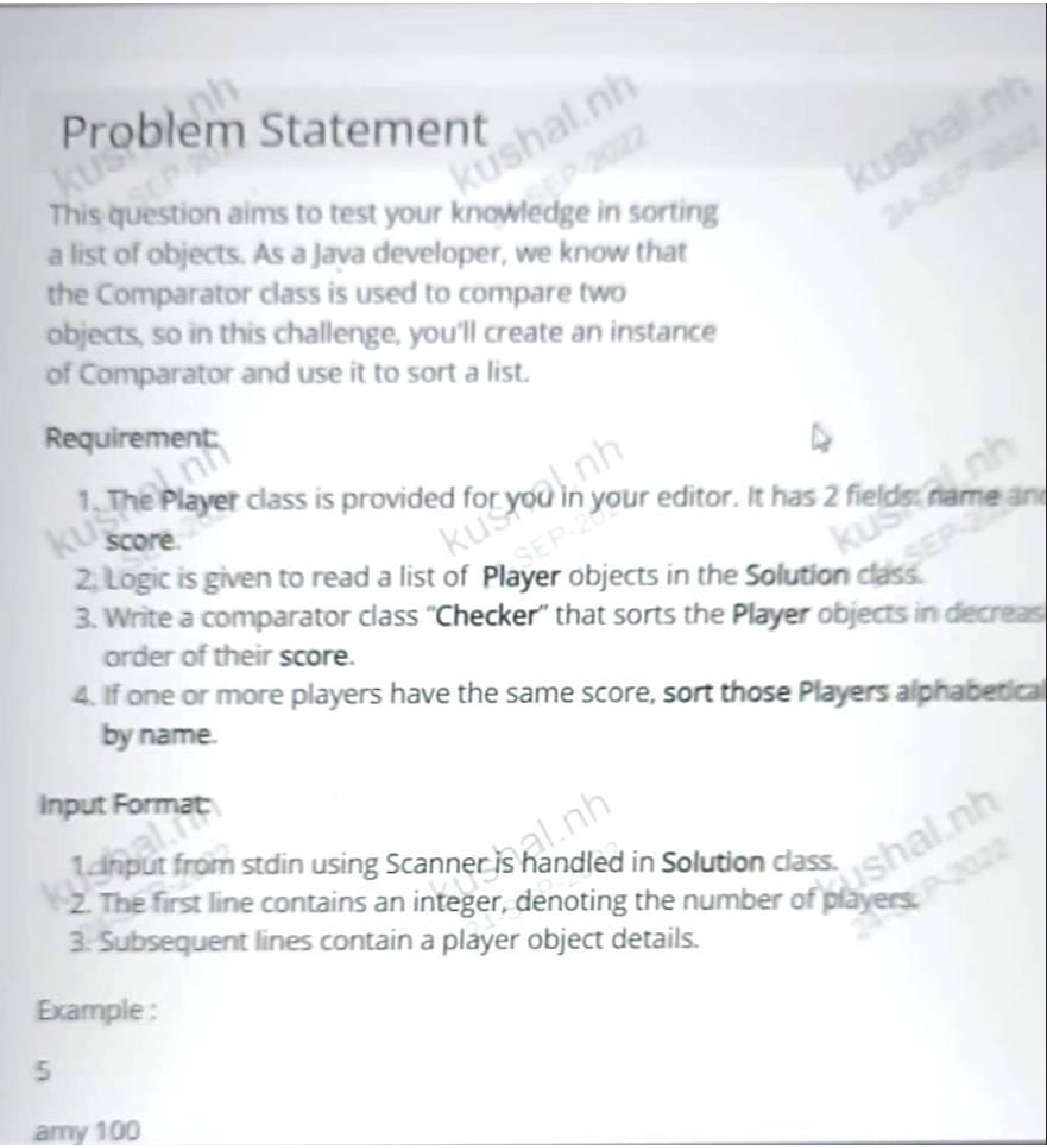


Problem Statement

This question aims to test your knowledge in sorting a list of objects. As a Java developer, we know that the Comparator class is used to compare two objects, so in this challenge, you'll create an instance of Comparator and use it to sort a list.

Requirement:

1. The **Player** class is provided for you in your editor. It has 2 fields: **name** an **score**.
2. Logic is given to read a list of **Player** objects in the **Solution** class.
3. Write a comparator class "**Checker**" that sorts the **Player** objects in decreas order of their **score**.
4. If one or more players have the same score, **sort those Players alphabetical by name**.

