Methods in Collections:  
ArrayList:

| **Method** | **Description** |
| --- | --- |
| [add(int index, Object element)](https://www.geeksforgeeks.org/java/java-util-arraylist-add-method-java/) | This method is used to insert a specific element at a specific position index in a list. |
| [add(Object o)](https://www.geeksforgeeks.org/java/java-util-arraylist-add-method-java/) | This method is used to append a specific element to the end of a list. |
| [addAll(Collection C)](https://www.geeksforgeeks.org/java/java-util-arraylist-addall-method-java/) | This method is used to append all the elements from a specific collection to the end of the mentioned list, in such an order that the values are returned by the specified collection’s iterator. |
| [addAll(int index, Collection C)](https://www.geeksforgeeks.org/java/java-util-arraylist-addall-method-java/) | Used to insert all of the elements starting at the specified position from a specific collection into the mentioned list. |
| [clear()](https://www.geeksforgeeks.org/java/arraylist-clear-method-in-java-with-examples/) | This method is used to remove all the elements from any list. |
| [clone()](https://www.geeksforgeeks.org/java/clone-method-in-java-2/) | This method is used to return a shallow copy of an ArrayList in Java. |
| [contains(Object o)](https://www.geeksforgeeks.org/java/arraylist-contains-java/) | Returns true if this list contains the specified element. |
| [ensureCapacity(int minCapacity)](https://www.geeksforgeeks.org/java/arraylist-ensurecapacity-method-in-java-with-examples/) | Increases the capacity of this ArrayList instance, if necessary, to ensure that it can hold at least the number of elements specified by the minimum capacity argument. |
| [forEach(Consumer<? super E> action)](https://www.geeksforgeeks.org/java/arraylist-foreach-method-in-java/) | Performs the given action for each element of the Iterable until all elements have been processed or the action throws an exception. |
| [get(int index)](https://www.geeksforgeeks.org/java/arraylist-get-method-java-examples/) | Returns the element at the specified position in this list. |
| [indexOf(Object O)](https://www.geeksforgeeks.org/java/java-util-arraylist-indexof-java/) | The index the first occurrence of a specific element is either returned or -1 in case the element is not in the list. |
| [isEmpty()](https://www.geeksforgeeks.org/java/arraylist-isempty-java-example/) | Returns true if this list contains no elements. |
| [lastIndexOf(Object O)](https://www.geeksforgeeks.org/java/arraylist-lastindexof-java-example/) | The index of the last occurrence of a specific element is either returned or -1 in case the element is not in the list. |
| [listIterator()](https://www.geeksforgeeks.org/java/arraylist-listiterator-method-in-java-with-examples/) | Returns a list iterator over the elements in this list (in proper sequence). |
| [listIterator(int index)](https://www.geeksforgeeks.org/java/arraylist-listiterator-method-in-java-with-examples/) | Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list. |
| [remove(int index)](https://www.geeksforgeeks.org/java/arraylist-linkedlist-remove-methods-java-examples/) | Removes the element at the specified position in this list. |
| [remove(Object o)](https://www.geeksforgeeks.org/java/arraylist-linkedlist-remove-methods-java-examples/) | Removes the first occurrence of the specified element from this list, if it is present. |
| [removeAll(Collection c)](https://www.geeksforgeeks.org/java/arraylist-removeall-method-in-java-with-examples/) | Removes from this list all of its elements that are contained in the specified collection. |
| [removeIf(Predicate filter)](https://www.geeksforgeeks.org/java/arraylist-removeif-method-in-java/) | Removes all of the elements of this collection that satisfy the given predicate. |
| [removeRange(int fromIndex, int toIndex)](https://www.geeksforgeeks.org/java/arraylist-removerange-java-examples/) | Removes from this list all of the elements whose index is between from Index, inclusive and to Index, exclusive. |
| [retainAll(Collection<?> c)](https://www.geeksforgeeks.org/java/arraylist-retainall-method-in-java/) | Retains only the elements in this list that are contained in the specified collection. |
| [set(int index, E element)](https://www.geeksforgeeks.org/java/arraylist-set-method-in-java-with-examples/) | Replaces the element at the specified position in this list with the specified element. |
| [size()](https://www.geeksforgeeks.org/java/arraylist-size-method-in-java-with-examples/) | Returns the number of elements in this list. |
| [spliterator()](https://www.geeksforgeeks.org/java/arraylist-spliterator-method-in-java/) | Creates a late-binding and fail-fast Spliterator over the elements in this list. |
| [subList(int fromIndex, int toIndex)](https://www.geeksforgeeks.org/java/arraylist-sublist-method-in-java-with-examples/) | Returns a view of the portion of this list between the specified fromIndex, inclusive and toIndex, exclusive. |
| [toArray()](https://www.geeksforgeeks.org/java/arraylist-array-conversion-java-toarray-methods/) | This method is used to return an array containing all of the elements in the list in the correct order. |
| [toArray(Object[] O)](https://www.geeksforgeeks.org/java/arraylist-array-conversion-java-toarray-methods/) | It is also used to return an array containing all of the elements in this list in the correct order same as the previous method. |
| [trimToSize()](https://www.geeksforgeeks.org/java/arraylist-trimtosize-java-example/) | This method is used to trim the capacity of the instance of the ArrayList to the list's current size. |

