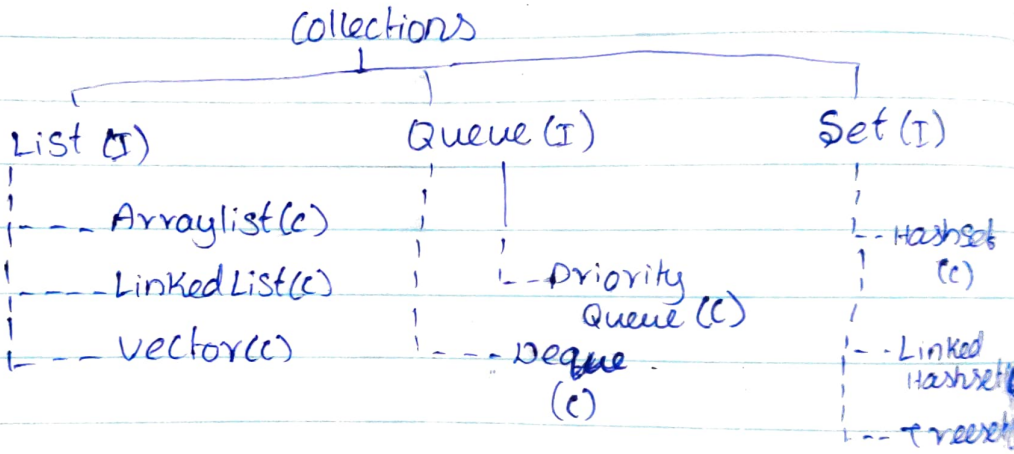
1.

Array	collection
* Array itself a data structure and has some restriction for entering values	collection has varies data structure available to use to manipulation of objects
* Arrays are not growable	collection is growable
* Array elements cannot be removed	Elements can be removed and modified allows null values

2.



3.

Arraylist	vector
* It is a non legacy class, i.e from v1.2	* It's an legacy class class, i.e from v1.0
* It is not synchronized	* It is synchronized
* It is not thread safe	* It is thread safe
* initially the new capacity is meant to be -> (3/2 * old capacity)+1 later changed to 50%	* The capacity is -> 2* old capacity

4.

ArrayList	LinkedList
1.) underlying data structure is Growable (or) Serializable resizable array	Data structure is Doubly LinkedList
2.) Best suited for data retrieval and insertion order	Best suited for insertion and data feasibility
3.) It implements the interfaces of Serializable Cloneable and Random access	 T - Tail, N - node, Head - H It only implements Serializable and Cloneable

5.

Iterator	ListIterator
* It is the universal class that helps to fetch the data (among other things)	It is the interface that extends Iterators
* It can only print forward (unidirectional)	It can print in both forward and backward
* It prints from first to last	It can print for any specified location

6.

List	Set
* The child class of list are • ArrayList • Vector • LinkedList	* The child class of Set are • HashSet • LinkedHashSet • TreeSet
* when insertion or deleting the elements wanted to be feasible	* when we don't want insertion order
* when sorting is last priority	* when we want to do sorting

7.

	Hashset	Treeset
1.)	Hashset don't preserve insertion order, but allows heterogeneous values and null acceptance	It doesn't allow anything, no heterogeneous is allowed, no null is no duplicates and no insertion order
2.)	under lying data Structuire.	*) data Str is Balanced
3.)	There is no guarantee that it will come in ascending order	*) 100% guarantee that output comes in ascending order

8.

Hashset	Hashmap
* If comes under collection	It doesn't belong to collection
* It is Stored as single value	It is Stored as key and value pairs
* add() method is used to add values	put() method is used to add values
* Data can be iterated through iterator directly	cannot be iterated directly

9. HashMap

* It is not thread safe
* It is not-synchronized
* If we want for faster application we can go for HashMap

Hashtable

* It is Thread Safe
* It is Synchronized
* If we want thread safe application by sacrificing the speed of process we can go for Hashtable

10. Comparable

* used to compare single object
* implements compare to method
* comes from Java.lang package

Comparator

It compare two object
implement compare method
comes from Java.util Package

11.) If we do any structural modification to the (List or set while iterating, then JVM throws concurrent modification Exception. this is nothing because of fail-fast iterators.

while we tend to add or remove any element from collection while a thread is iterating over that collection, then it a fail-fast process by JVM when it throws concurrent modification error.