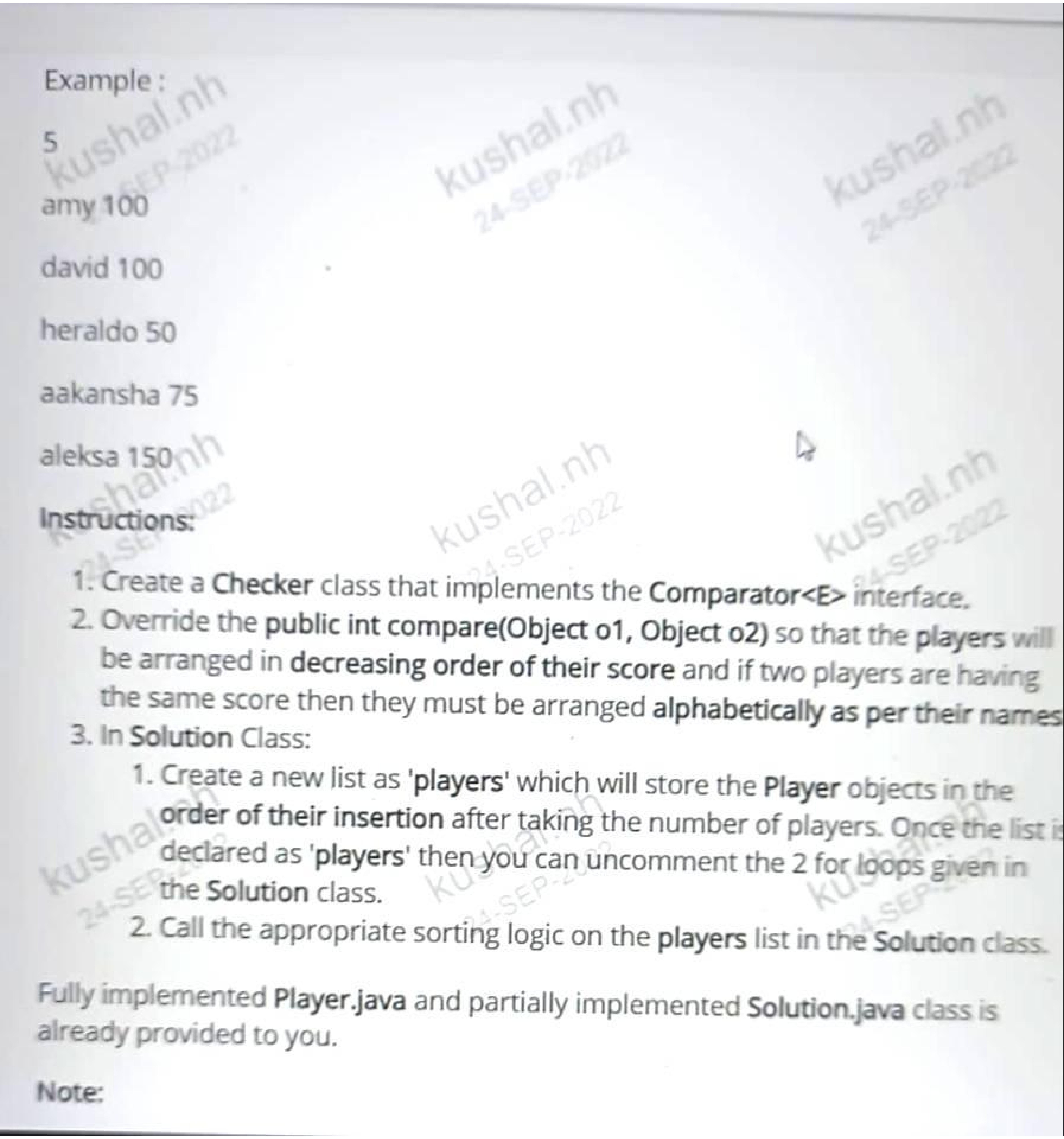
Input Format:

1. Input from stdin using Scanner is handled in **Solution** class.
2. The first line contains an integer, denoting the number of players.
3. Subsequent lines contain a player object details.

