Set extends Collection Interface

HashSet –

1. will not allow the duplicates
2. Insertion order will not be preserved
3. Insert will happen using hascode() of the objects.
4. If you add duplicate object – add() simply returns the false
5. Heterogenious objects will be stored
6. It will always allows null values
7. Works on principle of Hashtable -Underlying data structure used.
8. For Search operation – Best
9. Implements Seriliable and Cloneable
10. Null Insertion possible
11. Random access is not possible.

Constructors – 4 types.

1. HashSet h=new HashSet()

Creates an empty HastSet with initial capacity 16 and default load factor or fill ratio is 0.75

TreeSet

1. Balanced Tree- Underlying data structure
2. Duplicate Not allowed
3. Inserted order not allowed
4. Sorting order
5. Heterogenous objects not allowed ( if you add u will get Runtime Exception ClassCast Exception).
6. Null allowed but only once.

TreeSet constructors- 4 types

1. TreeSet tset=new TreeSet()

Insert the elements Using for Default Natural Sorting Order (D N S O)

1. TreeSet tset=new TreeSet(Comparator c)

* Customized sorting order -

1. TreeSet tset =new TreeSet(Collection c) ;
2. TreeSet tset =new TreeSet(SortedSet s);

**Hashmap vs Hashtable**  
1. HashMap is non synchronized. It is not-thread safe and can’t be shared between many threads without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads.  
2. HashMap allows one null key and multiple null values whereas Hashtable doesn’t allow any null key or value.  
3. HashMap is generally preferred over HashTable if thread synchronization is not needed

Why HashTable doesn’t allow null and HashMap does?  
To successfully store and retrieve objects from a HashTable, the objects used as keys must implement the hashCode method and the equals method. Since null is not an object, it can’t implement these methods. HashMap is an advanced version and improvement on the Hashtable. HashMap was created later.

A group of objects storing as Key, value pairs – then go for Map Interface.

HashMap, LinkedHashMap,TreeMap ,

HashMap – allows duplicate values , but duplicate keys are not allowed

Each Key , value pair is called as Entry in Map.

Map is not child interface of collection.

Map is considered as collection of Entries.

Put(Object key, Object value) - to add one key value pair to the Map.

If the key is already present , then old value will be replaced with new value.

If the key is not present , new object will replaces with null object .

Putall(Map) – stored all Mapped values

Map.get(key) , map.containsKey(key), map.size(), m.remove(key), map.conatinsValue(value),

Map.isEmpty();Map.clear();

keySet();🡺 Set returns

values() 🡪 Collection returns

entrySet()🡪 Entry returns

Collections views of Maps above all three methods.

Map is group of Entry objects.

Entry is inner interface of Map interface.

With out existing Map object , there is no chance of existing Entry object , Hence Entry interface defined inside Map Interface.

HashMap() works o underlying datastructure is Hashtable.

Insertion based on hashCode of keys- insertion not preserved and it is based on Hashcode of keys.

Duplicate keys not allowed

Duplicate values are allowed.

Heterogenious objects can be stored .

Nul is allowed for Key , but only once.

Null is allowed for values many times

HashMap implements Serlizable and cloneable not Rando Access Interfaces.

Search operation used for HashMap mostly – for frequent search operations.

4 constructors in HashMap()

1. HashMap m=new HashMap()

Creates and empty HashMap set with default capacity 16 and default initial ratio is 0.75 (load factor)

1. HashMap m=new HashMap(int initialCapaicty)
2. HashMap m=new HashMap(int initailCapacity, float fillratio)
3. HashMap m=new HashMap(Map m);

Q) Difference between HashMap and Hashtable

1) HashMap is not synchronized , Hashtable is synchronized

2) HashMap – multiple threads are allowed to operate – not thread safe.

3) Hashtable – Thread safe – only one thread at a time- multiple thread can’t operate

4) Perfomance is high – HashMap and Hasable performance is low.

5) Null key and Null values allowed in HashMap()

6) Null key and Null values are not allowed in Hashtable.

7) HashMap is not legacy and Hashtable is legacy .

Map m=Collections.synncronizedMap(HashMapObj); - to synchronize the HashMap object using collections.