1) Comparator < ? super E >
Lower bound wild card
in which you have a comparator
with generic type of something
that is super typed by E

2) SortedSet<E> headSet(E toElement)

SorteSet<E> is the return type of the
method ~~with~~ which takes parameter of toElement
of E type.

3) E first()
~~the~~ the method first returns
E type of value / element.

4) Vector(Collection<? extends E>c)

Constructor Vector takes collection c
which accepts E or subtypes of E

5) public boolean containsAll (Collection<?> c)

Collections in java helps us store and manipulate group of objects. It simply says that collection can accept any type of object.

6) public boolean removeAll (Collection<?> c)

It can remove all ~~any type~~ element from collection irrespective of type.

7) public boolean addAll (Collection <? extends E> c)

upper bound wildcards.

This just enforces the restriction on the type of elements that can be added to the collection.

8) public void insertElement (E obj, int index)

Simple generic method return nothing and accepting object of E element.

9) public static <T extends Comparable<? super T>>
void sort <List<T> list>.

Comparable is used to sorting sequence.

List with type T elements is to be sorted by restricting wildcard to the super type of T. T has to implement comparable interface

10) Whoever implements comparable will be restricted by the super type of T

11) List of list will accept any object