***Dt : 8/11/2022***

***1.Set<E>:***

***=>Set<E> is an interface from java.util package and which***

***organizes elements without index values,and which cannot hold***

***duplicate elements.***

***=>The following are some important methods of Set<E>:***

***public abstract int size();***

***public abstract boolean isEmpty();***

***public abstract boolean contains(java.lang.Object);***

***public abstract boolean add(E);***

***public abstract boolean remove(java.lang.Object);***

***public abstract boolean containsAll***

***(java.util.Collection<?>);***

***public abstract boolean addAll***

***(java.util.Collection<? extends E>);***

***public abstract boolean retainAll***

***(java.util.Collection<?>);***

***public abstract boolean removeAll***

***(java.util.Collection<?>);***

***public abstract void clear();***

***public default java.util.Spliterator<E> spliterator();***

***public abstract java.util.Iterator<E> iterator();***

***public abstract java.lang.Object[] toArray();***

***public abstract <T> T[] toArray(T[]);***

***-------------------------------------------------------***

***=>The following are the implementation classes of Set<E>:***

***(a)HashSet<E>***

***(b)LinkedHashSet<E>***

***(c)TreeSet<E>***

***(a)HashSet<E>:***

***=>HashSet<E> organizes elements without any order.***

***(b)LinkedHashSet<E>:***

***=>LinkedHashSet<E> organizes elements in insertion order.***

***(c)TreeSet<E>:***

***=>TreeSet<E> organizes elements automatically in ascending***

***order***

***-----------------------------------------------------***

***Ex-program : DemoGeneric.java***

***package maccess;***

***import java.util.\*;***

***public class DemoSet1 {***

***public static void main(String[] args) {***

***Scanner s = new Scanner(System.in);***

***String c = null;***

***Set<Integer> ob = null;***

***try(s;){***

***try {***

***while(true) {***

***System.out.println("====Choice====");***

***System.out.println***

***("1.HashSet\n2.LinkedHashSet\n3.TreeSet\n4.exit");***

***System.out.println("Enter the choice:");***

***switch(s.nextInt())***

***{***

***case 1:***

***ob = new HashSet<Integer>();***

***c = "HashSet<E>";***

***break;***

***case 2:***

***ob = new LinkedHashSet<Integer>();***

***c = "LinkedHashSet<E>";***

***break;***

***case 3:***

***ob = new TreeSet<Integer>();***

***c = "TreeHashSet<E>";***

***break;***

***case 4:***

***System.out.println("Set operation Stopped...");***

***System.exit(0);***

***default : System.out.println("Invalid choice");***

***continue; //skip the below lines from the Iteration***

***}//end of switch***

***System.out.println("\*\*\*\*perform operations ob "+c+"\*\*\*\*");***

***xyz:***

***while(true) {***

***System.out.println("====Choice====");***

***System.out.println("1.add\n2.remove\n3.exit");***

***System.out.println("Enter the Choice:");***

***switch(s.nextInt())***

***{***

***case 1:***

***System.out.println("Enter the ele:");***

***ob.add(new ~~Integer~~(s.nextInt()));***

***System.out.println(ob.toString());***

***break;***

***case 2:***

***if(ob.isEmpty()) {***

***System.out.println("Set<E> is empty...");***

***}else {***

***System.out.println("Enter the ele to be removed:");***

***Integer ele = new ~~Integer~~(s.nextInt());***

***if(ob.remove(ele)) {***

***System.out.println("Ele removed Successfully...");***

***System.out.println(ob.toString());***

***}else {***

***System.out.println("Ele not available...");***

***}***

***}***

***break;***

***case 3:***

***System.out.println("Operation on "+c+" Stopped");***

***break xyz;***

***default:***

***System.out.println("Invalid choice...");***

***}//end of switch***

***}//end of loop***

***}//end of loop***

***}catch(Exception e) {e.printStackTrace();}***

***}//end of try***

***}***

***}***

***o/p:***

***====Display from HashSet=====***

***[NIT, 121, JAVA]***

***====Display from HashSet=====***

***[Java, Hyd, Nit]***

***===================================================***