1. What is ArrayList?

ArrayList<E> class is Resizable-array implementation of List<E> interface. It allows duplication of elements as well as allows one and/or many null. ArrayList<E> class is not synchronized so should not be used in multi-threaded environment. ArrayList<E> uses Object[] array underneath to store the elements. It provides us the advantage of lookup of element by index in O(1) which is fast. Read about it more [here](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-introduction-and-how-it.html).

1. Which Interface ArrayList<E> implements?

ArrayList<E> class implements 4 different interfaces. List<E>, RandomAccess, Cloneable and Serializable.

List<E> interface defines a contract for List related operations. List<E> interface extends Collection<E> interface so all List(s) are Collection. And Collection<E> interface extends Iterable<E> interface so all Collection(s) are Itearble.

RandomAccess, Cloneable and Serializable interfaces are Marker interfaces. Marker Interfaces are also known as tagging design pattern.

RandomAccess Interface signifies that ArrayList<E> class supports the Random access i.e positional or index access.

Cloneable Interface signifies that ArrayList<E> class has a clone method which will return copy of this ArrayList<E> instance.

Serializable Interface signifies that ArrayList<E> is serializable.

1. Which class ArrayList<E> extend?

ArrayList<E> class extends AbstractList<E> class. AbstractList<E> class provides a skeleton implementation of List interface. It is can be extended by class to minimize the overhead or effort required to create a List backed by Object[] array.

1. What is initial capacity of ArrayList?

The initial capacity of ArrayList is 10. Once it is filled it resizes itself by factor roughly of 1.5 times and copy old elements to new array.

1. How does ArrayList increment the size internally?

ArrayList<E> class increments its size roughly by 1.5 times once the old capacity is filled. When we call add(E e) method to insert element in ArrayList<E> it handles resizing of Object[] array itself.

Below are the calls made to resize the Object[] array.

* add(E e) method calls ensureCapacityInternal(size + 1). ensureCapacityInternal() method is private method in class ArrayList<E>.
* ensureCapacityInternal() method calls ensureExplicitCapacity(int minCapacity).
* ensureExplicitCapacity(int minCapacity) method calls grow(int minCapacity) method.
* grow(int minCapacity) method increments the size as

**int** newCapacity = oldCapacity + (oldCapacity >> 1);

* All elements from old Object[] is copied to new Object[] array.

1. What data structure is used in ArrayList class?

ArrayList<E> class uses [array](http://data-structure-learning.blogspot.com/p/array.html) data structure.

1. How many constructors are in class ArrayList and how to use them?

I have discussed about three different constructors that used in ArrayList<E> class.

* No-Arguments constructor: No argument constructor creates a default list with size 10.
* Initial Capacity as Argument: In this type of constructor we give an int value to constructor and it create the list of that size.
* Collection as Argument: This constructor create a list from Collection passed as parameter to constructor.
* You can read about it [here](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-constructors.html) in more detail.

1. How search element in ArrayList using API and without API?
2. How to get index of element in ArrayList using API and without API?
3. How to get lastIndexOf element in ArrayList using API and without API?
4. How to get element from index?
5. How to set element on some index?
6. How to add element on some index?
7. How to remove element by index?
8. How to remove element by Object?
9. How to clear the ArrayList?
10. How can you add some different Collection to ArrayList?addAll
11. How to remove some collection from ArrayList? removeAll
12. How to retain some collection from ArrayList? retainAll
13. How to traverse ArrayList in different ways? Iterator and listiterator
14. How to get sublist? Sublist
15. Iterate ArrayList using Lambda
16. How can you remove elements using Predicate? removeIf
17. How can you replace elements using Predicate? replaceAll
18. How to sort in natural order? Collections.sort
19. How to sort in reverse order? Collections.sort
20. How to reverse the list? Collections.reverse
21. How to shuffle list Collections.shuffle
22. How to fill list? Collections.fill
23. How to replace all instance of some element which some element? Collections.replaceAll
24. How to get indexOf Sublist? Collections.sublist
25. How to make it unmodifiable? Collections.unmodifiable
26. How to generate ncopies of elements of List? Collections.nCopies
27. Get frequency of element? Collections.frequency