Collection Name	Underlying Data Structure	Duplicates	Index	Null values	Insertion Order	Insert	Read
ArrayList	Resizable array	Allowed	Present	Multiple Null values are Allowed	Preserved	add(value)	get(index)
Vector	Resizable array	Allowed	Present	Multiple Null values are Allowed	Preserved	add(value)	get(index)
LinkedList	Doubly LinkedList	Allowed	Present	Multiple Null values are Allowed	Preserved	add(value)	get(index)
HashSet	HashTable	**NOT ALLOWED**	**Absent**	Only ONE Null value is Allowed	Not Preserved	add(value)	**No specific method available. Data can be read using FOR-EACH loop or ITERATOR**
LinkedHashSet	HashTable , LinkedList	**NOT ALLOWED**	**Absent**	Only ONE Null value is Allowed	Not Preserved	add(value)	**No specific method available. Data can be read using FOR-EACH loop or ITERATOR**
TreeSet	Binary Search Tree	**NOT ALLOWED**	**Absent**	**NO NULL VALUES ARE ALLOWED (NOT EVEN ONE)**	Not Preserved	add(value)	**No specific method available. Data can be read using FOR-EACH loop or ITERATOR**
PriorityQueue	Binary Search Tree	**ALLOWED**	**Absent**	**NO NULL VALUES ARE ALLOWED (NOT EVEN ONE)**	Not Preserved	add(value)	**poll() - For sorted result. Data can also be read using FOR-EACH loop or ITERATOR without sorted order**
HashMap	HashTable	Value can be duplicate Key should be always Unique	**Absent**	**ONLY ONE KEY CAN BE NULL** MULTIPLE NULL VALUES ARE ALLOWED	Not Preserved	put(key, value)	get(**key**)

:ramework

Update	Delete	Search	Count Of Elements	Thread Safety	Sorted	Fast - Opts	Slow - Opts
add(index, value)	remove(index)	contains(value)	size()	NOT THREAD SAFE	NO	Read- get(index)	Insert- add(value) as the size of ArrayList increases
add(index, value)	remove(index)	contains(value)	size()	**THREAD SAFE**	NO	Read- get(index)	Insert- add(value) as the size of Vector increases
add(index, value)	remove(index)	contains(value)	size()	NOT THREAD SAFE	NO	Insert- add(value) as the size of Vector increases	Read- get(index) because we will not know address of a particular node
No specific method available. Data can be updated by contains(),remove() and add() used in combination of a LOGIC	remove(value)	contains(value)	size()	NOT THREAD SAFE	NO	Search - contains(value) because the data will be searched by finding the index using hashing technique.	Insert- add(value) because adding the data Involves **HASHING** process
No specific method available. Data can be updated by contains(),remove() and add() used in combination of a LOGIC	remove(value)	contains(value)	size()	NOT THREAD SAFE	NO	Search - contains(value) because the data will be searched by finding the index using hashing technique.	Insert- add(value) because adding the data Involves **HASHING** process
No specific method available. Data can be updated by contains(),remove() and add() used in combination of a LOGIC	remove(value)	contains(value)	size()	NOT THREAD SAFE	**YES**	Search - contains(value) because data is sorted.	Insert- add(value) because adding the data Involves **Comparison With Comparable or Comparator** process
No specific method available. Data can be updated by contains(),remove() and add() used in combination of a LOGIC	remove(value)	contains(value)	size()	NOT THREAD SAFE	**YES**	Search - contains(value) because data is sorted.	Insert- add(value) because adding the data Involves **Comparison With Comparable or Comparator** process
put(key, **NewValue**)	remove(key)	contains(value)	size()	NOT THREAD SAFE	**NO**	Search - contains(value) because the data will be searched by finding the index using hashing	Insert- add(value) because adding the data Involves **HASHING** process