⇒ Linked Hash Set :→

→ It is the child class of Hash Set

→ It is exactly same as Hash Set (including constructors and methods) except the following differences.

Hash Set	Linked Hash Set
① The underlined data structure is Hash Table.	① The underlined data structure is Linked List + Hash Table.
② Insertion order is not preserved.	② preserved.
③ Introduced in 1.2 V	③ 1.4 V

⇒ In above prog. (for Hash Set Ex), if we will replace Hash Set with the Linked Hash Set the output will be

[B, C, D, Z, null, 10]

NOTE :→

⇒ In general, we can use Linked Hash Set to develop cache based applications where duplicates are not allowed and Insertion order not preserved.

⇒ Sorted Set :-

→ It is the child interface of Set interface.

→ If we want to represent a group of individual objects according to some sorting order without duplicates. then we should go for Sorted Set.

⇒

Sorted Set interface defines the following methods :-

~~① first()~~ ① Object first ()→ returns ~~elements~~ first element of the sorted set

~~② last()~~

~~③ headSet~~

② Object last ():→ It returns the last element of the sorted set.

③ SortedSet headSet (Object obj)

↳ It returns Sorted set whose elements are less than obj.

(4) SortedSet tailSet(Object obj) :→
It returns SortedSet whose elements are >= obj.

(5) SortedSet subSet(Object obj1, Object obj2)
↳ returns SortedSet whose elements are >= obj1 and < obj2

(6) Comparator comparator()
↳ returns comparator Object that discribes underlying sorting technique. If we are using default natural sorting order then we will get null.

Note:—
The default natural sorting order for Number is Ascending order and for String Object Alphabetical Order.

(1) first() ⇒ 100
(2) last() ⇒ 120
(3) headSet(106) ⇒ [100, 101, 104]
(4) tailSet(106) ⇒ [106, 110, 115, 120]
(5) subSet(101, 115);
⇒ [101, 104, 106, 110]
(6) comparator()
[null] → default

Sorted set

100
101
104
106
110
115
120