**Collection Framework hierarchy**

**All classes and interfaces part of util package.**

**Iterable : interface part of lang package**

**Collection ----🡪 interface**

**extends map doesn’t extends collection**

**Set List Queue Map -🡪 interfaces**

**HashSet ArrayList PriorityQueue HashMap**

**LinkedHashSet LinkedList LinkedList LinkedHashMap**

**TreeSet Vector TreeMap**

**Stack Hashtable**

**Set : it allow to store more than one element or data of any type. Set doesn’t allow duplicate. Under set few classes maintain the order or unorder or sorted. Set doesn’t provide index concept.**

**List : List allow to store more than one data or element. List maintain the order using index position. List allow duplicate.**

**Queue : Queue is a type of Data structure which provide a features as FIFO (First In Fist Out).**

**Map : Map is use to store key-value pairs. Key is unique and value may be duplicate.**

**Set classes**

**HashSet : HashSet is a type of set class which doesn’t allow duplicate. HashSet doesn’t maintain the order.**

**LinkedHashSet : LinkedHashSet class internally extends HashSet. LinkedHashSet doesn’t provide any extra methods. Only it maintain order.**

**TreeSet : TreeSet is a type of Set class which internally implements SortedSet interfaces and That interface extends Set interface. SortedSet provide algorithms to display the element in ascending order. In TreeSet we need store same data types values else we get exception.**

**TreeSet provided few extra method like subset, headset, tailset etc.**

**ArrayList : ArrayList is a type of List classes which allow to store any types of values.**

**Normal Array Vs ArrayList**

1. **Normal array fixed in memory. ArrayList dynamic memory**
2. **Normal array allow to store same data types of values but by nature ArrayList allow to store same as well as different types of values.**
3. **In normal array if we add or remove any element in between we need to write custom code in ArrayList provided pre defined methods.**

**LinkedList : LinkedList is a type of List in Java which internally use Node concept to store the data.**

**LinkedList mainly divided into 4 types.**

1. **Single linked list**
2. **Double linked list**
3. **Single circular linked list**
4. **Double circular linked list**

**100 ref 200 ref 300 ref**

**Null**

**Pref 100 nref pref 200 nref pref 30 nref**

**Null null**

**class Node {**

**Node ref**

**int data;**

**}**

**class Node {**

**Node pref,nref**

**int data;**

**}**

**Vector is legacy class which present with java 1.0 version.**

**In Vector all methods by default are synchronized. Vector is thread safe but slow in performance.**

**Stack : Stack is a type of Data structure which use the features as First In Last Out (FILO) or (LIFO). In Java Stack is a pre defined class which internally extends Vector.**

**Push -🡪 add the data**

**Pop() 🡪 is remove top most data**

**Peek() : it is use to check top most data present**

**Empty() : stack is empty or not**

**Search() : it will search from top to bottom start from 1.**

**Queue : Queue is a data structure which allow First In First Out. In Java Queue is an interface.**

**PriorityQueue : First In First Out base upon priority. Lower value.**

**LinkedList : First In First Out**

**Map : Map is use to store the information in the form of key-value pairs. Key is unique and value may be duplicate.**

**HashMap : unorder**

**LinkedHashMap : insertion order**

**TreeMap : ascending order as key. So key must be same data types.**

**Hashtable : legacy class. By default all method in Hashtable synchronized. Hashtable is thread safe.**