⇒ Arraylist (dynamic in nature)

array → static

arraylist → dynamic

Note:- arraylist can only store objects

Ex. of objects:- Integer, Character, String, Boolean .... etc.

Syntax :-

```
Arraylist < DataType>  arr = new Arraylist<>();
```

↳

```
Arraylist<Integer> arr = new Arraylist<>();
```

(now, we have an arraylist of size zero)

⇒ Inbuilt functions // ArrayList<Integer> arr = new ArrayList<>();

1) add :- arr.add(value);

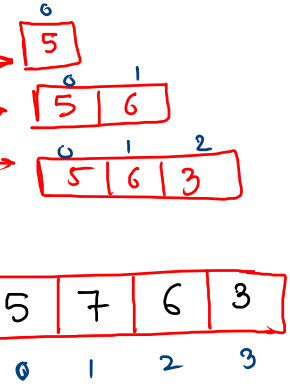
2) add :- arr.add(index, value);

```
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arr = new ArrayList<>(); // size = 0

        // add function
        arr.add(5); // size = 1
        arr.add(6); // size = 2
        arr.add(3); // size = 3

        arr.add(1, 7); // size = 4
    }
}
```

index value



2) get // ArrayList<Integer> arr = new ArrayList<>();

arr.get(index); // to access the value at 'index'

3) size // to get the length of arraylist

arr.size();

Code

```
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arr = new ArrayList<>(); // size = 0
        // add function
        arr.add(5); // size = 1
        arr.add(6); // size = 2
        arr.add(3); // size = 3
        arr.add(1, 7); // size = 4
        arr.add(1, 8); // size = 5
        for (int i = 0; i < arr.size(); i++) {
            System.out.print( arr.get(i) + " " );
        }
    }
}
```

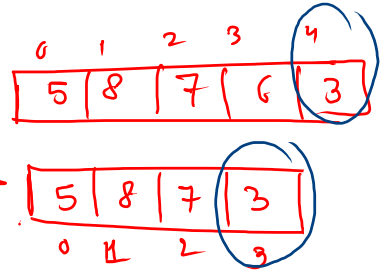
4) remove

arr.remove(index);

// used to remove memory of 'index'

code

```
public class Main {
    public static void main(String[] args) {
        ArrayList<Integer> arr = new ArrayList<>(); // size = 0
        // add function
        arr.add(5); // size = 1
        arr.add(6); // size = 2
        arr.add(3); // size = 3
        arr.add(1, 7); // size = 4
        arr.add(1, 8); // size = 5
        arr.remove(3);  // size = 4
        for (int i = 0; i < arr.size(); i++) {
            System.out.print( arr.get(i) + " " );
        }
    }
}
```



ArrayList Printing

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    ArrayList<Integer> arr = new ArrayList<>();
    for (int i = 0; i < n; i++) {
        int val = scn.nextInt();
        arr.add(val);
    }

    // print arraylist using for loop
    for (int i = 0; i < arr.size(); i++) {
        System.out.print( arr.get(i) + " " );
    }
    System.out.println();
    // print arraylist using for each loop
    for (Integer i : arr) {
        System.out.print(i + " ");
    }
}
```

input

ArrayList reverse printing

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    ArrayList<Integer> arr = new ArrayList<>();
    for (int i = 0; i < n; i++) {
        int val = scn.nextInt();
        arr.add(val);
    }

    // print reverse using for loop
    for (int i = arr.size() - 1; i >= 0; i--) {
        System.out.print( arr.get(i) + " " );
    }
    System.out.println();

    Collections.reverse(arr);
    for (Integer i : arr) {
        System.out.print(i + " ");
    }
}
```

How to reverse an arraylist

Collections.reverse(arr);

O(n)

→ Inbuilt function

To sort an array :- `Arrays.sort(arr);`

To sort an arraylist :- → O(nlogn)

↳ `Collections.sort(arr);` ↑ ing

↳ `Collections.sort(arr, Collections.reverseOrder());`

↓ ing

→ Revese :-

↳ `Collections.reverse(arr);` → O(n)

ArrayList with if-else

Code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    ArrayList<Integer> arr = new ArrayList<>();
    int t = scn.nextInt();
    for (int i = 0; i < t; i++) {
        int c = scn.nextInt();
        if ( c == 1 ) {
            printSize(arr);
        } else if ( c == 2 ) {
            printAndRemoveFromLast(arr);
        } else if ( c == 3 ) {
            int x = scn.nextInt();
            printAndAddAtLast(arr, x);
        } else if ( c == 4 ) {
            printAndRemoveFromFirst(arr);
        } else if ( c == 5 ) {
            int x = scn.nextInt();
            printAndAddAtBeginning(arr, x);
        } else if ( c == 6 ) {
            print(arr);
        } else {
            System.out.println("invalid-move");
        }
    }
}
```

T.C O(t)

1)
```java
public static void printSize(ArrayList<Integer> arr) {
    int ans = arr.size();
    System.out.println(ans);
}
```

2)
```java
public static void printAndRemoveFromLast(ArrayList<Integer> arr) {
    if ( arr.size() == 0 ) {
        System.out.println("invalid-move");
        return;
    }
    int val = arr.get( arr.size() - 1 );
    arr.remove(arr.size() - 1);
    System.out.println(val);
}
```

3)
```java
public static void printAndAddAtLast(ArrayList<Integer> arr, int x) {
    System.out.println(x);
    arr.add(x);
}
```

4)
```java
public static void printAndRemoveFromFirst(ArrayList<Integer> arr) {
    if ( arr.isEmpty() == true ) {
        System.out.println("invalid-move");
        return;
    }
    int val = arr.get(0);
    arr.remove(0);
    System.out.println(val);
}
```

5)
```java
public static void printAndAddAtBeginning(ArrayList<Integer> arr, int x) {
    System.out.println(x);
    arr.add(0, x);
}
```

6)
```java
public static void print(ArrayList<Integer> arr) {
    if ( arr.size() == 0 ) {
        System.out.println("invalid-move");
        return;
    }
    for (Integer i : arr) {
        System.out.print(i + " ");
    }
    System.out.println();
}
```