Example :

5

amy 100



Example :

5

amy 100

david 100

heraldo 50

aakansha 75

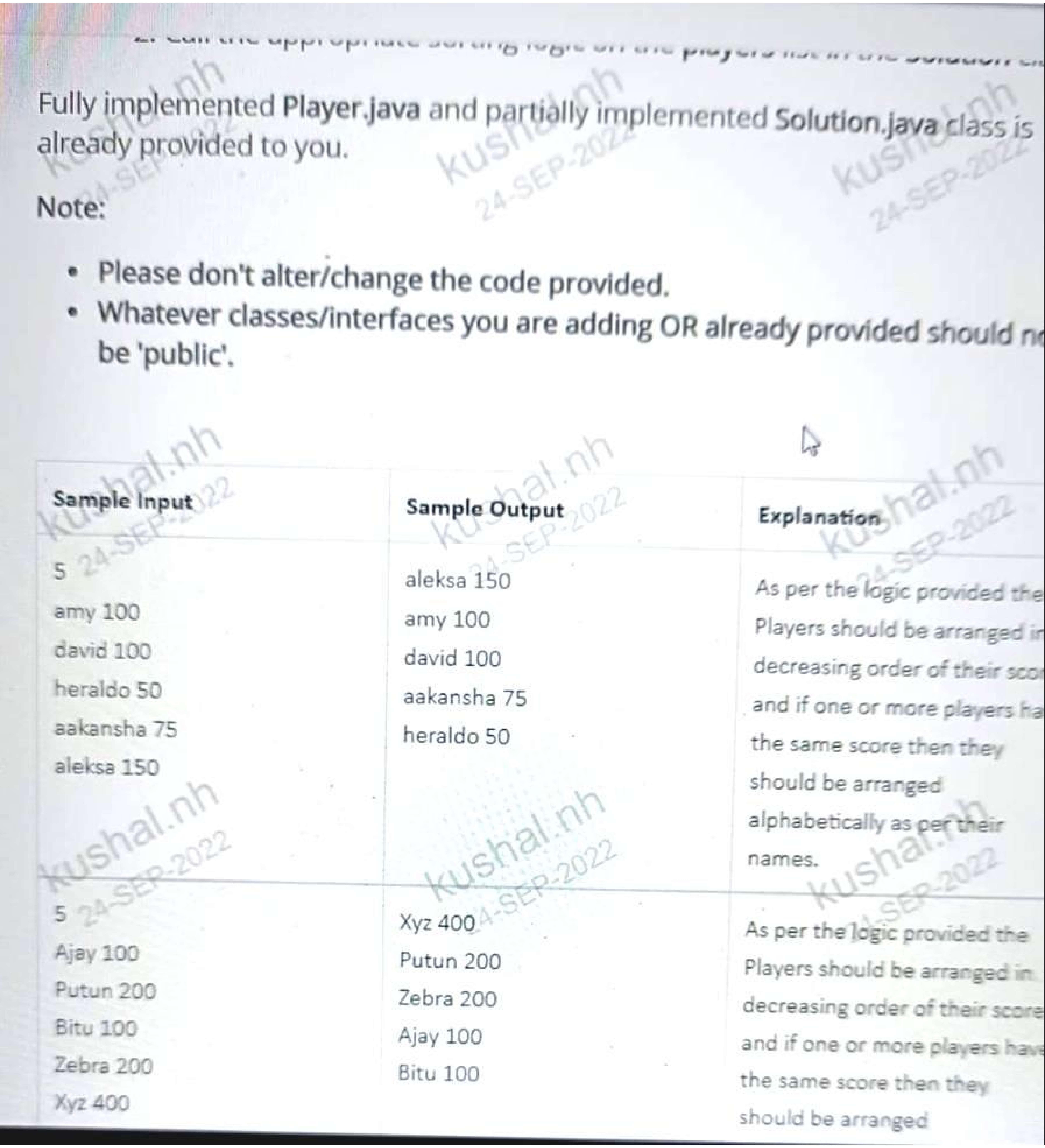
aleksa 150

Instructions:

1. Create a **Checker** class that implements the **Comparator<E>** interface.
2. Override the **public int compare(Object o1, Object o2)** so that the **players** will be arranged in **decreasing order of their score** and if two players are having the same score then they must be arranged **alphabetically as per their names**
3. In **Solution** Class:
    1. Create a new list as '**players**' which will store the **Player** objects in the **order of their insertion** after taking the number of players. Once the list i declared as '**players**' then you can uncomment the 2 for loops given in the **Solution** class.
    2. Call the appropriate sorting logic on the **players** list in the **Solution** class.

Fully implemented **Player.java** and partially implemented **Solution.java** class is already provided to you.

