

### ArrayList:

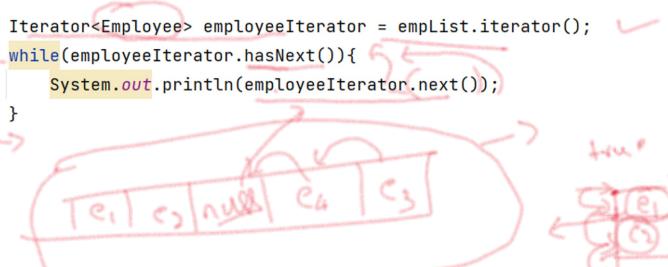
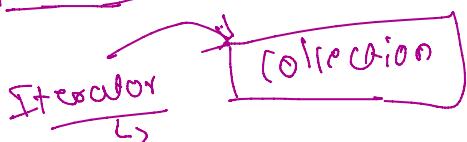
ArrayList < Integer > list =  
new ArrayList<>();

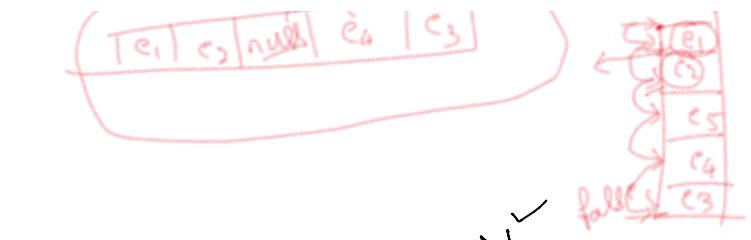
or String, float / Double, Employee → CRUD  
Employee → Set

ArrayList<Object> = Integer;  
String;  
Employee;

Iterate over a Collection

- 1) For loop → index
- 2) For each loop.
- 3) Iterator method.





Sorting of Employee

id  
name  
Sal  
desig

numbers → natural order  
strings → alphabetical

ArrayList.sort()

e<sub>1</sub>      e<sub>2</sub>      -

Student → USN      Sear no.  
name  
Sem  
cgpa      percent

Employee objects are not comparable

10 < 20

e<sub>1</sub>      e<sub>2</sub>      abc < def  
e<sub>1</sub> < e<sub>2</sub>

Employee

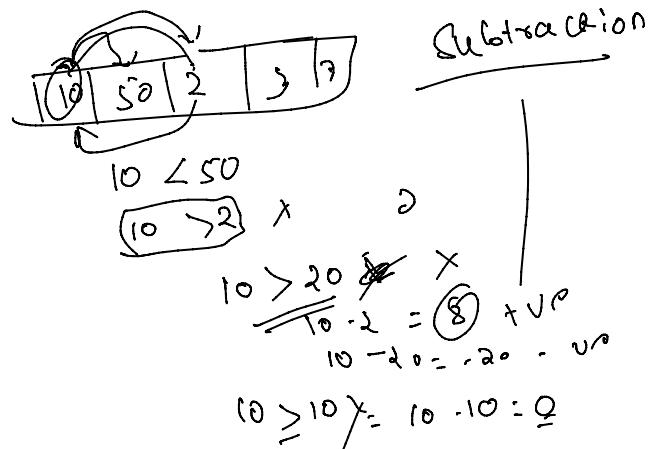
id  
name  
Sal  
designation

Comparable Interface

c<sub>1</sub>.compareTo(c<sub>2</sub>)

c<sub>1</sub>.cmpId      c<sub>2</sub>.cmpId.  
other Emp. o'

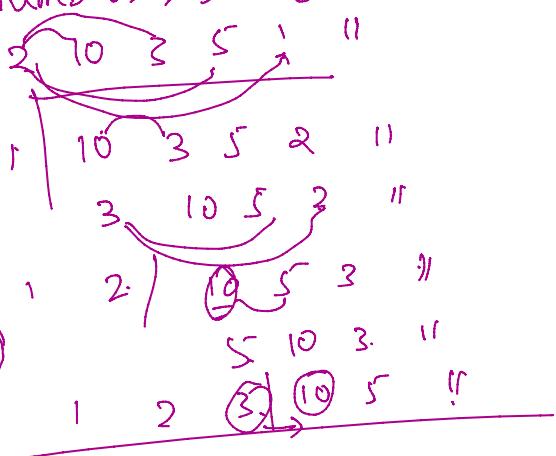
$c_1$   
 $c_1.empId$        $c_2.empId$   
 this  
 emp  
 $this.empId \rightarrow c_1.empId$



