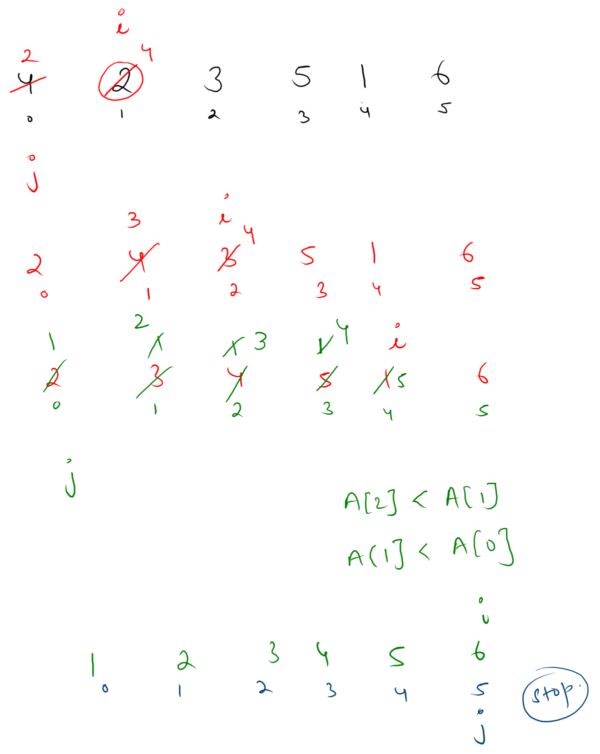
Insertion Sort

* correct position for current value.



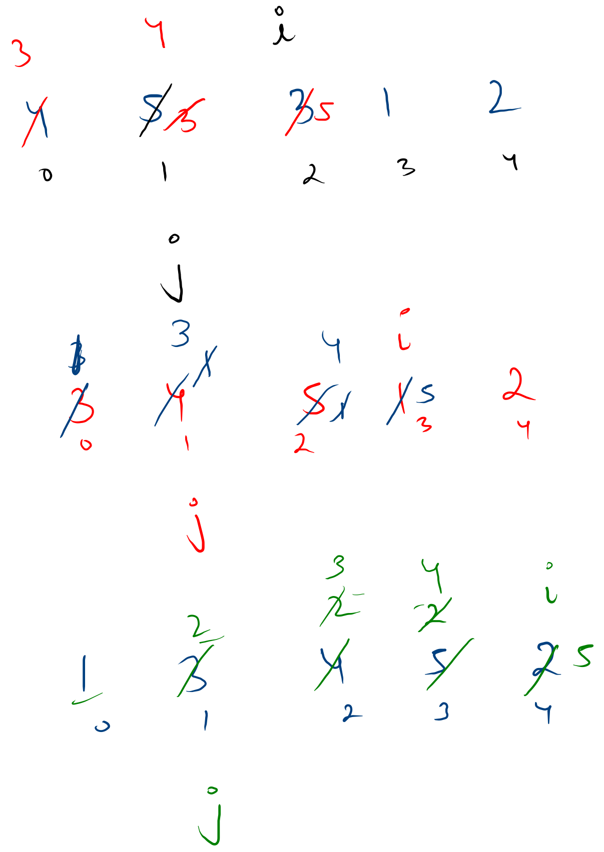
$A[2] < A[1]$

$A[1] < A[0]$

stop

$A[5] < A[4]$

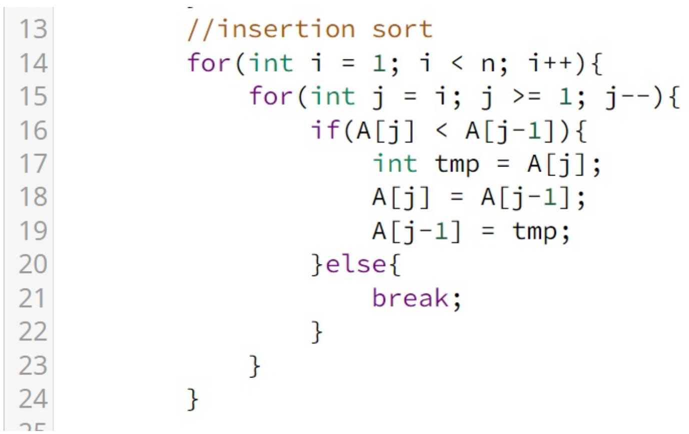
$$\begin{tikzcd}
& \boxed{A[j] < A[j-1]} \arrow[dl] \arrow[dr] & \\
\text{true} \arrow[d] & & \text{false} \arrow[r] & \text{stop} \\
\text{decrement } j & & &
\end{tikzcd}$$