Note:



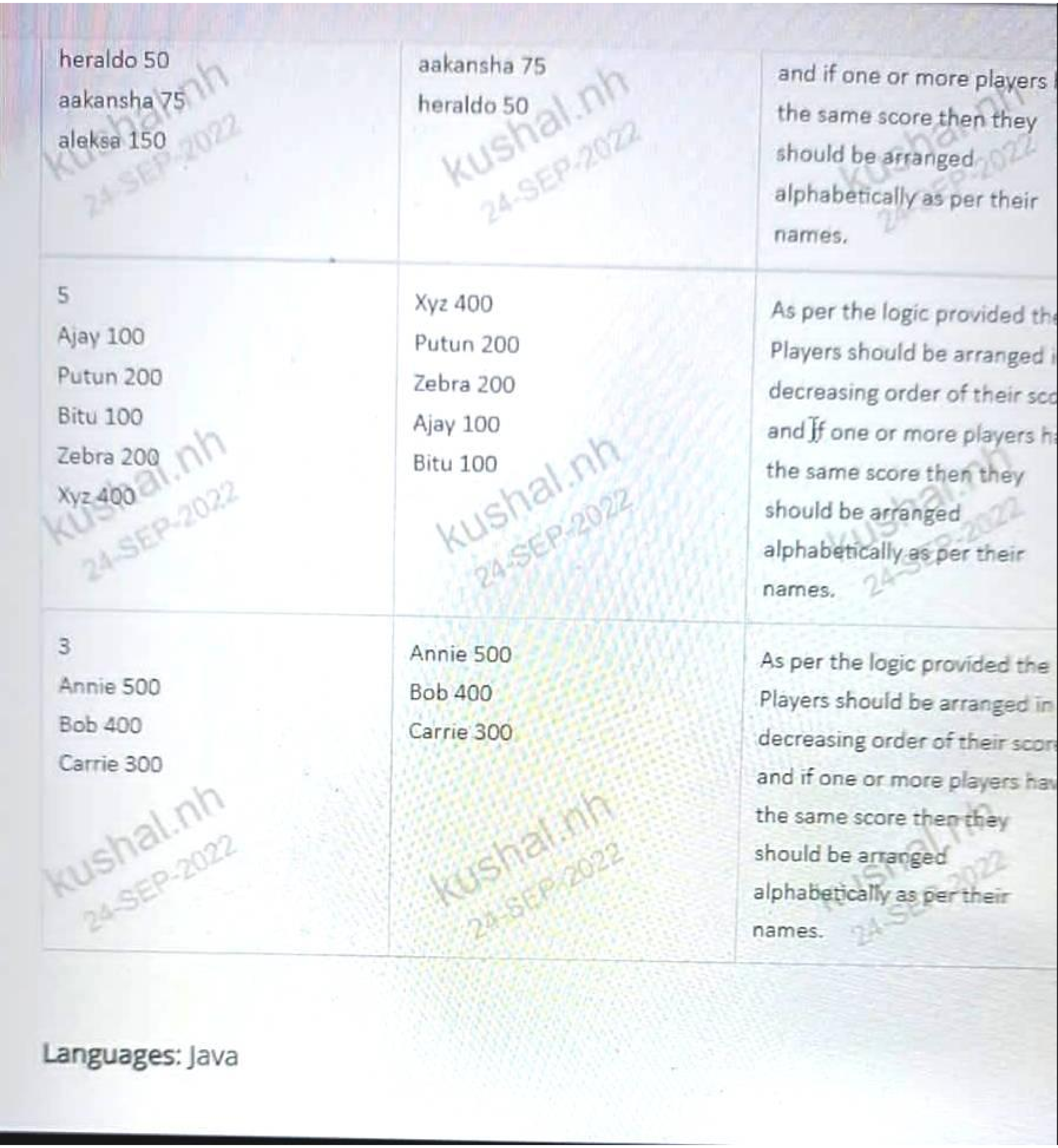
2. Call the appropriate sorting logic on the players list in the Solution cl

Fully implemented **Player.java** and partially implemented **Solution.java** class is already provided to you.

Note:

