<https://www.javatpoint.com/collections-in-java> \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

https://beginnersbook.com/2013/12/java-arraylist/ \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* best

INDIVIDUAL OBJECTS : USE>

**List** : duplicates are allowed and insertion order must be preserved. : arraylist, linkedlist, vector, stack.

**Set** : Insertion order not preserved and duplicates are not allowed. : Hashset, linked hash set,

**Sorted set** : sorting order is maintained but no duplicates are allowed

**Navigable set** : sorted set with navigation support, treeset

**Queue** : group of individual objects : priorityQ, blockingQ>priority blocking, linked blockingQ

GROUP OF KEY VALUE PAIRS : USE>

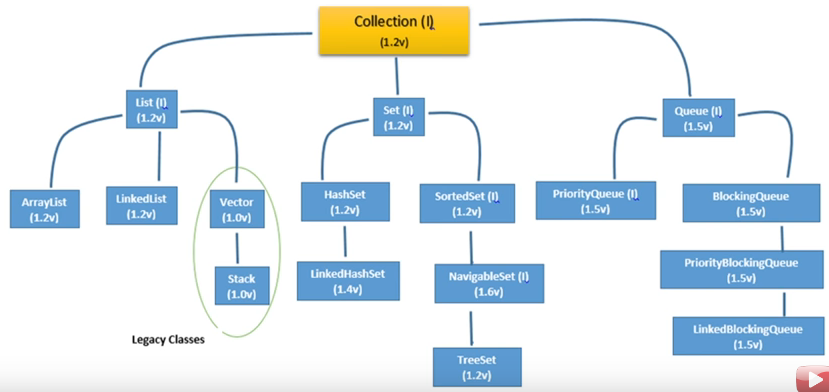
Hash map, linked hashmap

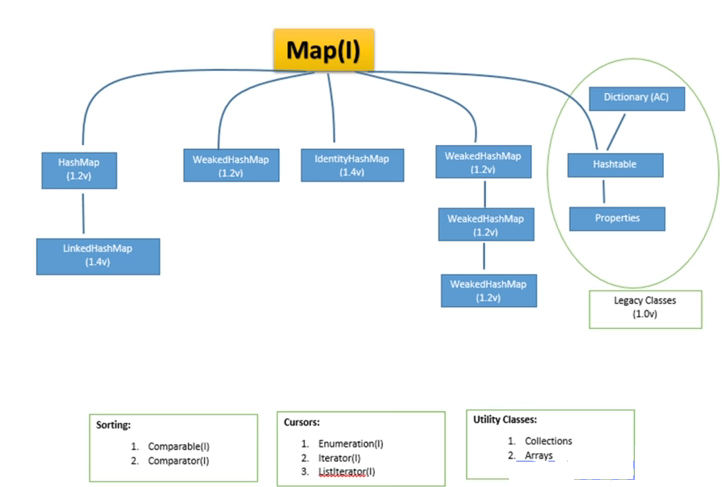
Weak hashmap, Identity hashmap

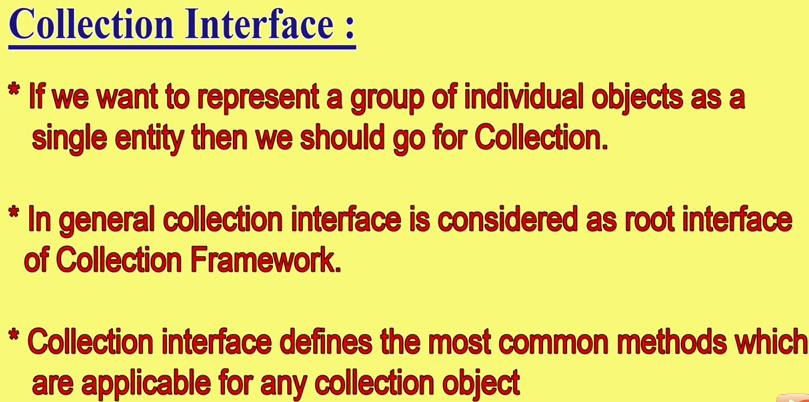
**sorted map** : If want to store key value pair with **insertion** order then use sorted map

