**COLLECTIONS:**

* Collections is nothing but a group of objects represents as single unit.
* Its main purpose is to store huge amount of data.
* It provides multiple API (Application Programming Interface) (methods) to store and manipulate data.
* Collections is used to store heterogeneous data as well as homogeneous data.
* Collections are numerous (growable in size).
* We should go for collection, when we don't know the elements in advance.
* All the classes and interfaces of Collections Framework are present in java.util package.

Collection interface is extended by 3 different interfaces

1. List<Interface>
2. Set<Interface>
3. Queue<Interface>

**1.List Interface:**

List is an Interface which extends Collection Interface

* **List allows heterogeneous Objects.**
* **List allows Null objects.**
* implements **List Interface**

1. ArrayList
2. LinkedList
3. Vector

**2.Set Interface:**

* Set interface does not have index based process.
* Default capacity is 16.
* **Duplicate objects are not allowed.**
* Set is only unidirectional, so it supports only Iterator interface(cursor).
* implements Set Interface

1. Hashset
2. LinkedHashSet
3. Treeset

**1.HashSet:**

* It is a class which **implements set interface**.
* **Heterogenous objects are allowed.**
* **Duplicates are not allowed.**
* In case if we add, we won't get compile time error, it will just add once.
* **Only one NULL object is allowed.**
* Data structure is **hashtable.**
* **Insertion order is preserved** (depends on hashCode number).

**2.LinkedHashSet:**

It is a class which extends HashSet and implements set interface.

* **Heterogenous objects are allowed**
* **Duplicates are not allowed.**
* In case if we add we won't get compile time error, it will just add once
* **only one NULL object is allowed**
* Data structure is **LinkedList.**
* **Insertion order is preserved**.

**3.Treeset:**

It is a class which implements set interface.

* **Heterogenous objects are not allowed** if we add we will get class cast exception
* **Duplicates are not allowed.** In case if we add we won't get compile time error, it will just add once
* **No NULL object is allowed**
* Data structure is **tree.**
* Output is in **Sorted Order.**

**3.Queue Interface:**

* Queue is a data structure which usually follows FIFO principle (First in First out)
* implements Queue Interface

1. PriorityQueue
2. LinkedList

----------------------------------- \*\*\*\*\* ------------------------------------------------------------------------

**Map Interface:**

Map is an interface which stores the objects as a key value pair. Each key, value pair is called as **entry**. So, Map is also referred as **collection of entry objects**.

Whenever we want to represents objects as key value pair we go for maps

Map interface is implemented by 3 different classes

1. HashMap
2. LinkedHashMap
3. TreeMap

**1.HashMap:**

HashMap is a class which implements Map interface.

* **heterogenous data allowed**
* Data structure is **hashtable**
* **Duplicate keys are not allowed** but values can be duplicate
* **Random order based on hashcode**
* Only one null key is allowed and multiple null values are allowed.

**2.LinkedHashMap:**

LinkedHashMap is a class which extends HashMap and implements Map interface.

* **heterogenous data allowed**
* Data structure is **hashtable**
* **Duplicate keys are not allowed** but values can be duplicate, if we add duplicate key it replaces with original one.
* **Insertion order is Maintained**
* **only one null key is allowed** and multiple null values are allowed.

**3.TreeMap:**

TreeMap is a class which implements Map interface.

* **only homogenous data allowed**, if we add heterogeneous objects will get **class cast exception**
* Data structure is **Tree**
* **Duplicate keys are not allowed** but values can be duplicate
* **Sorting order** is maintained
* **Null keys are not allowed.**

----------------------------------- \*\*\*\*\* ------------------------------------------------------------------------