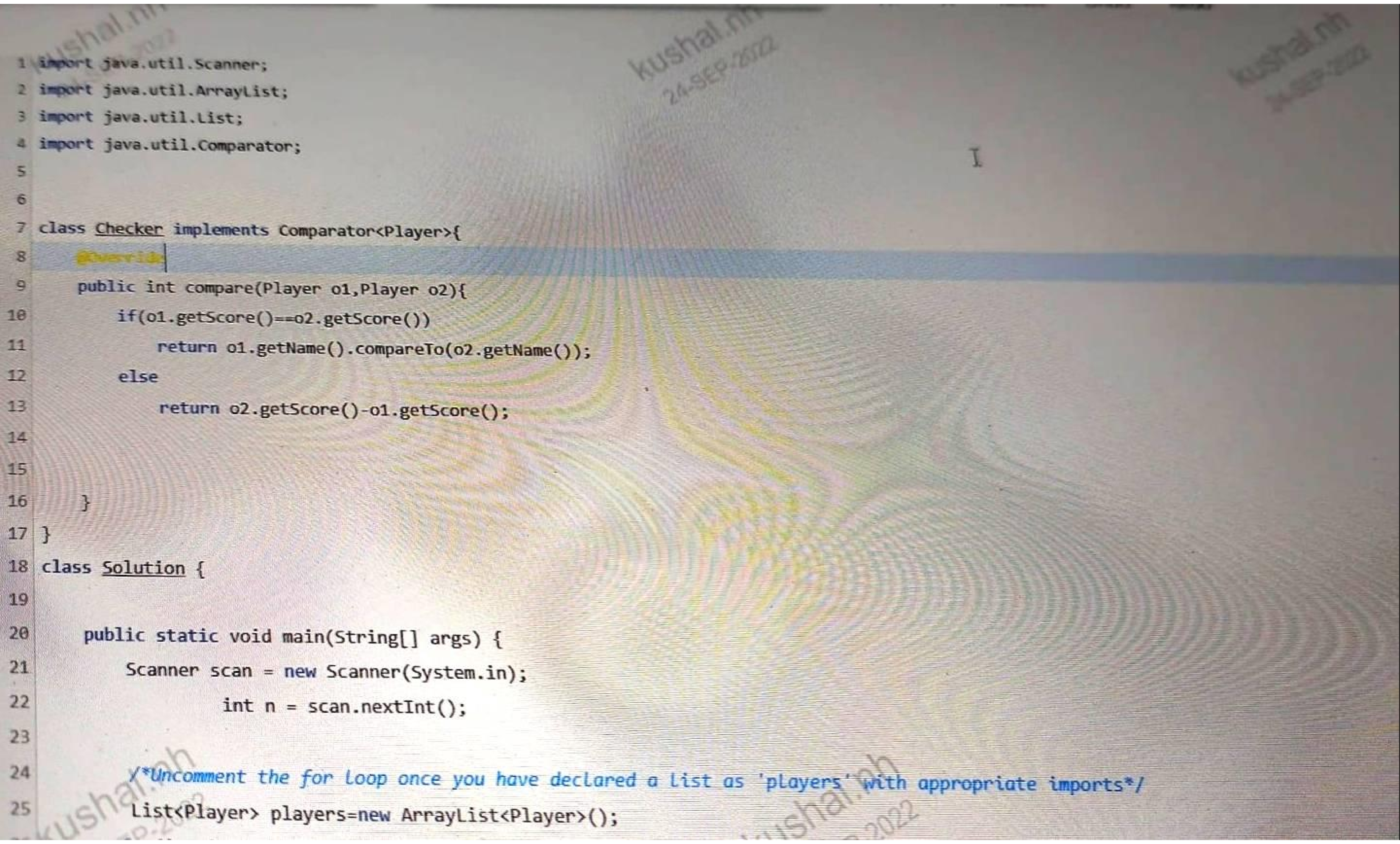
- Please don't alter/change the code provided.
- Whatever classes/interfaces you are adding OR already provided should n be 'public'.

Sample Input	Sample Output	Explanation
5 amy 100 david 100 heraldo 50 aakansha 75 aleksa 150	aleksa 150 amy 100 david 100 aakansha 75 heraldo 50	As per the logic provided the Players should be arranged in decreasing order of their sco and if one or more players ha the same score then they should be arranged alphabetically as per their names.
5 Ajay 100 Putun 200 Bitu 100 Zebra 200 Xyz 400	Xyz 400 Putun 200 Zebra 200 Ajay 100 Bitu 100	As per the logic provided the Players should be arranged in decreasing order of their score and if one or more players hav the same score then they should be arranged