**Navigable map** : If need navigation support

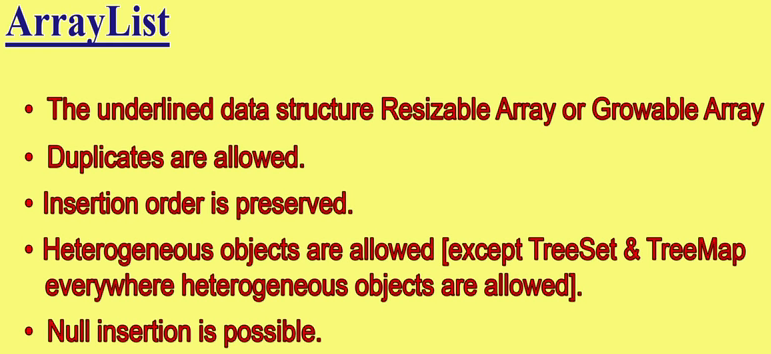
Legacy / old map : HashTable



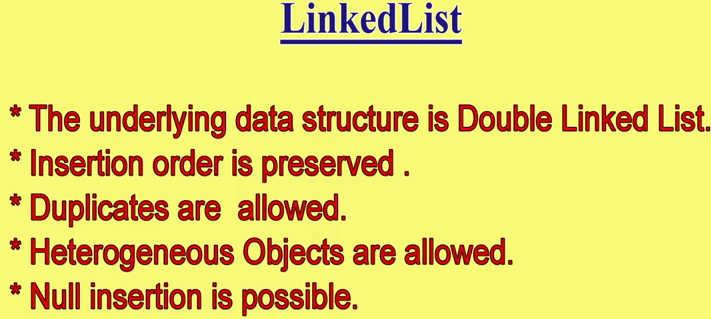


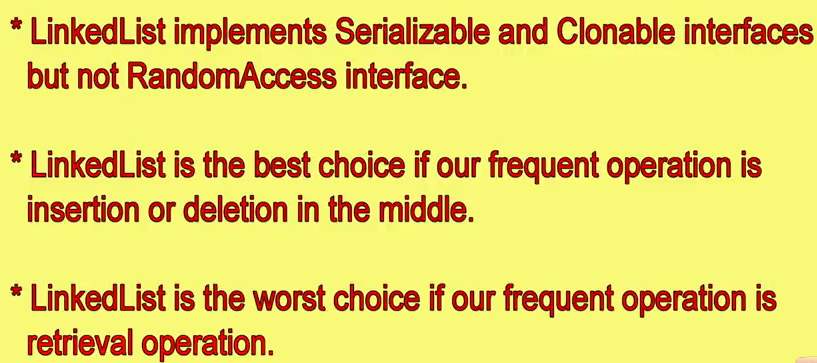
 What we are going to learn in Java Collections Framework :

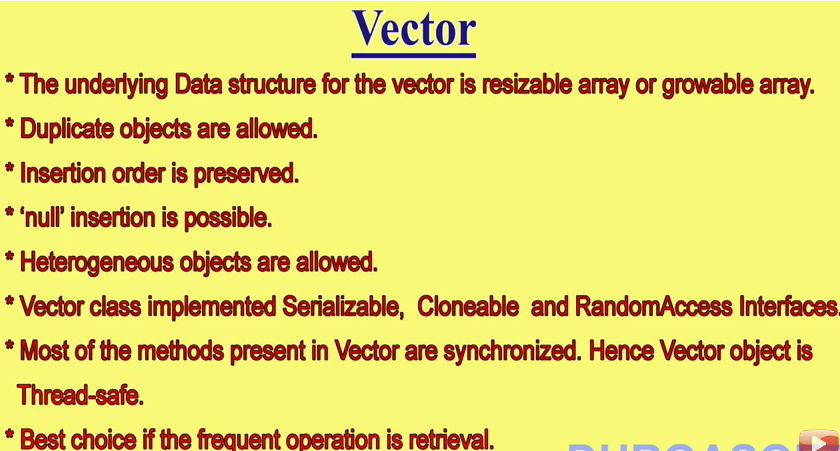
1. [ArrayList class](https://www.javatpoint.com/java-arraylist)
2. [LinkedList class](https://www.javatpoint.com/java-linkedlist)
3. [List interface](https://www.javatpoint.com/java-list)
4. [HashSet class](https://www.javatpoint.com/java-hashset)
5. [LinkedHashSet class](https://www.javatpoint.com/java-linkedhashset)
6. [TreeSet class](https://www.javatpoint.com/java-treeset)
7. [PriorityQueue class](https://www.javatpoint.com/java-priorityqueue)
8. [Map interface](https://www.javatpoint.com/java-map)
9. [HashMap class](https://www.javatpoint.com/java-hashmap)
10. [LinkedHashMap class](https://www.javatpoint.com/java-linkedhashmap)
11. [TreeMap class](https://www.javatpoint.com/TreeMap-class-in-collection-framework)
12. [Hashtable class](https://www.javatpoint.com/Hashtable-class-in-collection-framework)
13. [Sorting](https://www.javatpoint.com/Sorting-in-collection-framework)
14. [Comparable interface](https://www.javatpoint.com/Comparable-interface-in-collection-framework)
15. [Comparator interface](https://www.javatpoint.com/Comparator-interface-in-collection-framework)
16. [Properties class in Java](https://www.javatpoint.com/properties-class-in-java)

