Collection:

1. List(interface)
2. Duplicates allowed
3. Null values allowed
4. Insertion order

* ArrayList

1. Fetching the data arraylist is best choice. 🡺 RandomAccess Interface
2. Adding and deleting the data at the middle of list should not be done using arraylist.
3. By default size is 10.
4. Threshold limit or load factor=75% or 0.75%
5. new capacity = (current capacity\*3/2)+1
6. Aynchronous

* LinkedList

1. Adding and deleting the data from the list, linkedlist is best choice
2. Fetching the data linked list worst choice.

* Vector

1. Vector is same as arraylist only difference is it is synchronous in nature.
2. Default size is 10 and load factor is 75%
3. New capacity = 2 \* current capacity
4. Set(interface)
5. Duplicates are not allowed
6. Null values are not allowed
7. Asynchronous
8. Set internally uses map interface

* HashSet 🡺 uses internally Hashmap

1. Random order data will be print.

* LinkedHashset

1. Insertion order

* TreeSet

1. Sorting order (ascending order)
2. Map(interface)
3. Data will be stored in key value pair.
4. Key will be unique🡺 duplicates not allowed.
5. Keys can have only 1 null values.
6. Values can be duplicated and null values are allowed
7. Asynchronous
8. We cannot iterate map directly, we need to convert that into set to iterate.
9. **You cannot add or delete data in between forloop, if you do you will get concurrent modification exception.**

* HashMap

1. Random order

* LinkedHashMap

1. Insertion order

* TreeMap

1. Sorting or ascending order

ConcurrentHashMap:

1. **You can add or delete data in between forloop, you wont get concurrent modification exception.**
2. **Add and delete operations are synchronous.**
3. **Fetch/ get operations are asynchronous.**