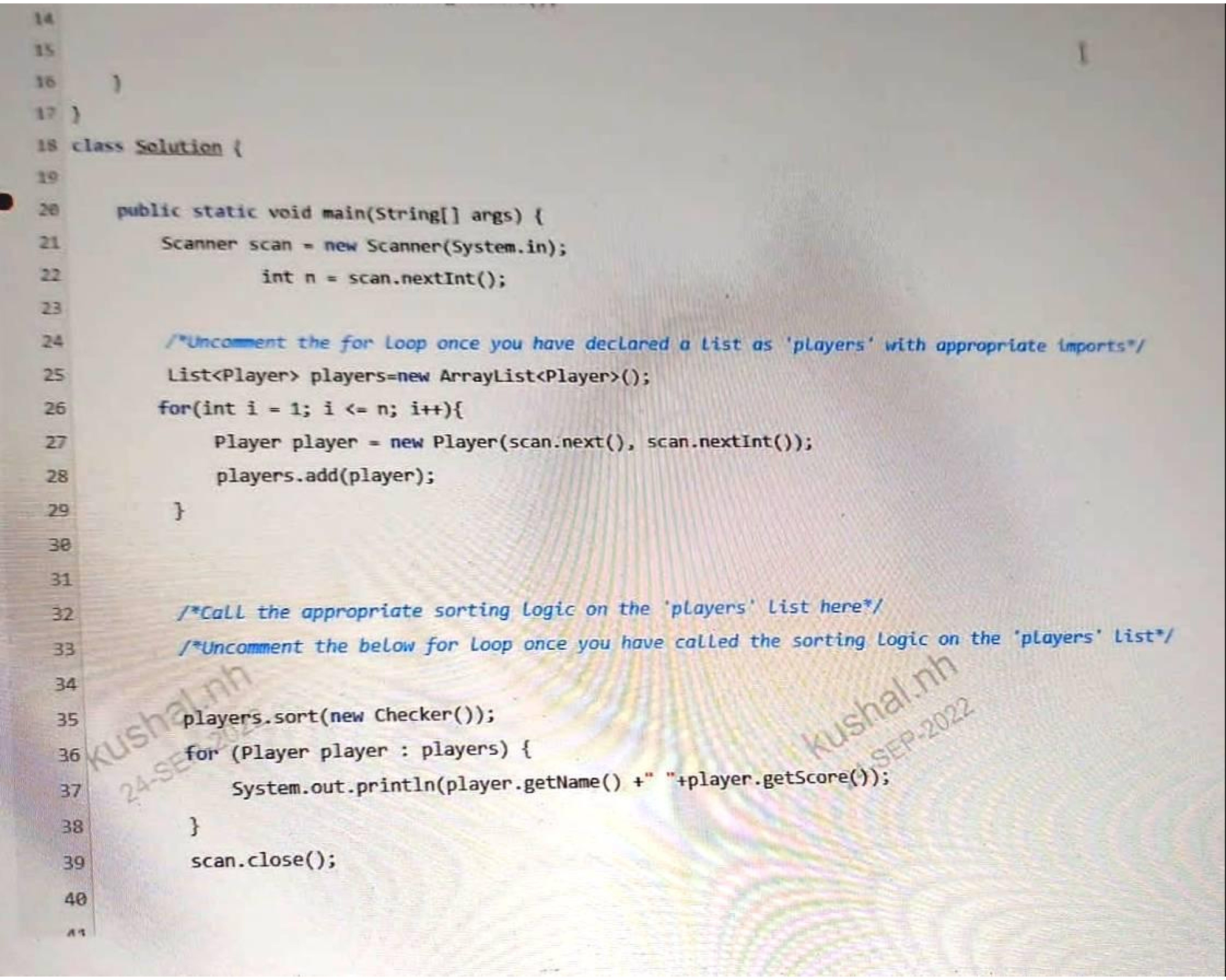


heraldo 50 aakansha 75 aleksa 150	aakansha 75 heraldo 50	and if one or more players the same score then they should be arranged alphabetically as per their names.
5 Ajay 100 Putun 200 Bitu 100 Zebra 200 Xyz 400	Xyz 400 Putun 200 Zebra 200 Ajay 100 Bitu 100	As per the logic provided th Players should be arranged i decreasing order of their sco and If one or more players h the same score then they should be arranged alphabetically as per their names.
3 Annie 500 Bob 400 Carrie 300	Annie 500 Bob 400 Carrie 300	As per the logic provided the Players should be arranged in decreasing order of their scor and if one or more players ha the same score then they should be arranged alphabetically as per their names.

