Limitation of Object type Arrays:

Student[] s=new Student[1000];

- Fixed in size, can’t increase and can’t reduced.
- If size is don’t know in the advance
- Not developed on standard data structure.'
- Every requirement of code we have to write code. No readymade code is available.
- Only Student type of object can be added. Object type array can any object.
- Homogeneous data type objects.

Collections
----
1. Growable in nature
2. Homogeneous and heterogeneous
3. Classes and methods available

Arrays	**Collection**
Fixed in size	Growable in nature
Only Homogeneous object storage	Both Homogeneous and heterogeneous
No readymade method, No antistructure internally	Support available for readymade method
Memory point of view it is good	Good
Performance vise Arrays are good	not good with large data performance
Both primitives and objects	Only for objects not for primitives.

Collections (I): multiple objects types in a single roof.
1. Set- duplicate not allowed, order not maintained
    a. HashSet
        i. LinkedHashSet
    b. TreeSet
2. List - insertion order perserved, duplicate allowed
    a. ArrayList
    b. LinkedList
    c. Vector
    d. stack
3. Queue - prior to processing
    a. Deque
    b. PriorityQuery

Map(I) : group of key and values

Methods in Collections:

Add(object)
AddAll(Collections c)
Remove(Object)
removeAll(Collection c)
retainAll(Collection c)
Clear()
Contains(Object)
ContainsAll(Collections c)
Size()
toArray()
toArray(T[])
Iterator()

List

Index related method

Add(int index , object o)
addAll(int index,Collection c)
Get(int index)
Remove(in index)
Set(index I, Object new) // replace
indexOf(x) // first occurance will be return, else -1
lastIndex(o) //

ListIterator - iterate in both direction.

ArrayList:

Resizable array - underlying data structure - growable array
Duplicate allowed
Null allowed
Insertion order maintained
Heterogeneous object allowed

Constructor of ArrayList

AL a =new AL(); // 10 initial capacity
                    //10+1 - create new object, copy and re reference to new object

New capacity = (current capacity * 3/2)+1
                     = 10, 16,25…. Likewise increase

AL a= new AL(int capacity);

AL a = new AL(Collection c);