**ArrayList vs LinkedList**

| **ArrayList** | **LinkedList** |
| --- | --- |
| Underlying structure is Dynamic Array | Underlying structure is Doubly linked list |
| O(1) - Fast random access | O(n) - Slow random access |
| Memory is lower (contiguous memory) | Memory is higher (extra pointers per node) |
| Iteration speed is faster | Iteration speed is slower. |
| Insertion and deletion is slower | Insertion and deletion are faster. |

LinkedList:

| **Method** | **Description** |
| --- | --- |
| [add(int index, E element)](https://www.geeksforgeeks.org/java/java-util-linkedlist-add-method-in-java/) | Inserts element at given index. |
| [add(E e)](https://www.geeksforgeeks.org/java/java-util-linkedlist-add-method-in-java/) | This method Appends the specified element to the end of this list. |
| [addAll(int index, Collection<E> c)](https://www.geeksforgeeks.org/java/java-util-linkedlist-addall-method-in-java/) | Inserts all elements of collection starting at index. |
| [addAll(Collection<E> c)](https://www.geeksforgeeks.org/java/java-util-linkedlist-addall-method-in-java/) | Appends all elements of collection to end |
| [addFirst(E e)](https://www.geeksforgeeks.org/java/linkedlist-addfirst-method-in-java/) | This method Inserts the specified element at the beginning of this list. |
| [addLast(E e)](https://www.geeksforgeeks.org/java/linkedlist-addlast-method-in-java/) | This method Appends the specified element to the end of this list. |
| [clear()](https://www.geeksforgeeks.org/java/linkedlist-clear-method-in-java/) | This method removes all of the elements from this list. |
| [clone()](https://www.geeksforgeeks.org/java/linkedlist-clone-method-in-java/) | This method returns a shallow copy of this LinkedList. |
| [contains(Object o)](https://www.geeksforgeeks.org/java/linkedlist-contains-method-in-java/) | This method returns true if this list contains the specified element. |
| [descendingIterator()](https://www.geeksforgeeks.org/java/linkedlist-descendingiterator-method-in-java-with-examples/) | This method returns an iterator over the elements in this deque in reverse sequential order. |
| [element()](https://www.geeksforgeeks.org/java/linkedlist-element-method-in-java-with-examples/) | This method retrieves but does not remove, the head (first element) of this list. |
| [get(int index)](https://www.geeksforgeeks.org/java/linkedlist-get-method-in-java/) | This method returns the element at the specified position in this list. |
| [getFirst()](https://www.geeksforgeeks.org/java/java-util-linkedlist-get-getfirst-getlast-java/) | This method returns the first element in this list. |
| [getLast()](https://www.geeksforgeeks.org/java/linkedlist-getlast-method-in-java/) | This method returns the last element in this list. |
| [indexOf(Object o)](https://www.geeksforgeeks.org/java/linkedlist-indexof-method-in-java/) | Return first index of element or -1 if element is not present |
| [lastIndexOf(Object o)](https://www.geeksforgeeks.org/java/linkedlist-lastindexof-method-in-java-with-examples/) | Return Last index of element or -1 if element is not present |
| [listIterator(int index)](https://www.geeksforgeeks.org/java/linkedlist-listiterator-method-in-java/) | This method returns a list-iterator of the elements. |
| [offer(E e)](https://www.geeksforgeeks.org/java/java-util-linkedlist-offer-offerfirst-offerlast-java/) | This method Adds the specified element as the tail (last element) of this list. |
| [offerFirst(E e)](https://www.geeksforgeeks.org/java/java-util-linkedlist-offer-offerfirst-offerlast-java/) | This method Inserts the specified element at the front of this list. |
| [offerLast(E e)](https://www.geeksforgeeks.org/java/java-util-linkedlist-offer-offerfirst-offerlast-java/) | This method Inserts the specified element at the end of this list. |
| [peek()](https://www.geeksforgeeks.org/java/java-util-linkedlist-peek-peekfirst-peeklast-java/) | This method retrieves but does not remove, the head (first element) of this list. |
| [peekFirst()](https://www.geeksforgeeks.org/java/java-util-linkedlist-peek-peekfirst-peeklast-java/) | This method retrieves, but does not remove, the first element of this list or returns null if this list is empty. |