$c_1 < e_4$   
 $c_1.empId - e_4.empId \rightarrow 0.$   
 this

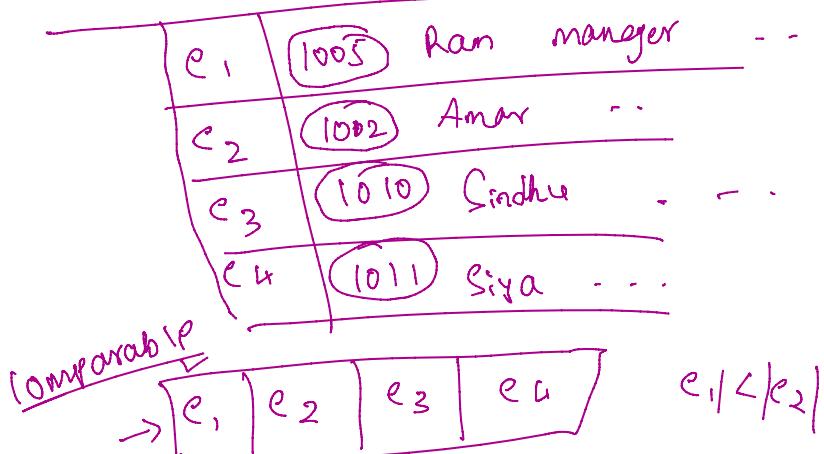
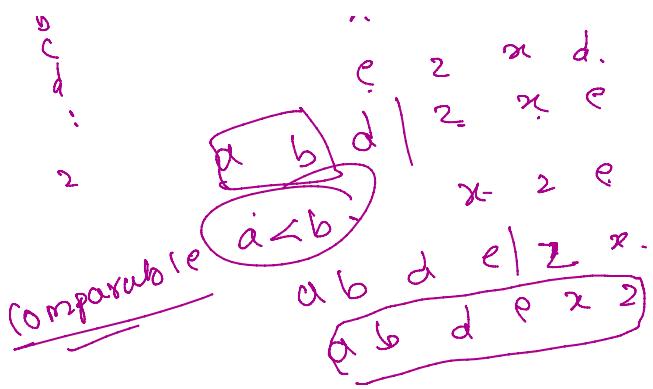
ArrayList.sort();

numbers, strings is

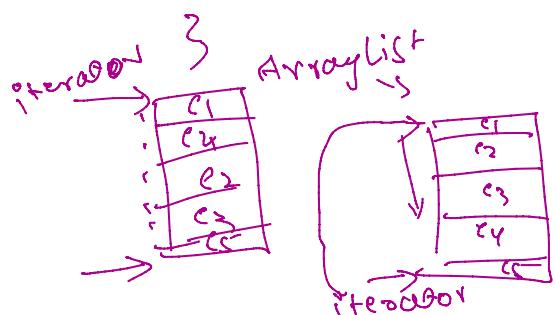
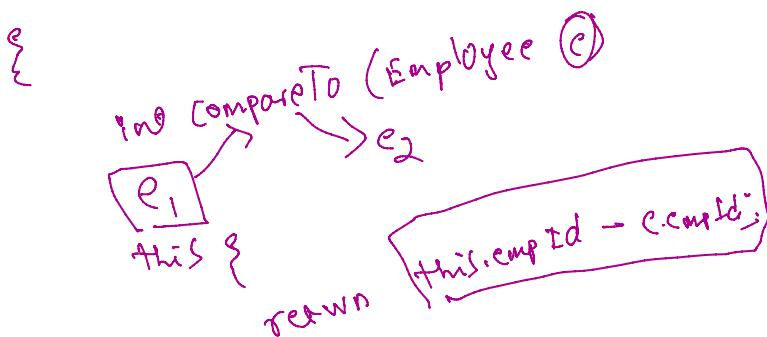