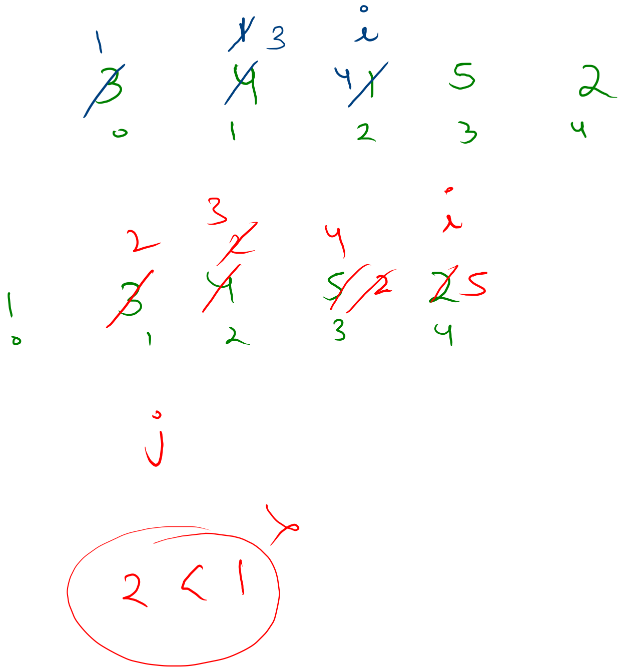
$2 < 5$

$A[j] < A[j-1]$

$2 < 3$

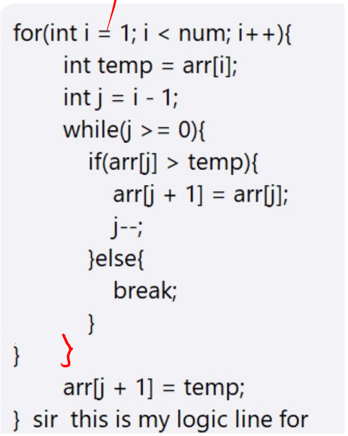


```
        //insertion sort
        for(int i = 1; i < n; i++){
            for(int j = i; j >= 1; j--){
                if(A[j] < A[j-1]){
                    int tmp = A[j];
                    A[j] = A[j-1];
                    A[j-1] = tmp;
                }else{
                    break;
                }
            }
        }
```



```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int [] A = new int[n];
    for(int i = 0; i < n; i++){
        A[i] = scn.nextInt();
    }
    //insertion sort
    for(int i = 1; i < n; i++){
        for(int j = i; j >= 1; j--){
            if(A[j] < A[j-1]){
                int tmp = A[j];
                A[j] = A[j-1];
                A[j-1] = tmp;
            }else{
                break;
            }
        }
    }

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



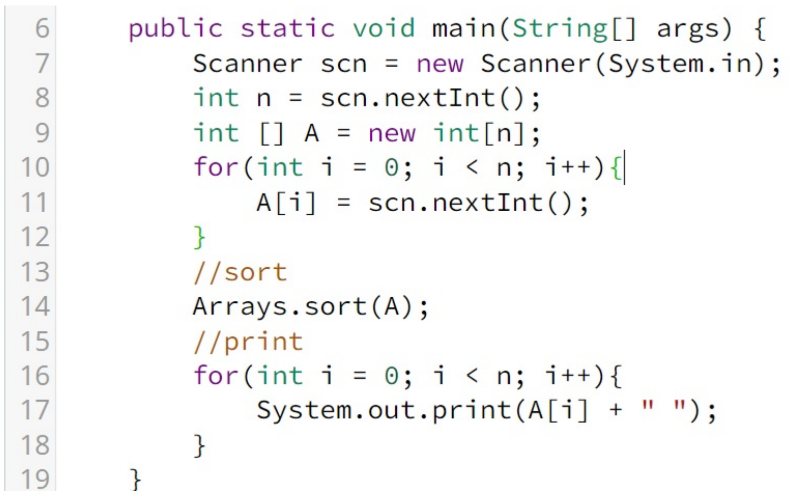
```
for(int i = 1; i < num; i++){
       int temp = arr[i];
       int j = i - 1;
       while(j >= 0){
          if(arr[j] > temp){
             arr[j + 1] = arr[j];
             j--;
          }else{
             break;
          }
}     }
       arr[j + 1] = temp;
}  sir  this is my logic line for
```

```
4    5    3    2    1
0    1    2    3    4
     i
```

tmp =

increasing order using inbuilt sort

$$\begin{tikzcd}
\underline{\text{Java}} \arrow[r] \arrow[d] & \text{function} \arrow[r] & \text{sort} \\
\text{default} & \text{behaviour} \arrow[r] & \boxed{\underline{\text{asc.}}} \ \star\star\star
\arrow[from=2-1, to=2-1]
\end{tikzcd}$$