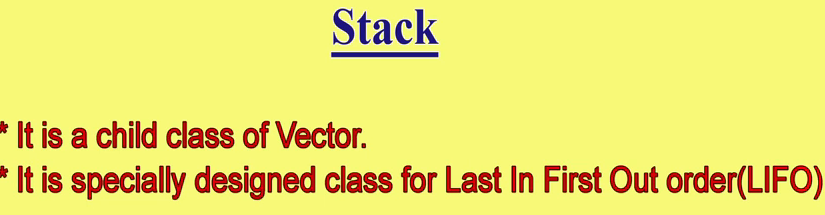


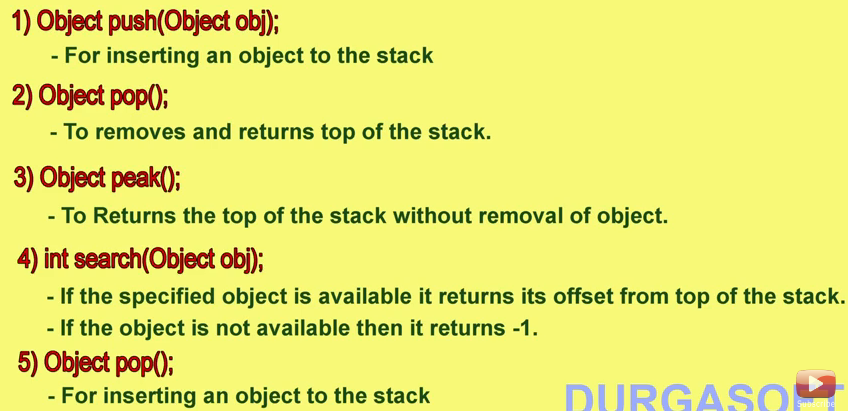


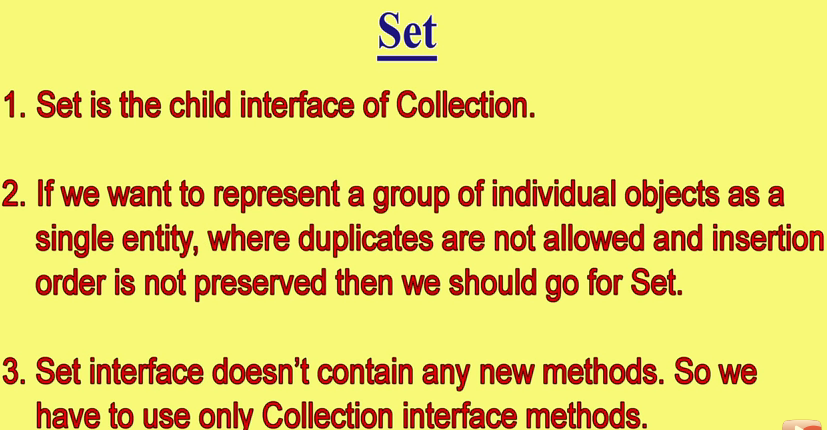


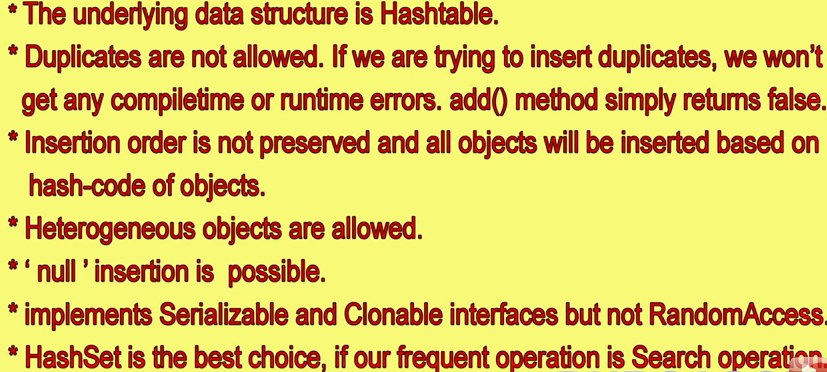