Comparable

$a \leq b \mid 2 \leq e \leq d$   
 $b \leq c \leq d$   
 $c \leq d \leq e$



class Employee implements Comparable<Employee>

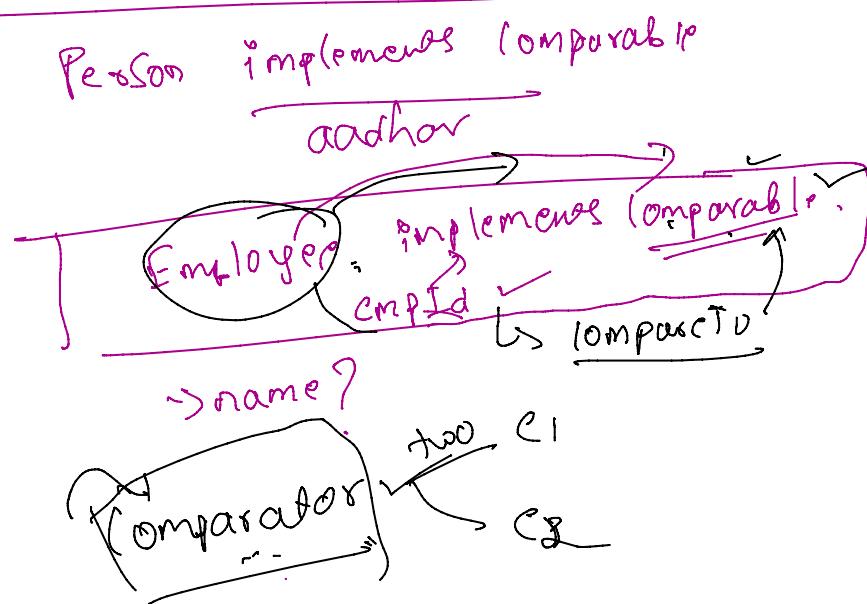


\* Create class Person with  

- addressNumber
- Name
- State
- occupation

 → Create ArrayList of person &  
 ↳ run operations.

- O(n^2)
- Create ArrayList of Person & do CRUD operations.
- Sort the persons based on Author number & display



(class) EmployeeNameComparator implements Comparable<Employee>

@Override  
int compare(Employee emp1, Employee emp2)

{  
return emp1.getName().compareTo(emp2.getName());}

}

\* By Id  
† By Name

- 1) Sort Em by their Salary
- 2) Sort Employees by their Designation.

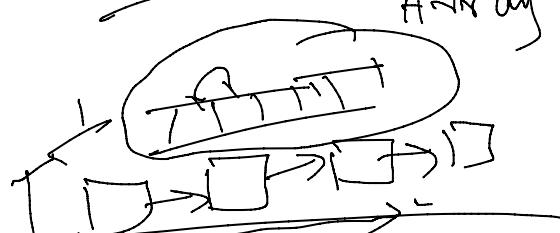
Ask user for a choice:  
To Sort the employees based on

- 1) Emp Id ✓
- 2) By Name
- 3) By Salary
- 4) By Designation.

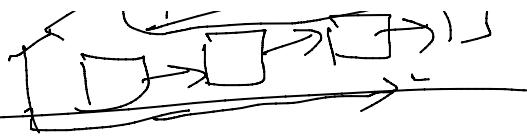
Use

LinkedList  
(Collection)

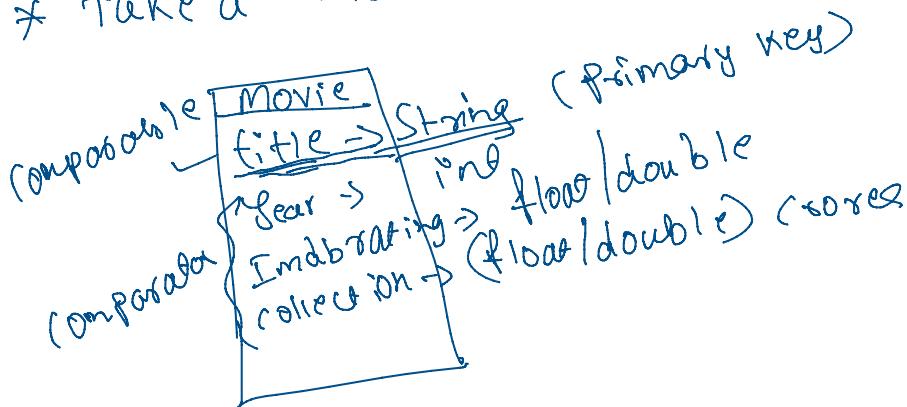
ArrayList



→ By Design



\* Take a Movie class



Sort Selection

- 1) Title
- 2) Year of Release
- 3) Imdb (reverse)
- 4) Collection (reverse)

