ConcurrentSkipListSet in Java with Examples

The **ConcurrentSkipListSet** class in Java is a part of the [Java Collection Framework](https://www.geeksforgeeks.org/collections-in-java-2/) and implements the **Collection interface** and the **AbstractSet class**. It provides a scalable and concurrent version of [NavigableSet in Java](https://www.geeksforgeeks.org/navigableset-java-examples/). The implementation of ConcurrentSkipListSet is based on ConcurrentSkipListMap. The elements in ConcurrentSkipListSet are sorted by default in their natural ordering.

**Class Hierarchy:**

java.lang.Object

↳ java.util.AbstractCollection<E>

↳ java.util.AbstractSet<E>

↳ Class ConcurrentSkipListSet<E>

**Syntax:**

public class ConcurrentSkipListSet<E>

extends AbstractSet<E>

implements NavigableSet<E>, Cloneable, Serializable

Where **E** is the type of elements maintained

by this collection

**Constructors in Java ConcurrentSkipListSet:**

* **ConcurrentSkipListSet()**: This constructor is used to construct an empty set.
* **ConcurrentSkipListSet(Collection<E> c)**: This constructor is used to construct a set with the elements of the Collection passed as the parameter.
* **ConcurrentSkipListSet(Comparator<E> comparator)**: This constructor is used to construct a new, empty set that orders its elements according to the specified comparator.
* **ConcurrentSkipListSet(SortedSet<E> s)**: This constructor is used to construct a new set containing the same elements and using the same ordering as the specified sorted set.

Below is a sample program to illustrate ConcurrentSkipListSet in Java:

|  |
| --- |
| // Java program to demonstrate ConcurrentSkipListSet    import java.util.concurrent.ConcurrentSkipListSet;    class ConcurrentSkipListSetLastExample1 {      public static void main(String[] args)      {            // Initializing the set using ConcurrentSkipListSet()          ConcurrentSkipListSet<Integer>              set = new ConcurrentSkipListSet<Integer>();            // Adding elements to this set          set.add(78);          set.add(64);          set.add(12);          set.add(45);          set.add(8);            // Printing the ConcurrentSkipListSet          System.out.println("ConcurrentSkipListSet: "                             + set);            // Initializing the set using          // ConcurrentSkipListSet(Collection)          ConcurrentSkipListSet<Integer>              set1 = new ConcurrentSkipListSet<Integer>(set);            // Printing the ConcurrentSkipListSet1          System.out.println("ConcurrentSkipListSet1: "                             + set1);      }  } |