| [peekLast()](https://www.geeksforgeeks.org/java/java-util-linkedlist-peek-peekfirst-peeklast-java/) | This method retrieves, but does not remove, the last element of this list or returns null if this list is empty. |
| [poll()](https://www.geeksforgeeks.org/java/java-util-linkedlist-poll-pollfirst-polllast-examples-java/) | This method retrieves and removes the head (first element) of this list. |
| [pollFirst()](https://www.geeksforgeeks.org/java/java-util-linkedlist-poll-pollfirst-polllast-examples-java/) | This method retrieves and removes the first element of this list or returns null if this list is empty. |
| [pollLast()](https://www.geeksforgeeks.org/java/java-util-linkedlist-poll-pollfirst-polllast-examples-java/) | This method retrieves and removes the last element of this list or returns null if this list is empty. |
| [pop()](https://www.geeksforgeeks.org/java/linkedlist-pop-method-in-java/) | This method Pops an element from the stack represented by this list. |
| [push(E e)](https://www.geeksforgeeks.org/java/linkedlist-push-method-in-java/) | This method pushes an element onto the stack represented by this list. |
| [remove()](https://www.geeksforgeeks.org/java/linkedlist-remove-method-in-java/) | This method retrieves and removes the head (first element) of this list. |
| [remove(int index)](https://www.geeksforgeeks.org/java/linkedlist-remove-method-in-java/) | This method removes the element at the specified position in this list. |
| [remove(Object o)](https://www.geeksforgeeks.org/java/linkedlist-remove-method-in-java/) | This method removes the first occurrence of the specified element from this list if it is present. |
| [removeFirst()](https://www.geeksforgeeks.org/java/linkedlist-removefirst-method-in-java/) | This method removes and returns the first element from this list. |
| [removeFirstOccurrence(Object o)](https://www.geeksforgeeks.org/java/linkedlist-removefirstoccurrence-method-in-java/) | This method removes the first occurrence of the specified element in this list. |
| [removeLast()](https://www.geeksforgeeks.org/java/linkedlist-removelast-method-in-java/) | This method removes and returns the last element from this list. |
| [removeLastOccurrence(Object o)](https://www.geeksforgeeks.org/java/linkedlist-removelastoccurrence-method-in-java-with-example/) | This method removes the last occurrence of the specified element in this list. |
| [set(int index, E element)](https://www.geeksforgeeks.org/java/linkedlist-set-method-in-java/) | This method replaces the element at the specified position in this list with the specified element. |
| [size()](https://www.geeksforgeeks.org/java/linkedlist-size-method-in-java/) | This method returns the number of elements in this list. |
| [spliterator()](https://www.geeksforgeeks.org/java/linkedlist-spliterator-method-in-java/) | This method creates a late-binding and fail-fast Spliterator over the elements in this list. |
| [toArray()](https://www.geeksforgeeks.org/java/linkedlist-toarray-method-in-java-with-example/) | This method returns an array containing all of the elements in this list in proper sequence. |
| [toArray(T[] a)](https://www.geeksforgeeks.org/java/linkedlist-toarray-method-in-java-with-example/) | Returns array of all elements in given type. |
| [toString()](https://www.geeksforgeeks.org/java/object-tostring-method-in-java/) | This method returns string representation of list. |

HashSet:

| **Method** | **Description** |
| --- | --- |
| add(E e) | Adds an element if not already present |
| remove(Object o) | Removes the specified element |
| contains(Object o) | Checks if an element is present |
| size() | Returns number of elements |
| clear() | Removes all elements |
| isEmpty() | Checks if set is empty |
| iterator() | Returns an iterator over elements |
| clone() | Returns a shallow copy |
| toArray() | Converts to an array |
| addAll(Collection c) | Adds all elements from another collection |
| removeAll(Collection c) | Removes all elements in another collection |
| retainAll(Collection c) | Keeps only elements that are also in another collection |

HashMap:

**Methods of HashMap**

* **K** – The type of the keys in the map.
* **V** – The type of values mapped in the map.

