Arraylist → dynamic array, generic

Array → int arr[] = new int[n];

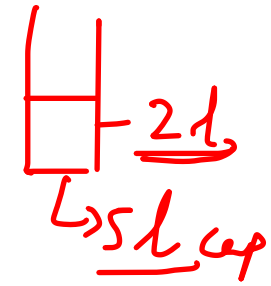
T arr[] = new T[n] ✗

Syntax

Arraylist <Datatype> arr = new Arraylist<>();

Integer, String, Boolean, Double, Character etc..

. size() → will show 0 initially.



size & Capacity

size ⇒ how much stored till now by user.

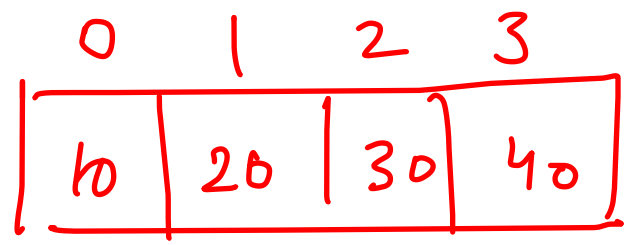
Capacity ⇒ max data stored // by default capacity is 10.

Insertion in Array list

→ . add(value);

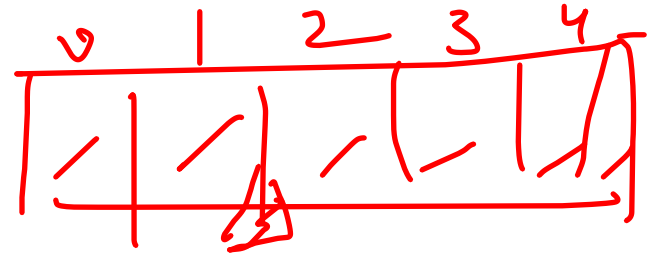
. add(10);

. add(20);



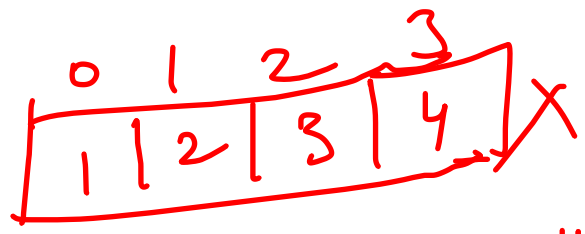
To access a particular elem at particular index

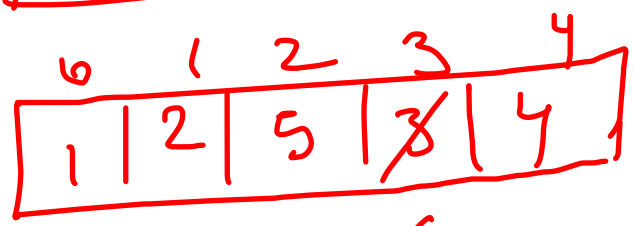
→ .get(index);



Add a elem at particular index

→ .add(index, element);





elem will get added & other values shift by one.

Remove element

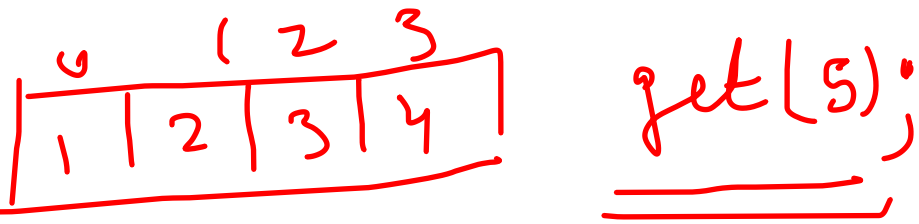
→ .remove(index);

change or replace value

→ .set(index, element);

Print all elements of arraylist

```
for( int i=0; i < list.size(); i++){
    System.out( list.get(i) + " ");
```



get(5);

// Arraylist also have index out of bound error.

```java
import java.util.ArrayList;

class Main {
    public static void main(String[] args) {
       ArrayList<Integer> list1 = new ArrayList<>();
       //System.out.println(list1.size());

       list1.add(5);
       list1.add(10);
       list1.add(15);
       System.out.println(list1.size());
       list1.add(2,30);
        System.out.println(list1.size());
        list1.remove(1);
        list1.set(1,50);

        for(int i = 0; i < list1.size(); i++){
            System.out.print(list1.get(i) + " ");
        }
        System.out.print(list1.get(5));
```

Output

```
3
4
5 50 15 ERROR!
Exception in thread "main" java.lang.IndexOutOfBoundsExcep
    of bounds for length 3
```

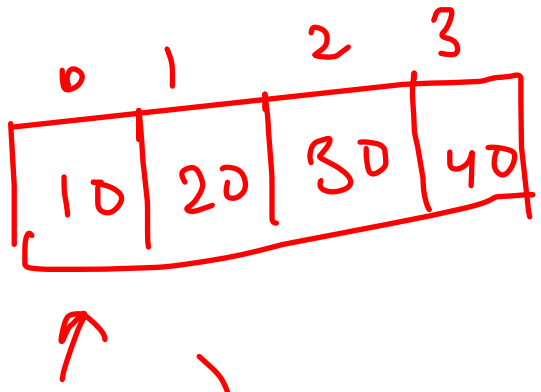
for-each loop.

for-each loop doesn't go to index, directly work on elements.

```
for( int i : list1){
    syso(i + " ");
}
```

}- Syntax

int i → for loop



for each

→ quick traversal

→ only for traversal

→ not for adding or changing values.

→ used with arrays also

Reverse a Arraylist

Collections.reverse(list);

TreeSet → unique
→ sorted
→ TC

merge two sorted lists

```
// Use a TreeSet to merge and keep only unique elements in sorted order
TreeSet<Integer> mergedSet = new TreeSet<>();

for (int i : A) {
    mergedSet.add(i);
}

for (int i : B) {
    mergedSet.add(i);
}

// Convert the TreeSet to an ArrayList
ArrayList<Integer> mergedList = new ArrayList<>(mergedSet);

for (int i : mergedList) {
    System.out.print(i + " ");
}
```

TC → O(log n)