

# Basic Implementation of ArrayList by me

if array is full  
double the size of  
array

default size of  
AL is 4

2 usages

```
public class MyArrayList {
```

7 usages

```
private int size;
```

11 usages

```
private int[] arr;
```

12 usages

```
private int idx;
```

2 usages

```
private void initialise(int sz){
```

```
    this.size=sz;
```

```
    arr=new int [sz];
```

```
    idx=0;
```

```
}
```

1 usage

```
public MyArrayList(){
```

```
    initialise( sz: 4);
```

```
}
```

2 usages

```
private void OutOfBoundException(int i) throws Exception{
```

```
    if(i<0||i>=this.size){
```

```
        throw new Exception("IndexOutOfBound");
```

```
    }
```

```
}
```

1 usage

```
public int get(int i) throws Exception{
```

```
    OutOfBoundException(i);
```

```
    return arr[i];
```

```
}
```

1 usage

```
private void addValue(int val){
```

```
    if(this.idx<=this.size-1){
```

```
        arr[idx]=val;
```

```
        idx++;
```

```
        return;
```

```
    }
```

```
    else{
```

```
        int[] temp=new int [this.size];
```

```
        for(int i=0;i<this.size;i++){
```

```
            temp[i]=arr[i];
```

```
        }
```

```
        initialise( sz: 2*this.size);
```

```
        for(int i=0;i<temp.length;i++){
```

```
            arr[i]=temp[i];
```

```
            this.idx++;
```

```
        }
```

```
        arr[idx]=val;
```

```
        idx++;
```

```
        return;
```

```
    }
```

```
}
```

7 usages

```
public void add(int val){
```

```
    addValue(val);
```

```
}
```

New Let's modify it on Base of  
Sumeet Sir lecture

added a set function

Moreover if remove krne ke  
band

if array ke Elements =  $\frac{\text{size}}{4}$

then we make arr of size  
new [size/2] & copy values of  
old array to new array

```
48 public String toString(){
49     StringBuilder s=new StringBuilder();
50     s.append("[");
51     for(int i=0;i<idx;i++){
52         s.append(arr[i]);
53         if(i!=idx-1)
54             s.append(",");
55     }
56     s.append("]");
57     return s.toString();
58 }
59 5 usages
60 public int size(){
61     return this.size;
62 }
63 2 usages
64 public int remove(int i)throws Exception{
65     OutOfBoundsException(i);
66     int val=arr[i];
67     for(int j=i;j<this.idx;j++){
68         arr[j]=arr[j+1];
69     }
70     arr[this.idx-1]=0;
71     idx--;
72     return val;
73 }
```

```

1  public class MyArrayList {
    9 usages
2      private int size;
    13 usages
3      private int[] arr;
    14 usages
4      private int idx;
5
    3 usages
6      private void initialise(int sz){
7          this.size=sz;
8          arr=new int [sz];
9          idx=0;
10     }
    1 usage
11     public MyArrayList(){
12         initialise( sz: 4);
13     }
    3 usages
14     private void OutOfBoundException(int i) throws Exception{
15         if(i<0||i>=this.size){
16             throw new Exception("IndexOutOfBound");
17         }
18     }

```

```

19     public int get(int i) throws Exception{
20         OutOfBoundException(i);
21         return arr[i];
22     }
    1 usage
23     private void addValue(int val){
24         if(this.idx==this.size){
25             System.out.println("Resize up");
26             int[] temp=new int [this.size];
27             for(int i=0;i<this.size;i++){
28                 temp[i]=arr[i];
29             }
30             initialise( sz: 2*this.size);
31             for(int i=0;i<temp.length;i++){
32                 arr[i]=temp[i];
33                 this.idx++;
34             }
35         }
36         arr[idx]=val;
37         idx++;
38     }
    7 usages
39     public void add(int val){
40         addValue(val);
41     }

```



```

43 public String toString(){
44     StringBuilder s=new StringBuilder();
45     s.append("[");
46     for(int i=0;i<idx;i++){
47         s.append(arr[i]);
48         if(i!=idx-1)
49             s.append(",");
50     }
51     s.append("]");
52     return s.toString();
53 }
7 usages
54 public int size(){
55     return this.size;
56 }

```

Remove ke  
 h'e bri  
 aley & fr  
 here a  
 here the  
h'v'ate

```

82 public void set(int i,int val) throws Exception{
83     OutOfBoundException(i);
84     setvalue(i,val);
85 }
86 }
87

```

```

57 public int remove(int i)throws Exception{
58     OutOfBoundException(i);
59     int val=arr[i];
60     for(int j=i;j<this.idx;j++){
61         arr[j]=arr[j+1];
62     }
63     arr[this.idx-1]=0;
64     idx--;
65     if(idx==size/4){
66         System.out.println("Resize down");
67         int[] temp=new int [this.idx];
68         for(int j=0;j<=this.idx-1;j++){
69             temp[j]=arr[j];
70         }
71         initialise( sz: this.size/2);
72         for(int j=0;j<temp.length;j++){
73             arr[j]=temp[j];
74             this.idx++;
75         }
76     }
77     return val;
78 }
1 usage

```

Main

```
1  ▶ public class Main {  
2  ▶      public static void main(String[] args) throws Exception {  
3      MyArrayList al=new MyArrayList();  
4      System.out.println(al+" size is "+al.size());  
5      al.add(1);  
6      al.add(2);  
7      System.out.println(al+" size is "+al.size());  
8      al.add(3);  
9      al.add(4);  
10     System.out.println(al+" size is "+al.size());  
11     al.add(5);  
12     al.add(6);  
13     al.add(7);  
14     System.out.println(al+" size is "+al.size());  
15     al.remove(3);  
16     System.out.println(al+" size is "+al.size());  
17     al.remove(0);  
18     System.out.println(al+" "+al.get(2));  
19     al.set(3,42);  
20     al.remove(1);  
21     System.out.println(al+" size is "+al.size());  
22     al.remove(1);  
23     al.remove(1);  
24     System.out.println(al+" size is "+al.size());  
25     }  
26 }
```

O/p

```
[ ] size is 4  
[1,2] size is 4  
[1,2,3,4] size is 4  
Resize up  
[1,2,3,4,5,6,7] size is 8  
[1,2,3,5,6,7] size is 8  
[2,3,5,6,7] 5  
[2,5,42,7] size is 8  
Resize down  
[2,7] size is 4
```

Single Array list

different when

