

→ once ArrayList reaches its max capacity then a new ArrayList object will be created with

$$(\text{new Capacity} = \text{Current Capacity} * \frac{3}{2} + 1)$$

↳ 10, 16, 25, 38, ...

② → ArrayList l = new ArrayList (int initialCapacity)
→ creates an empty ArrayList object with specified initial capacity.

③ ArrayList l = new ArrayList (Collection c);
→ creates an equivalent ArrayList object for the given collection.

Ex:-

```
import java.util.*;
class ArrayListDemo
{
    public static void main (String[] args)
    {
        ArrayList l = new ArrayList();
        l.add("A");
        l.add(10);
        l.add("A");
        l.add(null);
        s.o.p(l);    → [A, 10, A, null]
        l.remove(2);
        s.o.p(l);    → [A, 10, null]
        l.add(2, "M");
        l.add("N");
        s.o.p(l);    → [A, 10, M, null, N].
    }
}
```

→ Usually, we can use collection to hold and transfer objects from one location to another location (containers). To provide support for this requirement, every collection class by default implements

- ArrayList and Vector classes implements RandomAccess interface so that any random element we can access with the same speed.

→ RandomAccess :-

- It is presents in java.util pkg.
 → It is a marker interface. it doesn't contain any method. Where required ability will be provided automatically by the JVM.

- ArrayList is the best choice if our frequent opⁿ is retrieval opⁿ (get) bcoz ArrayList implements RandomAccess interface.
 → ArrayList is the worst choice if our frequent opⁿ is @ insertion or deletion in the middle. bcoz of shifting of elements.

Diff b/w ArrayList & Vector :-

<u>ArrayList</u>	<u>Vector</u>
① Every method present in AL is non-synchronized	① Synchronized
② Not thread safe.	② Thread Safe.
③ performance is high	③ Performance down (low).
④ 1.2 v (non legacy)	④ 1.0 v (legacy class)

⇒ How to get Synchronized Version of ArrayList Object?

- By default ArrayList is non Synchronized. But we can get Synchronized Version of ArrayList Object by using SynchronizedList() method of Collections class.

```
public static List synchronizedList(List l);
```