1

Comparable	Comparator
Comparable provides a single sorting sequences. In other words, we can sort the collection on the basisc of a single element such as id, name, price	The Comparator provide multiple sorting sequences In other words we can sort the collection on the basic of multiple elements such as. id, name, and price etc
Comparable affect the orginal class, ie actual class is modified	Comparator doesn't affect the orgiral class, the actual class is not modified.
Comparable provides compareTo() method to sort elements.	Comparator provide compare() method to sort elements
Comparable is present in java.lang package	A comparator is present in java.util package.
We can sort the list elements of Comparable type by collections.sort(List) method	We can sort the elements of Comparator type by collections. sort(List, Comparator) method

ArrayList	Vector
Arraylist not synchronized	Vector is Synchronized
Arraylist increments 50% of current array size if the number of elements exceeds from its capacity	Vector increments 100% means double the array size if the total number of elements exceeds than its capacity.
Arraylist is not a legacy class, it is introduced in JDK 1.2	Vector is legacy class.
Arraylist is fast because it is non Synchronized.	Vector is slow because it is Synchronized ie in a multithreading environment, it holds the other threads in runnable or non_runnable state until current thread release lock of the object
ArrayList uses the Iterator interface to traverse the elements.	A Vector can use the Iterator interface or Enumeration interface to traverse the elements

3

Arraylist	LinkedList
Arraylist internally uses a dynamic array to store the elements	LinkedList internally uses a doubly linked list to store the elements.
Manipulation with Arraylist is slow because it internally uses an array if any element is removed from the array, all the bits are shifted in memory.	Manipulation with LinkedList is faster than Arraylist because it uses a doubly linked list so no bit shifting is required in memory.
An Arraylist class can act as a list only because it implements List only	LinkedList class can act as a list and queue both because it implements List and Deque interface.
Arraylist is better for storing and accessing data	LinkedList is better for manipulating data

4

HashMap	Hashtable
HashMap is non-synchronized. it is not thread safe and can't be shared between many threads without proper synchronized code	Hashtable is synchronized it is thread safe and can be shared with many threads.
HashMap allows one null key and multiple null values	Hashtable doesn't allow any null key or value

HashMap is a new class introduced in JDK 1.2	Hashtable is a legacy class
HashMap is fast	Hashtable is slow.
We can make the Hashmap as Synchronized by calling this code Map m = collections.SynchronizedMap (HashMap);	Hashtable is internally Synchronized and can't be unsynchronized
HashMap is traversed by Iterator	Hashtable is traversed by Enumerator and Iterator
Iterator in HashMap is fail-fast	Enumerator in Hashtable is not fail-fast
HashMap inherits AbstractMap class	Hashtable inherits Dictionary class.

5	ITERATOR	LISTITERATOR
	Iterator can traverse the elements in a collection only in forward direction	ListIterator can traverse the elements in a collection in forward as well as the backward direction.

Iterator is unable to add elements to a collection.	ListIterator can add elements to a collection.
Iterator can not modify the elements in a collection	ListIterator can modify the elements in a collection using set().
Iterator can traverse Map, List and Set	ListIterator can traverse List objects only.
Iterator has no method to obtain an index of the element in a collection.	Using ListIterator, you can obtain an index of the element in a collection.

6

List	Set
The List provides positional acess of the elements in the collection	Set doesn't provide positional acess to the elements in the Collections
Implementation of List are ArrayList, LinkedList, Vector, Stack	Implementation of a Set interface is HashSet and LinkedHashSet
We can store the duplicates elements in the List	We can't store duplicates elements in Set
List maintains insertion order of elements in the collection	Set doesn't maintain any order

The List can store multiple null elements	Set can store only one null elements

HashSet	TreeSet
HashSet is implemented using Hash-Table	The tree set is implemented using a tree structure
HashSet allows a null object	The tree set does not allow the null object. It throws the null pointer exception.
Hash set use equals method to compare two objects	Tree set use Compare method for Comparing two objects
Hash set doesnt now allow a Heterogeneous object.	Tree set allows a heterogeneous object
HashSet does not maintain any order.	TreeSet maintain an object in Sorted order.

KEY INTERFACES OF COLLECTON FRAMEWORK

- Collection
- List
- Set
- SortedSet
- NavigableSet
- Queue
- Map
- SortedMap
- NavigableMap

9

Arrays	Collection
Array are fixed in size that is once we create an array we can not increased or decreased based on our requirement.	Collections are growable in nature that is based on our requirement we can increase or decrease of size
With respect to memory Arrays are not recommended to use.	with respect to performance Collections are not recommended to use
Arrays can hold only homogenous data types elements	Collection can hold both homogenous and heterogeneous elements
There is no underlying data structure for arrays and hence ready made method support is not available	Every collections class is implemented based on some standard data structure and hence for every requirement ready made method support is available being a performance, we can use directly not to implements.

HashMap	HashSet
HashSet is non-Synchronized it is not thread-safe and can't be shared between many threads without proper Synchronized code	HashSet is implemented using Hash-table
HashMap allow one null Key and Multiple null value	HashSet allows a null object.
HashMap is new class introduced in JDK-1.2	HashSet uses equals method to Compare two objects
HashMap is fast	HashSet doesn't allow a Heterogenous objects
HashMap is traversed by iterator.	Hashset does not maintain any order.
HashMap inherits AbstractMap class	