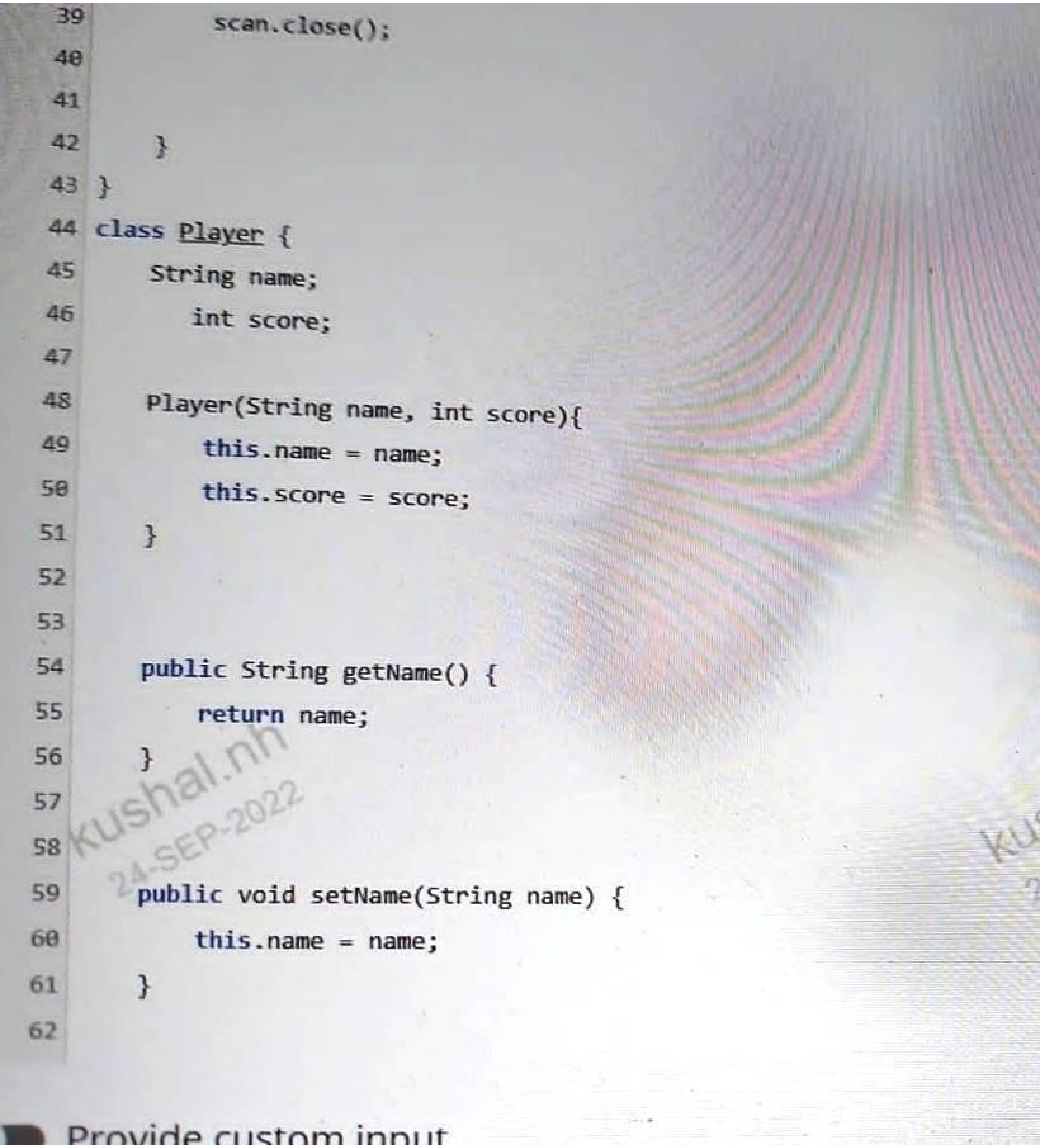
Languages: Java



```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;
import java.util.Comparator;


class Checker implements Comparator<Player>{
    @Override
    public int compare(Player o1,Player o2){
        if(o1.getScore()==o2.getScore())
            return o1.getName().compareTo(o2.getName());
        else
            return o2.getScore()-o1.getScore();


    }
}
class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
                int n = scan.nextInt();

        /*Uncomment the for Loop once you have declared a List as 'players' with appropriate imports*/
        List<Player> players=new ArrayList<Player>();
```

