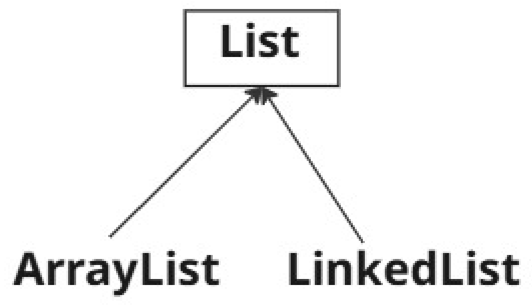
List Interface:-


List

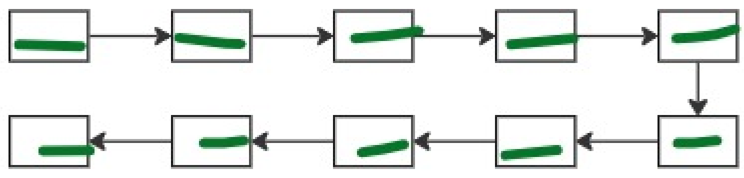
ArrayList LinkedList

1) ArrayList

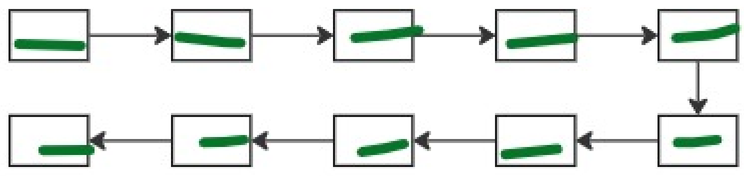
Arrays
=> size of the array is fixed
=> store only same type of data

ArrayList
=> It is called as resizeable array(size of the array can be increase OR decreased depending on the elements)
=> It can store same type of data OR different type of data

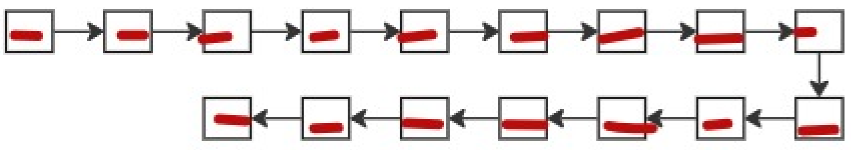
ArrayList al = new ArrayList() with a capacity of 10 elements



newcapacity = (oldcapacity * 3) / 2 + 1
newcapacity = 16



al ====>



==> It allows duplicate elements to be stored
==> It maintains the insertion order

In java whenever we print any reference variable, it print the address of that object because, internally when we print reference variable it automatically calls toString().

```
class Object {
        public void toString()
        {
          ======
          ======                    s.o.p(s1)
        }                                      add

}
        overide

        public void toString()
        {
          ========
        }
```

sort() fails to sort elements in the arraylist because, the elements in teh array list are student objects.
If we want to sort the objects, then take the help of another person along with sort() ==> Comparator(I)
Comparator(I) ==> compare() ==>

```
//   student1 = "Ramu", 14
//   student2 = "Mary", 13

public int compare(Student student1, Student student2) {
    student1.age - student2.age
}
```

14 - 13 ==> +ve ==> get swapped

"Mary 13", "Raju 14"

When to use ArrayList and when not to use ArrayList

==> We should never choose ArrayList if our operation frequently is INSERTION.
==> Use ArrayList if the operation like reading has to be done