Employee implements Comparable

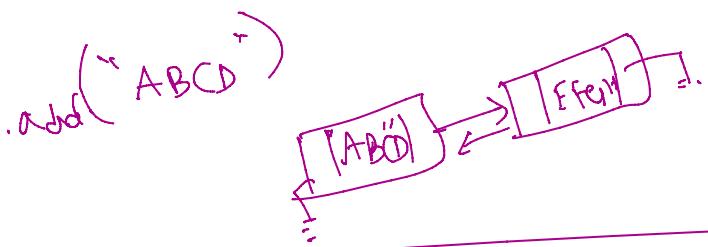
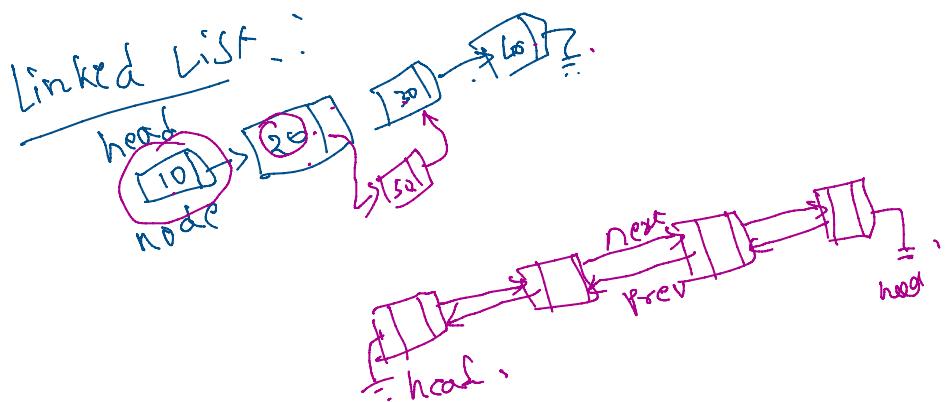
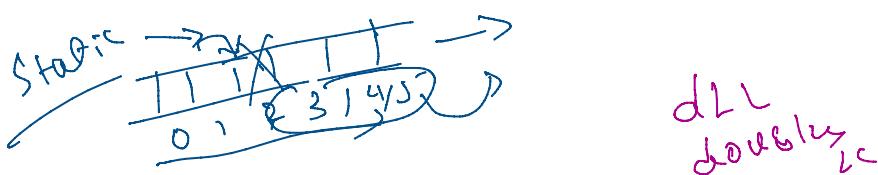
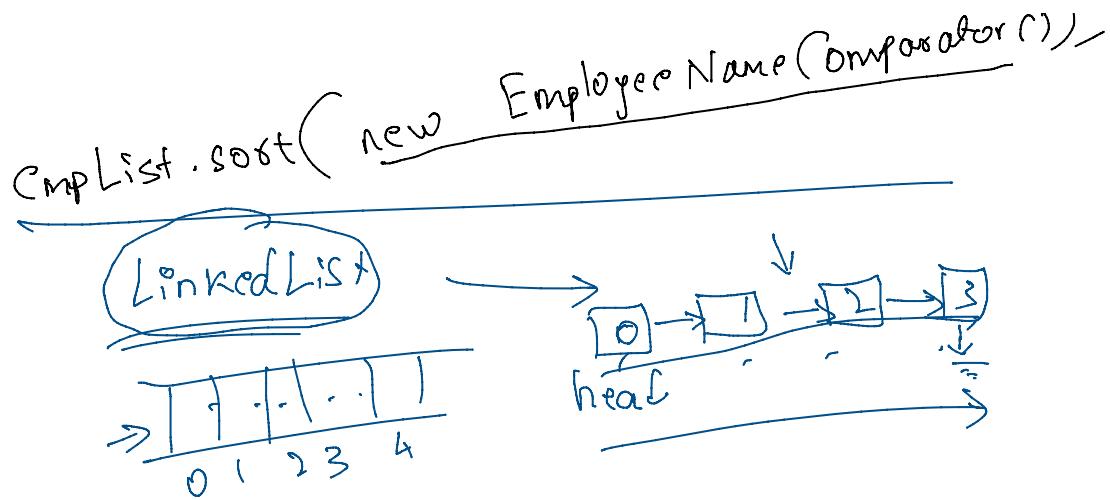
@Override  
compareTo( ) // Natural Sorting order!

empId  
USN  
Ranjithan -  
row → Chassis no./eng.no

(name:)  
Employee Name Comparator implements Comparator<Employee>

3  
1. compare(Emp e1, Emp e2)  
2. equals(Emp e1, Emp e2)  
e1.getName().compareTo(e2.getName())

3  
Employee Name Comparator();



```

float imdb;
float compare()
{
    float o1.getImdb - o2.getImdb);
    if (o1.getImdb - o2.getImdb) <= 0.8
        return 1;
    else
        return 0;
}

```

Vector ✓  
Stack  
Copy On Write ArrayList  
Thread Safe ✓