-explore the methods present in Collection interface, List interface, Set interface, Queue interface

-explore collection framework library besides the concepts covered in sessions

-methods in ListIterator

-programs for using all the methods present in ListIterator

-Create your own ArrayList implementation class (new)

-Enter different data types in the array list and display only integers using iterator ; after that only even integers

-explore all methods present in String class

-Write programs for all methods present in String class and push it to git

-As compared to array, why array has less performance

**Methods in collection interface**

**1. boolean add(E e)**

Ensures that this collection contains the specified element (optional operation).

**2. boolean addAll(**[**Collection**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html)**<? extends**[**E**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html)**> c)**

Adds all of the elements in the specified collection to this collection (optional operation).

**3. void clear()**

Removes all of the elements from this collection (optional operation).

**4. boolean** [**contains**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#contains(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html)**o)**

  Returns true if this collection contains the specified element.

**5. boolean** [**containsAll**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#containsAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html)**<?> c)**

Returns true if this collection contains all of the elements in the specified collection.

**6. boolean** [**equals**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#equals(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html)**o)**

Compares the specified object with this collection for equality.

**7. int hashCode()**

Returns the hash code value for this collection.

**8. boolean isEmpty()**

Returns true if this collection contains no elements.

**9.** [**Iterator**](https://docs.oracle.com/javase/6/docs/api/java/util/Iterator.html)**<**[**E**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html)**>** [**iterator**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#iterator())**()**

Returns an iterator over the elements in this collection.

**10. boolean** [**remove**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#remove(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html)**o)**

Removes a single instance of the specified element from this collection, if it is present (optional operation).

**11. boolean** [**removeAll**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#removeAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html)**<?> c)**

Removes all of this collection's elements that are also contained in the specified collection (optional operation).

**12. boolean** [**retainAll**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#retainAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html)**<?> c)**

Retains only the elements in this collection that are contained in the specified collection (optional operation).

**13. int** [**size**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#size())**()**

Returns the number of elements in this collection.

**14.** [**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html)**[]**[**toArray**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#toArray())**()**

Returns an array containing all of the elements in this collection.

**15. <T> T[]** [**toArray**](https://docs.oracle.com/javase/6/docs/api/java/util/Collection.html#toArray(T[]))**(T[] a)**

Returns an array containing all of the elements in this collection; the runtime type of the returned array is that of the specified array.

**Methods in list interface**

**1. boolean add(E e)**

Appends the specified element to the end of this list (optional operation).

**2. void add(int index, E element)**

It is used to insert the specified element at the specified position in a list.

**3. boolean addAll(Collection<? extends E> c)**

Appends all of the elements in the specified collection to the end of this list, in the order that they are returned by the specified collection's iterator (optional operation).

**4. boolean addAll(int index, Collection<? extends E> c)**

Inserts all of the elements in the specified collection into this list at the specified position (optional operation).

**5. void clear()**

It is used to remove all of the elements from this list.

**6. boolean** [**contains**](https://docs.oracle.com/javase/6/docs/api/java/util/List.html#contains(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/6/docs/api/java/lang/Object.html)**o)**

Returns true if this list contains the specified element.

**7. boolean** [**containsAll**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#containsAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<?> c)**

Returns true if this list contains all of the elements of the specified collection.

**8. boolean equals(Object o)**

It is used to compare the specified object with the elements of a list.

**9. E get(int index)**

Returns the element at the specified position in this list.

**10. int hashcode()**

It is used to return the hash code value for a list.

**11. int indexOf(Object o)**

Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.

**12. boolean isEmpty()**

It is used to fetch the element from the particular position of the list.

**13.** [**Iterator**](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html)**<**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**> iterator()**

Returns an iterator over the elements in this list in proper sequence.

**14. int lastIndexOf(Object o)**

It is used to return the index in this list of the last occurrence of the specified element, or -1 if the list does not contain this element.

**15.** [**ListIterator**](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html)**<**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**>** [**listIterator**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#listIterator())**()**

Returns a list iterator over the elements in this list (in proper sequence).

**16.** [**ListIterator**](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html)**<**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**>**  [**listIterator**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#listIterator(int))**(int index)**

Returns a list iterator over the elements in this list (in proper sequence), starting at the specified position in the list.

**17. E**  [**remove**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#remove(int))**(int index)**

Removes the element at the specified position in this list (optional operation).

**19.boolean** [**remove**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#remove(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)**o)**

Removes the first occurrence of the specified element from this list, if it is present (optional operation).

**20. boolean**  [**removeAll**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#removeAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<?> c)**

Removes from this list all of its elements that are contained in the specified collection (optional operation).

**21. boolean** [**retainAll**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#retainAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<?> c)**

Retains only the elements in this list that are contained in the specified collection (optional operation).

