**SET**

Set is an Interface that extends Collection interface

It cannot hold duplicate data.ie., only unique values

It may or may not contain the insertion order

**Classes Implementing SET Interface**

* HashSet-Based on hashing Algorithm(random)
* TreeSet-Based on ASCII order(Alphabetic/numerical order)
* LinkedHashSet-Based on Insertion Order(First One First)

**Methods of Set**

* size()
* add(class ele)
* remove(class ele)
* clear()
* get(int index)—This method is not available in SET

**Declaring Set**

**Set<datatype> setname = new HashSet<datatype>();**

**Set<datatype> setname = new TreeSet<datatype>();**

**Set<datatype> setname = new LinkedHashSet<datatype>();**

**Note: datatype name should be the corresponding Wrapper Class**

**DEMO PROGRAM FOR SET DECLARATION**

Set<String> Colors\_Set = **new** HashSet<String>();

Colors\_Set.add("Red");

Colors\_Set.add("Green");

System.out.println(Colors\_Set); //[Green,Red]

Set<String> tree\_Set = **new** TreeSet<String>(Colors\_Set);

System.out.println(tree\_Set); //[Green,Red]

Set<String> linkedhash\_Set = **new** LinkedHashSet<String>(Colors\_Set);

System.out.println(linkedhash\_Set); //[Red,Green]

**ITERATING THROUGH SET**

        Set<String> cities\_Set = **new** HashSet<String>();

        cities\_Set.add("Bangaluru");

        cities\_Set.add("Pune");

        cities\_Set.add("Hyderabad");

        // Print the set contents

        System.out.println(cities\_Set);

        // print the set contents using forEach loop

**for**(String val : cities\_Set) {

            System.out.print(val);

        }

**HOW TO CONVERT SET TO ARRAY**

Set<String> setOfColors= **new** HashSet<>();

setOfColors.add("Red");

setOfColors.add("Green");

setOfColors.add("Blue");

//print the set

System.out.println(setOfColors);

//convert Set to Array using toArray () method

String[] colors\_Array = setOfColors.toArray(**new** String[setOfColors.size()]);

//print the Array

System.out.println("Set converted to Array:" + Arrays.toString(colors\_Array));

**HOW TO CONVERT ARRAY TO SET**

**Note: First, we need to convert array into list using Arrays.asList() and then pass this list as an argument to set constructor.**

Integer[] numArray = {10,50,40,20,60,30,80,70};

System.out.println("The input array:" + Arrays.toString(numArray));

List<Integer> lst = new ArrayList<Integer>(Arrays.asList(numArray));

Set<Integer> st =new HashSet<Integer>(lst);

**CONVERT SET TO LIST :**

Set<String> strSet= **new** HashSet<String>();

strSet.add("one");

strSet.add("two");

strSet.add("three");

//declare an ArrayList

List<String> strList = **new** ArrayList<String>(strSet);

//print the ArrayList

System.out.println("The ArrayList from set : " + strList);

**CONVERT LIST TO SET**

List<String> strList = **new** ArrayList<String>();

strList.add("one");

strList.add("two");

strList.add("three");

Set<String> strSet= **new** HashSet<String>(strList);

System.out.println(strSet);

**SORT A SET**

**Note: We cannot sort a SET directly,need to follow an indirect approach.**

**TreeSet provides the ordered set by default.For the rest, first we need to convert set to list.Then use Collections.sort() method and then convert back list to set.**

Set<Integer> NumSet = **new** LinkedHashSet<Integer>();

NumSet.add(47);

NumSet.add(74);

NumSet.add(80);

System.out.println(Numset);

List<Integer> Numlst = new ArrayList<Integer>(NumSet);

Collections.sort(Numlst);

Set<Integer> NumSet = **new** LinkedHashSet<Integer>(Numlst);

System.out.println(Numlst);