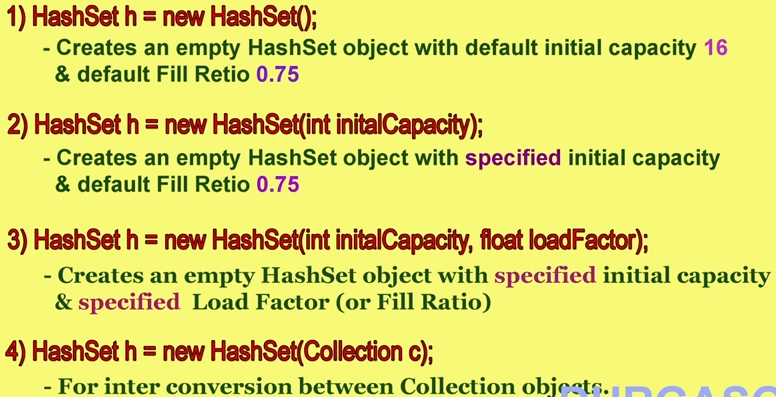


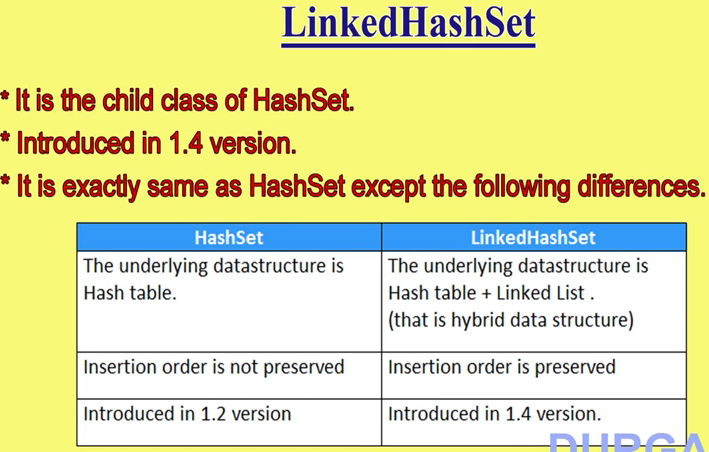


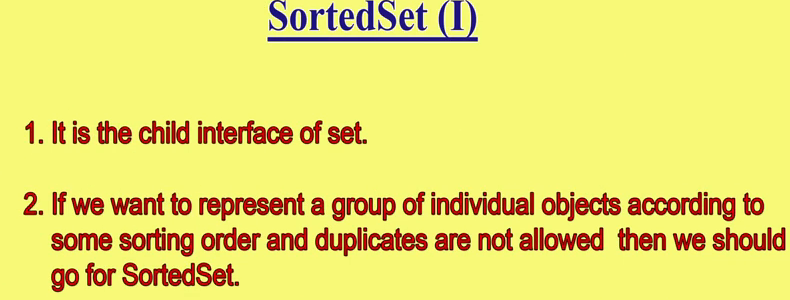
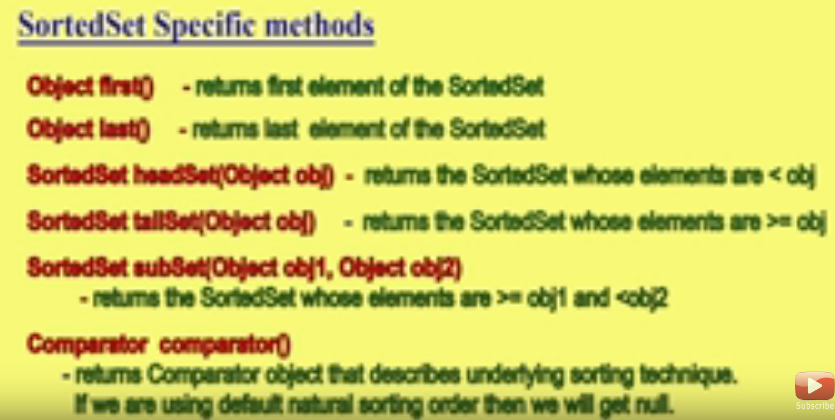