**22. E**  [**set**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#set(int,%20E))**(int index,**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**element)**

Replaces the element at the specified position in this list with the specified element (optional operation).

**23. int size()**

Returns the number of elements in the list

**24.** [**List**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**<**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html)**>** [**subList**](https://docs.oracle.com/javase/7/docs/api/java/util/List.html#subList(int,%20int))**(int fromIndex, int toIndex)**

Returns a view of the portion of this list between the specified fromIndex, inclusive, and toIndex, exclusive.

**25. Object[] toArray()**

It is used to return an array containing all of the elements in this list in the correct order.

**26. <T> T[] toArray(T[] a)**

Returns an array containing all of the elements in this list in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.

**Methods in Set Interface**

**1. boolean add(E e)**

Adds the specified element to this set if it is not already present (optional operation).

**2. boolean** [**addAll**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#addAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<? extends**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html)**> c)**

Adds all of the elements in the specified collection to this set if they're not already present (optional operation).

**3. void clear()**

Removes all of the elements from this set (optional operation).

**4. boolean** [**contains**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#contains(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)**o)**

Returns true if this set contains the specified element.

**5. boolean** [**containsAll**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#containsAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<?> c)**

Returns true if this set contains all of the elements of the specified collection.

**6. boolean** [**equals**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#equals(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)**o)**

Compares the specified object with this set for equality.

**7. int hashCode()**

Returns the hash code value for this set.

**8. boolean isEmpty()**

Returns true if this set contains no elements.

**9.** [**Iterator**](https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html)**<**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html)**>** [**iterator**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#iterator())**()**

Returns an iterator over the elements in this set.

**10. boolean** [**remove**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#remove(java.lang.Object))**(**[**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)**o)**

Removes the specified element from this set if it is present (optional operation).

**11. boolean** [**removeAll**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#removeAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<?> c)**

Removes from this set all of its elements that are contained in the specified collection (optional operation).

**12. boolean** [**retainAll**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#retainAll(java.util.Collection))**(**[**Collection**](https://docs.oracle.com/javase/7/docs/api/java/util/Collection.html)**<?> c)**

Retains only the elements in this set that are contained in the specified collection (optional operation).

**13. int size()**

Returns the number of elements in this set (its cardinality).

**14.** [**Object**](https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html)**[]** [**toArray**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#toArray())**()**

Returns an array containing all of the elements in this set.

**15. <T> T[]**[**toArray**](https://docs.oracle.com/javase/7/docs/api/java/util/Set.html#toArray(T[]))**(T[] a)**

Returns an array containing all of the elements in this set; the runtime type of the returned array is that of the specified array.

**Methods in Queue Interface**

**1. boolean add(E e)**

Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions, returning true upon success and throwing an IllegalStateException if no space is currently available.

**2. E element()**

Retrieves, but does not remove, the head of this queue.

**3. boolean** [**offer**](https://docs.oracle.com/javase/7/docs/api/java/util/Queue.html#offer(E))**(**[**E**](https://docs.oracle.com/javase/7/docs/api/java/util/Queue.html)**e)**

Inserts the specified element into this queue if it is possible to do so immediately without violating capacity restrictions.

**4. E peek()**

Retrieves, but does not remove, the head of this queue, or returns null if this queue is empty.

**5. E poll()**

Retrieves and removes the head of this queue, or returns null if this queue is empty.

**6. E remove()**

Retrieves and removes the head of this queue.

**Methods in Iterator interface**

**1. boolean hasNext()**

Returns true if the iteration has more elements.

**2. E next()**

Returns the next element in the iteration.

**3. void remove()**

Removes from the underlying collection the last element returned by this iterator (optional operation).

**Methods in ListIterator interface**

**1. void add(E e)**

Inserts the specified element into the list (optional operation).

**2. boolean hasNext()**

Returns true if the iteration has more elements.

**3. boolean hasPrevious()**

Returns true if this list iterator has more elements when traversing the list in the reverse direction.

**4. E next()**

Returns the next element in the list and advances the cursor position.

**5. int** [**nextIndex**](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#nextIndex())**()**

Returns the index of the element that would be returned by a subsequent call to [next()](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#next()).

**6. E previous()**

Returns the previous element in the list and moves the cursor position backwards.

**7. int previousIndex()**

Returns the index of the element that would be returned by a subsequent call to [previous()](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#previous()).

**8. void remove()**

Removes from the list the last element that was returned by [next()](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#next()) or [previous()](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#previous()) (optional operation).

**9. void set(E e)**

Replaces the last element returned by [next()](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#next()) or [previous()](https://docs.oracle.com/javase/7/docs/api/java/util/ListIterator.html#previous()) with the specified element (optional operation).

.