Collection - interface
Collections - class
Collection Framework - is a combination of interfaces and Classes.

List

1. ArrayList
2. LinkedList
3. Vector
4. Stack

Set

1. Hashset
2. LinkedHashset
3. Treeset

Map

1. hashmap
2. LinkedHashmap
3. Hashtable
4. TreeMap

1. List: -

- This category is used to store group of individual elements where the elements can be duplicated.

1. ArrayList
2. LinkedList
3. Vector
4. Stack

```
ArrayList al = new ArrayList();
LinkedList ll = new LinkedList();
Vector v = new Vector();
Stack s = new Stack();

List al = new ArrayList();
List ll = new LinkedList();
List v = new Vector();
list s = new Stack();
```

CRUD

```
Create   - add
Read     - get
Update   - set
Delete   - remove
```

1. List -

- It is the child interface of Collection
- duplicates are allowed
- insertion order is preserved
- Below are implementation classes

1. ArrayList
2. LinkedList
3. Vector
4. Stack

1. ArrayList

- ArrayList is an implementation class of List interface
- data structure is resizable
- Duplicate Objects are allowed
- Insertion order is preserved
- Heterogeneous Objects are allowed

- Null insertion is possible

2. LinkedList -

- LinkedList is an implementation class of List interface
- data structure is double LinkedList
- Duplicate Objects are allowed
- Insertion order is preserved
- Heterogeneous Objects are allowed
- Null insertion is possible

ArrayList - retrieve (get)
LinkedList - Insertion or delete

3. Vector -

- Vector is an implementation class of List interface
- data structure is resizable
- Duplicate Objects are allowed
- Insertion order is preserved
- Heterogeneous Objects are allowed
- Null insertion is possible

- Vector is a synchronized class
- Vector is also called legacy class

4. Stack -

- Stack is a child class of Vector and an implementation class of List interface
- Stack stores a group of objects b using a mechanism called LIFO
- Duplicate Objects are allowed
- Insertion order is preserved
- Heterogeneous Objects are allowed
- Null insertion is possible

push - add
pop - delete
peek - get

Set -

- It is the child interface of Collection
- duplicates are not allowed
- insertion order is not preserved
- Below are implementation classes

1. Hashset
2. LinkedHashset
3. Treeset

1. Hashset -

- HashSet is the implementation class of Set interface
- HashSet internally follows hashtable structure
- duplicate values are not allowed
- Insertion order is not preserved - unordered Collection
- HashSet is not a synchronized class
- HashSet supports only one null value.

Create - add
Read - get
Update - set
Delete - remove

2. LinkedHashSet -

- LinkedHashSet is the implementation class of Set interface
- LinkedHashSet internally follows hashtable + doubly linked list structures
- duplicate values are not allowed

- Insertion order is preserved - ordered Collection
- LinkedHashSet is not a synchronized class
- LinkedHashSet supports only one null value.

3. Treeset -

- Treeset is the implementation class of Set interface
- Treeset internally tree structure
- duplicate values are not allowed
- Insertion order is not preserved - but stored in ascending order
- Treeset is not a synchronized class
- Treeset don't support null value.

Map -

A map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys.

1. hashmap
2. LinkedHashmap
3. Hashtable
4. TreeMap

1. hashmap -

- hashmap is the implementation class of Map interface
- hashmap internally hashtable data structure
- duplicate keys are not allowed and duplicate values are allowed
- Insertion order is not preserved - but stored in ascending order
- hashmap is not a synchronized class
- hashmap support null value (keys - only one - values - multiple).

add
access
remove
update

2. LinkedHashMap

- LinkedHashMap is the implementation class of Map interface
- LinkedHashMap internally hashtable + doubly data structure
- duplicate keys are not allowed and duplicate values are allowed
- Insertion order is preserved
- LinkedHashMap is not a synchronized class
- LinkedHashMap support null value (keys - only one - values - multiple).

add
access
remove
update

3. TreeMap

- TreeMap is the implementation class of Map interface
- TreeMap internally tree structure
- TreeMap keys are not allowed but values are added
- Insertion order is not preserved - but stored in ascending order
- TreeMap is not a synchronized class
- TreeMap don't support null value at keys (multiple null value).

add
access
remove
update

4. Hashtable -

- Hashtable is the implementation class of Map interface
- Hashtable internally tree structure
- Hashtable keys are not allowed but values are added
- Insertion order is not preserved - but stored in descending order
- Hashtable is synchronized class
- Hashtable don't support null value at keys and values

add
access
remove
update