

















All Domains > Tutorials > Cracking the Coding Interview > Sorting: Comparator

Sorting: Comparator **■**



Problem

Submissions

Leaderboard

Discussions

Editorial

Check out the resources on the page's right side to learn more about sorting. The video tutorial is by Gayle Laakmann McDowell, author of the best-selling interview book Cracking the Coding Interview.

Comparators are used to compare two objects. In this challenge, you'll create a comparator and use it to sort an array. The *Player* class is provided in the editor below; it has two fields:

- 1. A string, name.
- 2. An integer, score

Given an array of *n Player* objects, write a comparator that sorts them in order of decreasing score; if **2** or more players have the same score, sort those players alphabetically by name. To do this, you must create a *Checker* class that implements the *Comparator* interface, then write an *int compare(Player a, Player b)* method implementing the Comparator.compare(T o1, T o2) method.

Input Format

Locked stub code in the Solution class handles the following input from stdin:

The first line contains an integer, n, denoting the number of players.

Each of the n subsequent lines contains a player's respective name and score.

Constraints

- $0 \le score \le 1000$
- Two or more players can have the same name.
- Player names consist of lowercase English alphabetic letters.

Output Format

You are not responsible for printing any output to stdout. Locked stub code in *Solution* will create a *Checker* object, use it to sort the *Player* array, and print each sorted element.

Sample Input

5 amy 100 david 100 heraldo 50 aakansha 75 aleksa 150

Sample Output

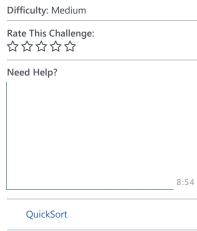
aleksa 150 amy 100 david 100 aakansha 75 heraldo 50

Explanation

As you can see, the players are first sorted by decreasing score and then sorted alphabetically by name.

f ⊮ in

Submissions: 4793 Max Score: 35



```
More
 Current Buffer (saved locally, editable) \ \mathscr{V} \ \mathfrak{O}
                                                                                         Java 7
 1 import java.util.*;
 2
     // Write your Checker class here
 3
    class Checker implements Comparator<Player> {
 5
         @override
 6
         public int compare(Player a, Player b) {
             int diff = b.score - a.score;
 8
              return diff != 0 ? diff : a.name.compareTo(b.name);
 9
10
11 v class Player{
12
        String name;
13
        int score;
14
15 🔻
        Player(String name, int score){
            this.name = name;
16
            this.score = score;
17
18
19
   1
20
21 ▼ class Solution {
22
23 '
        public static void main(String[] args) {
24
            Scanner scan = new Scanner(System.in);
25
            int n = scan.nextInt();
26
27
            Player[] player = new Player[n];
28
            Checker checker = new Checker();
29
            for(int i = 0; i < n; i++){
30 ▼
31
                player[i] = new Player(scan.next(), scan.nextInt());
32
33
            scan.close();
34
35
            Arrays.sort(player, checker);
36 ▼
            for(int i = 0; i < player.length; i++){</pre>
37
                System.out.printf("%s %s\n", player[i].name, player[i].score);
38
39
        }
40
   }
                                                                                                                  Line: 10 Col: 2
```

<u>**1**</u> <u>Upload Code as File</u> □ Test against custom input

Run Code

Submit Code

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature