Day 4 :

**Collection Framework**

Collection framework contains collection of classes and interfaces which help to store collection of object or data or item of same or different types it provided lot of pre defined methods which help to add, remove, search, iterate very easily. All collection classes support dynamic memory allocation concept.

Collection Framework util package.

Collection --🡪 interface

Doesn’t extends Collection

Set List Queue Map

**Set** : Set doesn’t allow duplicate date. Set can be order, unorder or sorted. Set doesn’t provide index concept.

**Set classes**

**HashSet : unorder**

**LinkedHashSet : LinkedhashSet internally extends HashSet and it maintain the order.**

**TreeSet : TreeSet internally implements SortedSet interface. This interface provide logic to do sorting. So in TreeSet we need to store same data types values.**

**List** : List allow to store more one value using index concept. List allow duplicate. It maintain the order.

**List classes**

**ArrayList : Normal array is known as fixed in memory. ArrayList is known as dynamic in memory. Normal array allow to store same data types values. By nature ArrayList allow to store same as well as different data types values. Using normal array adding or removing data in between more complex. ArrayList provide the methods.**

**LinkedList**

**Linked list use node concept to store the data.**

**Node contains**

1. **Reference of previous node or next node**
2. **Data : data can be any type like int, float, char, object etc**

**Linked list**

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

**Data nextRef data nextRef data nextref ->null**

**Pref data nref pref data nref pref data nref**

**LinkedList : it consider as double linked list.**

**Vector :** Vector is known as legacy class. by default all methods in Vector are synchronized. It is thread safe class.

**Stack :** Stack is a type of data structure which support LIFO(last in first out).

**Push :** add element

**Pop:** remove element from top

**Peek :** retrieve top most element

**Search :** to search element present in stack

**Size :** display the size.

**Queue** : Queue is a type data structure which support features as First In First Out.

Queue classes

**PriorityQueue**

**LinkedList**

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

**Map classes**

**HashMap : Unorder**

**LinkedHashMap : order**

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

**Hashtable : synchronized (legacy )**