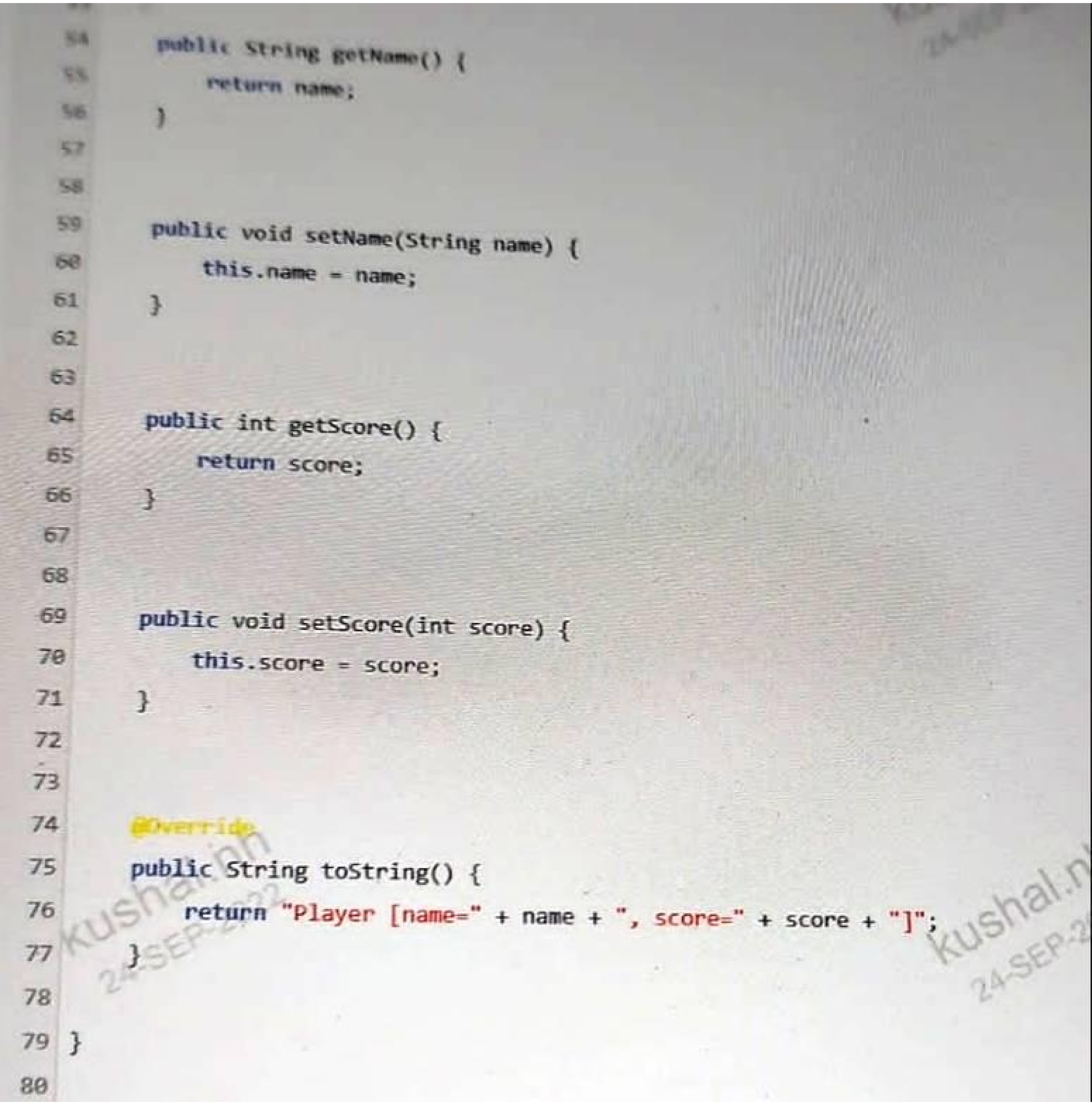


```


    }
}
class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
                 int n = scan.nextInt();

        /*Uncomment the for Loop once you have declared a List as 'players' with appropriate imports*/
        List<Player> players=new ArrayList<Player>();
       for(int i = 1; i <= n; i++){
            Player player = new Player(scan.next(), scan.nextInt());
            players.add(player);
        }


        /*Call the appropriate sorting Logic on the 'players' List here*/
        /*Uncomment the below for Loop once you have called the sorting Logic on the 'players' List*/

        players.sort(new Checker());
        for (Player player : players) {
            System.out.println(player.getName() +" "+player.getScore());
        }
        scan.close();

```



```
        scan.close();


    }
}
class Player {
    String name;
        int score;

    Player(String name, int score){
        this.name = name;
        this.score = score;
    }


    public String getName() {
        return name;
    }


    public void setName(String name) {
        this.name = name;
    }

```

Provide custom input



```
    public String getName() {
        return name;
    }


    public void setName(String name) {
        this.name = name;
    }


    public int getScore() {
        return score;
    }


    public void setScore(int score) {
        this.score = score;
    }


    @Override
    public String toString() {
        return "Player [name=" + name + ", score=" + score + "]";
    }

}
```