JAVA.UTIL.TREESET.HEADSET METHOD

Description

The **headSet*EtoElement*** method is used to return a view of the portion of this set whose elements are strictly less than **toElement*input***.The returned set is backed by this set, so changes in the returned set are reflected in this set, and vice-versa.

Declaration

Following is the declaration for **java.util.TreeSet.headSet** method.

```
public SortedSet<E> headSet(E toElement)
```

Parameters

- **toElement** -- This is the high endpoint *exclusive* of the returned set.

Return Value

The method call returns a view of the portion of this set whose elements are strictly less than toElement.

Exception

- **ClassCastException** -- This exception is thrown if 'toElement' cannot be compared to elements currently in the set.
- **NullPointerException** -- This exception is thrown if 'toElement' is null and this set uses natural ordering, or its comparator does not permit null elements.
- **IllegalArgumentException** -- This exception is thrown if this set itself has a restricted range, and 'toElement' lies outside the bounds of the range.

Example

The following example shows the usage of java.util.TreeSet.headSet method.

```
package com.tutorialspoint;

import java.util.Iterator;
import java.util.TreeSet;

public class TreeSetDemo {
   public static void main(String[] args) {
   // creating TreeSet
   TreeSet <Integer>tree = new TreeSet<Integer>();
   TreeSet <Integer>treeheadset = new TreeSet<Integer>();

   // adding in the tree
   tree.add(12);
   tree.add(13);
   tree.add(14);
   tree.add(15);
   tree.add(16);
   tree.add(17);

   // getting values less than 15
   treeheadset = (TreeSet)tree.headSet(15);

   // creating iterator
   Iterator iterator;
   iterator = treeheadset.iterator();
```

```
   //Displaying the tree set data
   System.out.println("Tree set data: ");
   while (iterator.hasNext()){
   System.out.println(iterator.next() + " ");
   }
   }
}
```

Let us compile and run the above program, this will produce the following result.

```
Tree set data:
12
13
14
```