| **Method** | **Description** |
| --- | --- |
| [clear()](https://www.geeksforgeeks.org/java/hashmap-clear-method-in-java/) | Removes all of the mappings from this map. |
| [clone()](https://www.geeksforgeeks.org/java/hashmap-clone-method-in-java/) | Returns a shallow copy of this HashMap instance. |
| [compute(K key, BiFunction<? super K,? super V,? extends V> remappingFunction)](https://www.geeksforgeeks.org/java/hashmap-compute-method-in-java-with-examples/) | Attempts to compute a mapping for the specified key and its current mapped value |
| [computeIfAbsent(K key, Function<?super K,? extends V> mappingFunction)](https://www.geeksforgeeks.org/java/hashmap-computeifabsent-method-in-java-with-examples/) | Adds computed value if key absent/null. |
| [computeIfPresent(K key, BiFunction<? super K,? super V,? extends V> remappingFunction)](https://www.geeksforgeeks.org/java/hashmap-computeifpresentkey-bifunction-method-in-java-with-examples/) | If the value for the specified key is present and non-null, attempts to compute a new mapping given the key and its current mapped value. |
| [containsKey(Object key)](https://www.geeksforgeeks.org/java/hashmap-containskey-method-in-java/) | Returns true if this map contains a mapping for the specified key. |
| [containsValue(Object value)](https://www.geeksforgeeks.org/java/hashmap-containsvalue-method-in-java/) | Returns true if this map maps one or more keys to the specified value. |
| [entrySet()](https://www.geeksforgeeks.org/java/hashmap-entryset-method-in-java/) | Returns a Set view of the mappings contained in this map. |
| [get(Object key)](https://www.geeksforgeeks.org/java/hashmap-get-method-in-java/) | Returns the value to which the specified key is mapped, or null if this map contains no mapping for the key. |
| [isEmpty()](https://www.geeksforgeeks.org/java/hashmap-isempty-method-in-java/) | Returns true if this map contains no key-value mappings. |
| [keySet()](https://www.geeksforgeeks.org/java/hashmap-keyset-method-in-java/) | Returns a Set view of the keys contained in this map. |
| [merge(K key, V value, BiFunction<? super V,? super V,? extends V> remappingFunction)](https://www.geeksforgeeks.org/java/hashmap-mergekey-value-bifunction-method-in-java-with-examples/) | If the specified key is not already associated with a value or is associated with null, associate it with the given non-null value. |
| [put(K key, V value)](https://www.geeksforgeeks.org/java/hashmap-put-method-in-java/) | Associates the specified value with the specified key in this map. |
| [putAll(Map<? extends K,? extends V> m)](https://www.geeksforgeeks.org/java/hashmap-putall-method-in-java/) | Copies all of the mappings from the specified map to this map. |
| [remove(Object key)](https://www.geeksforgeeks.org/java/hashmap-remove-method-in-java/) | Removes the mapping for the specified key from this map if present. |
| [size()](https://www.geeksforgeeks.org/java/hashmap-size-method-in-java/) | Returns the number of key-value mappings in this map. |
| [values()](https://www.geeksforgeeks.org/java/hashmap-values-method-in-java/) | Returns a Collection view of the values contained in this map. |

**Methods inherited from class java.util.AbstractMap**

| **Method** | **Description** |
| --- | --- |
| [equals()](https://www.geeksforgeeks.org/java/abstractmap-equals-method-in-java-with-examples/) | Compares the specified object with this map for equality. |
| [hashCode()](https://www.geeksforgeeks.org/java/abstractmap-hashcode-method-in-java-with-examples/) | Returns the hash code value for this map. |
| [toString()](https://www.geeksforgeeks.org/java/abstractmap-simpleentry-tostring-method-in-java-with-examples/) | Returns a string representation of this map. |