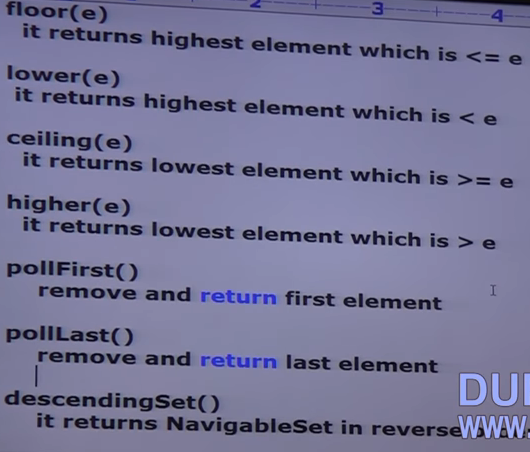


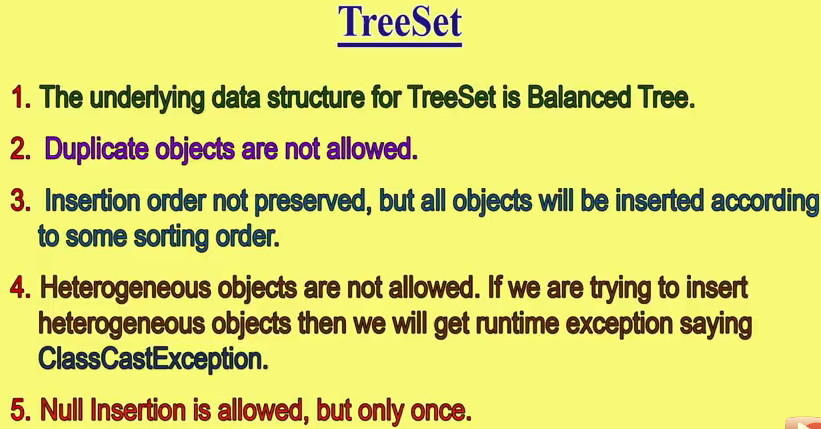


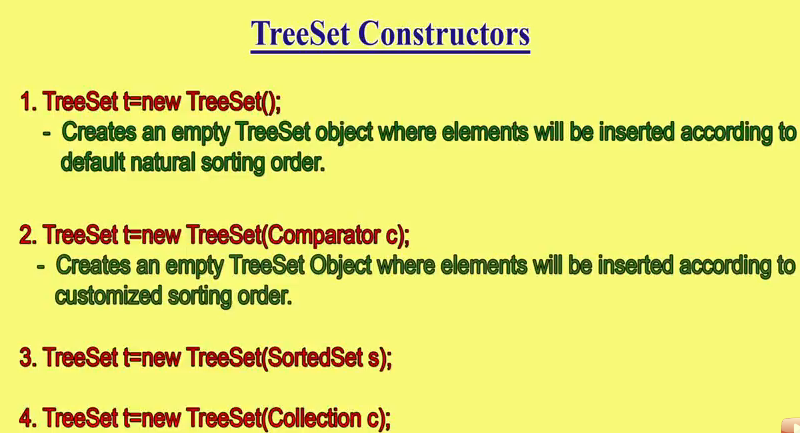


** Navigable set :**

**Navigable set allows you to navigate in set like previous element of any desired element, nest method of any desired element. Etc.**

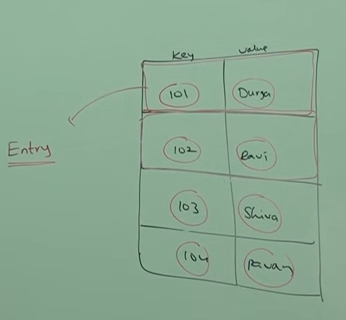




Queue : Contains first in last out behavior

Priority Q : usually Q follows fIfO order but based on our requirement we can implement our own priority order also.

**MAP** : hashmap : insertion order not preserved, it is based on hash code of keys. when we want to implement / store key values then we should use map. E.g.: student, roll no. **keys cannot be duplicate** but duplicate values are allowed.



Linked hash map :