**Sorting an Array of Employee Object -Arrays.sort() method**

**we have an array of custom objects of type employee**

We want to sort all the employees in an array by their salary field in ascending order -

//To do that we create the **Employee class** which implements comparable Interface and we need to compareTo() method.

**So We need to**

1) create a class employee.like private int ID,salary, Name.

2) Create a getter setter methods to access these private fields and 3) Create parameterized constructor to right click source and

4) Create a toString() method to right click source and generate toString

Now we need to Implement a comparable interface .The reason is that Arrays.sort() method requires an object type to Implement comparable interface.

so that array can be sorted according to the natural ordering of its elements .

The natural ordering is defined by implementation of compareTo() method which determines **how the current object to be compared to another object of the same type.**

**S0**

**1) Implement Comparable interface and pass the Employee type**

**2) Create object of Employee Class and in this object we have to include all info about ID,Name and Salary.**

**3) In the Employee class –**

**a) Implements Comparable Interface to use compareTo() in which we can sort the current object to be compared to another object of the same type.**

**package** com.array.collection;

**public** **class** Employee **implements** Comparable<Employee>{

//Sorting an Array of Employee Object-Arrays.sort() method

**private** **int** ID;

**private** String Name;

**private** **int** salary;

**public** Employee(**int** iD, String name, **int** salary) {

**super**();

**this**.ID = iD;

**this**.Name = name;

**this**.salary = salary;

}

**public** **int** getID() {

**return** ID;

}

**public** **void** setID(**int** iD) {

ID = iD;

}

**public** String getName() {

**return** Name;

}

**public** **void** setName(String name) {

Name = name;

}

**public** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

@Override // to print like this

**public** String toString() {

**return** "Employee [ID=" + ID + ", Name=" + Name + ", salary=" + salary + "]";

}

//It comes from comparable interface

@Override // make a logic based on Salary.

**public** **int** compareTo(Employee o) {

return this.salary - o.salary;

}

}

**package** com.array.collection;

**import** java.util.Arrays;

**public** **class** Sorting {

**public** **static** **void** main(String[] args) {

//Sorting an Array of primitives-Arrays.sort() method

**int** [] array = {10,202,32,12,34};

System.***out***.println("Before Sorting : "+Arrays.*toString*(array)); // use Arrays.toString() to print Array to console

Arrays.*sort*(array);

System.***out***.println("After Sorting : "+Arrays.*toString*(array));

//Sorting an Array of String-Arrays.sort() method

String stringArray []= {"Afsheen", "Taseen", "Shaheen","Sumiya"};

System.***out***.println("Before Sorting : "+Arrays.*toString*(stringArray));

Arrays.*sort*(stringArray);

System.***out***.println("After Sorting : "+Arrays.*toString*(stringArray));

**//Sorting an Array of Employee objects- Array.sort() method**

**Employee [] employees = {**

**new Employee(10,"Mr X", 40000),**

**new Employee(20,"Mr y", 120000),**

**new Employee(30,"Mr z", 70000),**

**new Employee(40,"Mr D", 20000)**

**};**

**System.*out*.println("Before Sorting : "+Arrays.*toString*(employees));**

**Arrays.*sort*(employees);**

**System.*out*.println("After Sorting : "+Arrays.*toString*(employees));**

**}**

**}**