**Methods inherited from interface java.util.Map**

| **Method** | **Description** |
| --- | --- |
| [equals()](https://www.geeksforgeeks.org/java/map-equals-method-in-java-with-examples/) | Compares the specified object with this map for equality. |
| [forEach(BiConsumer<? super K, ? super V> action)](https://www.geeksforgeeks.org/java/hashmap-foreachbiconsumer-method-in-java-with-examples/) | Performs the given action for each entry in this map until all entries have been processed or the action throws an exception. |
| [getOrDefault(Object key, V defaultValue)](https://www.geeksforgeeks.org/java/hashmap-getordefaultkey-defaultvalue-method-in-java-with-examples/) | Returns the value to which the specified key is mapped, or defaultValue if this map contains no mapping for the key. |
| [hashCode()](https://www.geeksforgeeks.org/java/map-hashcode-method-in-java-with-examples/) | Returns the hash code value for this map. |
| [putIfAbsent(K key, V value)](https://www.geeksforgeeks.org/java/hashmap-putifabsentkey-value-method-in-java-with-examples/) | If the specified key is not already associated with a value (or is mapped to null) associates it with the given value and returns null, else returns the current value. |
| [remove(Object key, Object value)](https://www.geeksforgeeks.org/java/hashmap-remove-method-in-java/) | Removes the entry for the specified key only if it is currently mapped to the specified value. |
| [replace(K key, V value)](https://www.geeksforgeeks.org/java/hashmap-replacekey-value-method-in-java-with-examples/) | Replaces the entry for the specified key only if it is currently mapped to some value. |
| [replace(K key, V oldValue, V newValue)](https://www.geeksforgeeks.org/java/hashmap-replacekey-oldvalue-newvalue-method-in-java-with-examples/) | Replaces the entry for the specified key only if currently mapped to the specified value. |
| [replaceAll(BiFunction<? super K,? super V,? extends V> function)](https://www.geeksforgeeks.org/java/hashmap-replaceallbifunction-method-in-java-with-examples/) | Replaces each entry's value with the result of invoking the given function on that entry until all entries have been processed or the function throws an exception. |

HashSet vs HashMap:

| **Basis** | **HashSet** | **HashMap** |
| --- | --- | --- |
| **Implementation** | HashSet implements a Set interface. | HashMap implements a storesMap interface. |
| **Duplicates** | HashSet doesn't allow duplicate values. | HashMap stores key-value pairs and doesn’t allow duplicate keys. A duplicate key replaces the old value. |
| **Number of objects during storing objects** | HashSet requires only one object add(Object o). | HashMap requires two objects put(K key, V Value) to add an element to the HashMap object. |
| **Dummy value** | HashSet internally uses a HashMap, where each element added is stored as a key with a dummy value. | HashMap does not have any concept of dummy value. |
| **Storing or Adding a mechanism** | HashSet internally uses the HashMap object to store or add the objects. | HashMap internally uses hashing to store or add objects |
| **Faster** | HashSet is slower than HashMap. | HashMap is faster than HashSet. |
| **Insertion** | HashSet uses the add() method for adding or storing data. | HashMap uses the put() method for storing data. |

Queue:

| **Method** | **Description** |
| --- | --- |
| add(E e) | Inserts an element; throws exception if capacity exceeded |
| offer(E e) | Inserts an element; returns false if capacity exceeded |
| remove() | Retrieves and removes the head; throws exception if empty |
| poll() | Retrieves and removes the head; returns null if empty |
| element() | Retrieves head without removing; throws exception if empty |
| peek() | Retrieves head without removing; returns null if empty |

Comparison between ArrayList, LinkedList, HashSet, HashMap and Queue:

| **Feature / Data Structure** | **ArrayList** | **LinkedList** | **HashSet** | **HashMap** | **Queue** |
| --- | --- | --- | --- | --- | --- |
| **Type** | List (ordered) | List & Deque | Set (unordered) | Map (key-value pairs) | Interface |
| **Underlying Implementation** | Resizable array | Doubly linked list | Hash table | Hash table | Depends on implementation (LinkedList, PriorityQueue, ArrayDeque, etc.) |
| **Order Maintained** | Yes (insertion order) | Yes (insertion order) | No | No | Depends (LinkedList – FIFO, PriorityQueue – priority order) |
| **Allows Duplicates** | Yes | Yes | No | Keys: No, Values: Yes | Depends on implementation |
| **Allows Null** | Yes | Yes | Yes (one null element) | One null key, multiple null values allowed | Depends on implementation |
| **Access by Index** | Fast (O(1)) | Slow (O(n)) | Not applicable | Not applicable | Not applicable |
| **Insertion/Deletion** | Slow in middle (O(n)) | Fast at ends (O(1)) | Fast (O(1)) | Fast (O(1)) | Fast at ends depending on implementation |
| **Searching** | O(n) | O(n) | O(1) | O(1) | Depends (usually O(1) or O(log n)) |
| **Thread Safety** | Not synchronized | Not synchronized | Not synchronized | Not synchronized | Depends (LinkedList not synchronized, ConcurrentLinkedQueue is) |
| **Use Cases** | Storing list of items | Frequent insertion/deletion | Unique collection | Mapping keys to values | Task scheduling, buffers |
| **Memory Overhead** | Less overhead | More overhead (extra pointers) | Depends on hash capacity | Depends on hash capacity | Depends on implementation |
| **When to Use** | Random access, frequent reads | Frequent insertions/deletions | Unique items | Fast key-based lookup | Ordered processing tasks |
| **Examples** | List of users | Browser history | Unique tags | Configuration settings | Print jobs, BFS traversal |