll = new ArrayList();
        coll.add(44);
        coll.add(22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add(22);
        System.out.println("Elements : "+coll);
    }
}
```

Output:-

```
Elements : [44, 22, 99, 55, 11, 22]
```

ArrayDeque

1. It preserve the order of insertion.

2. It allow duplicate element.

Code:-

```
import java.util.*;
public class CollectionsDemo{
    public static void main(String[] args){
        ArrayDeque coll = new ArrayDeque();
        coll.add(44);
        coll.add(22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add(22);
        System.out.println("Elements : "+ coll);
    }
}
```

Output:-

Elements : [44, 22, 99, 55, 11, 22]

LinkedList

1. It allow duplicate element.

2. It preserve the order of insertion.

Code:-

```
import java.util.*;
public class CollectionsDemo{
    public static void main(String[] args){
        LinkedList coll = new LinkedList();
        coll.add(44);
        coll.add(22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add(22);
        System.out.println("Elements : "+ coll);
    }
}
```

Output:-

Elements : [44, 22, 99, 55, 11, 22]

PriorityQueue

1. It does not preserve the order of insertion.

2. It allow duplicate element.

Code:-

```
import java.util.*;
public class CollectionsDemo{
    public static void main(String[] args){
        PriorityQueue coll = new PriorityQueue();
        coll.add(44);
        coll.add(22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add(22);
        System.out.println("Elements :"+ coll);
    }
}
```

Output:-

```
Elements : [11, 22, 22, 55, 44, 99]
```

TreeSet

1. It does not preserve the order of insertion.

2. It does not allow duplicate element.

Code:-

```
import java.util.*;
public class CollectionsDemo {
    public static void main (String[] args) {
        TreeSet coll = new TreeSet();
        coll.add (44);
        coll.add (22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add (22);
        System.out.println (" Elements : "+ coll);
    }
}
```

Output:-

Elements : [11, 22, 44, 55, 99]

HashSet

1. It does not allow duplicate element.

2. It does not preserve the order of insertion.

Code:-

```
import java.util.*;
public class CollectionsDemo{
    public static void main(String[] args){
        HashSet coll = new HashSet();
        coll.add(44);
        coll.add(22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add(22);
        System.out.println("Elements : "+ coll);
    }
}
```

Output:-

```
Elements : [99, 22, 55, 11, 44]
```

LinkedHashSet

1. It preserve the order of insertion.

2. It does not allow duplicate element.

Code:-

```
import java.util.*;
Public class CollectionsDemo{
    Public static void main(String[] args){
        LinkedHashSet coll = new LinkedHashSet();
        coll.add(44);
        coll.add(22);
        coll.add(99);
        coll.add(55);
        coll.add(11);
        coll.add(22);
        System.out.println("Elements :" + coll);
    }
}
```

Output:-

Elements : [44, 22, 99, 55, 11]