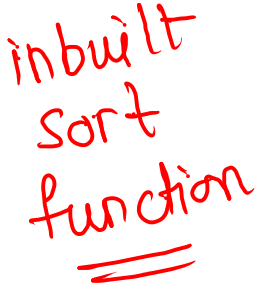
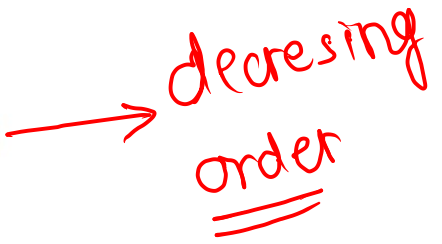
increasing order using inbuilt sort

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

    Arrays.sort(arr);

    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}
```



decreasing order using inbuilt sort

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    Integer[] arr = new Integer[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    Arrays.sort(arr, Collections.reverseOrder());

    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}
```



Custom Sort

↳ concept which is used to modify a given function accordingly.

logic

1) Comparable
2) Comparator

```java
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output t
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    Integer[] arr = new Integer[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    Arrays.sort(arr, new myComparator());

    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}

public static class myComparator implements Comparator<Integer> {
    @Override
    public int compare(Integer a, Integer b) {
        return b - a;
    }
}
```

syntax

logic

```java
public static class myComparator implements Comparator<Integer> {
    @Override
    public int compare(Integer a, Integer b) {

    }
}
```

self

other

syntax (OOPs)

syntax

short version :- lambda function

asc

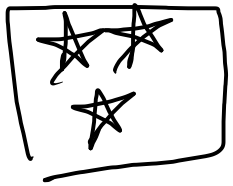
return a-b ; // smaller value will come first

des

return b-a ; // larger value will come first

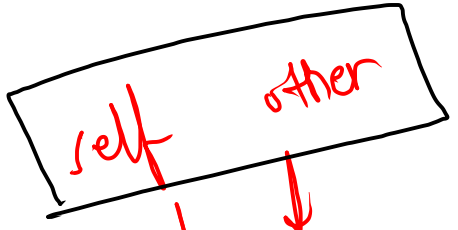
} Java

⇒ lambda function



// definition

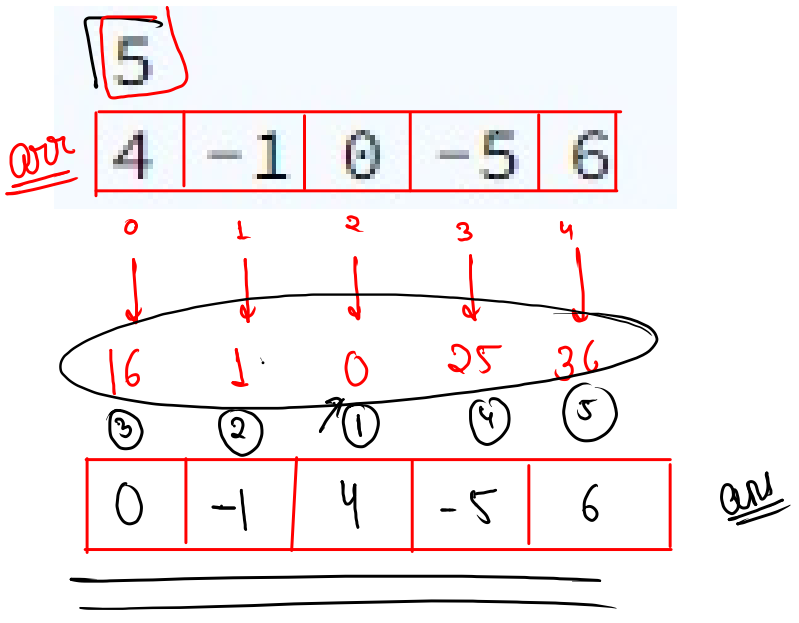
```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    Integer[] arr = new Integer[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    Arrays.sort(arr, (a, b) -> {    // lambda function
        return b - a;
    });

    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}
```



increasing: a - b } logic

decreasing: b - a

Sort the array according to their Square of each element



n=5

Comparable / lambda / concept

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

    Arrays.sort( arr, (a, b) -> {   // lambda function
        return a * a - b * b;
    } );

    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}
```

logic