```java
    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int [] A = new int[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        //sort
        Arrays.sort(A);
        //print
        for(int i = 0; i < n; i++){
            System.out.print(A[i] + " ");
        }
    }
```

TC $\rightarrow$ $O(n\log n)$

SC $\rightarrow$ $O(1)$

$$\left.\begin{array}{l}\text{bubble sort}\\ \text{insertion sort}\\ \text{selection sort}\end{array}\right\} \quad \begin{array}{c}\text{TC}\\ \left.O(n^2)\right\}\end{array} \quad \begin{array}{c}\text{SC}\\ O(1)\end{array}$$

$$\begin{tikzcd}
 & \text{Comparator} \arrow[dl] \arrow[dr] \arrow[r] & \text{issue} \\
\text{inbuilt (Java)} & & \text{myOwn (User)}
\end{tikzcd}$$

* Collection Framework

↳ Objects

int [ ] A

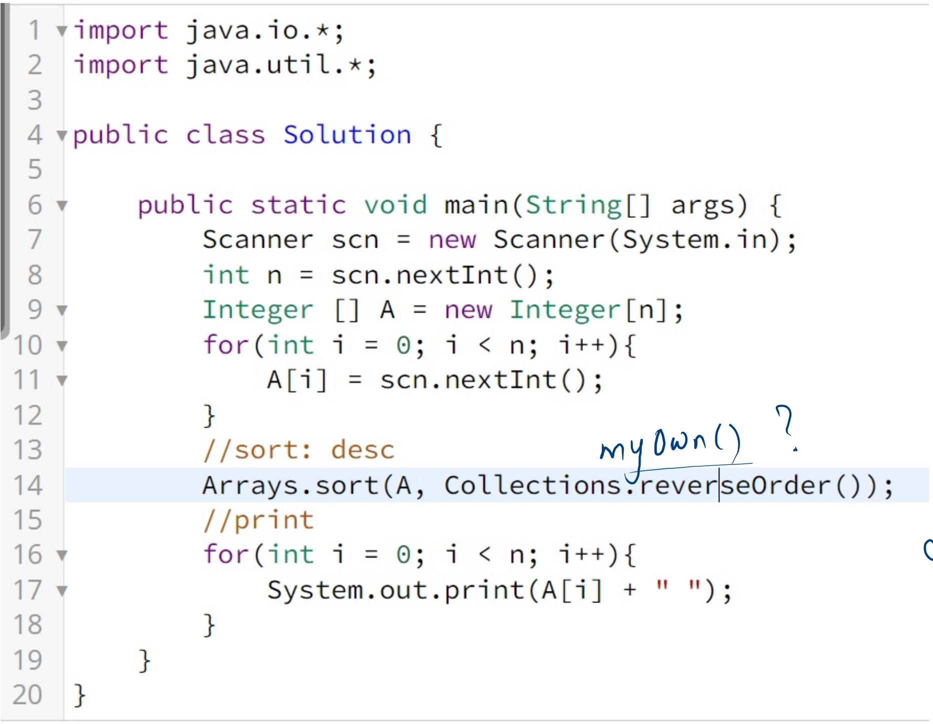
↳ primitive data-type

↳ ! obj

* Comparator will not work for ~~int~~ array.

$\downarrow$

Integer



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        Integer [] A = new Integer[n];
        for(int i = 0; i < n; i++){
            A[i] = scn.nextInt();
        }
        //sort: desc
        Arrays.sort(A, Collections.reverseOrder());
        //print
        for(int i = 0; i < n; i++){
            System.out.print(A[i] + " ");
        }
    }
}
```

myOwn() ?

sum( ) ✓

function( )

Comparator $\longrightarrow$ inbuilt

Math. maximum

~~Math. max~~

Comparator → contains the logic that helps to compare two objects at a time.

Syntax → to write the comparator.

Comparator

↳ has a function

↓

compare()

↳ logic

* * *

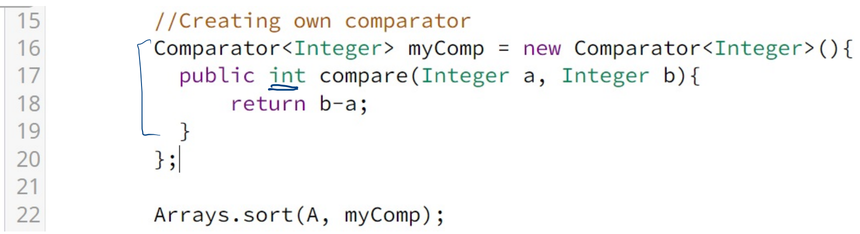
Classes & Objects → Constructor

Wrapper Class

Collections.

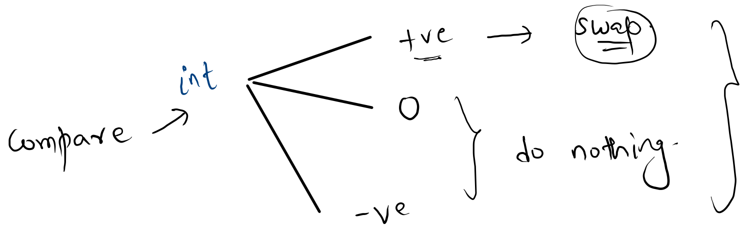
Interface

Comparator.



```java
//Creating own comparator
Comparator<Integer> myComp = new Comparator<Integer>(){
  public int compare(Integer a, Integer b){
      return b-a;
  }
};

Arrays.sort(A, myComp);
```



$\boxed{a-b} \rightarrow$ asc. $\rightarrow$ 2 4

$a=2$
$b=4$

$\boxed{2-4} = -ve$

$\boxed{b-a} \rightarrow$ desc. 4 2

4 2
~~2~~ ~~4~~
a b

$4-2 = +ve$