Java Method Overriding

When a subclass inherits from a superclass, it also inherits its methods; however, it can also override the superclass methods (as well as declare and implement new ones). Consider the following Sports class:

```
class Sports{
    String getName(){
        return "Generic Sports";
    }
    void getNumberOfTeamMembers(){
        System.out.println( "Each team has n players in " + getName() );
    }
}
```

Next, we create a Soccer class that inherits from the Sports class. We can override the getName method and return a different, subclass-specific string:

```
class Soccer extends Sports{
    @Override
    String getName(){
        return "Soccer Class";
    }
}
```

Note: When overriding a method, you should precede it with the `@Override` annotation. The parameter(s) and return type of an overridden method must be exactly the same as those of the method inherited from the supertype.

Task

Complete the code in your editor by writing an overridden getNumberOfTeamMembers method that prints the same statement as the superclass' getNumberOfTeamMembers method, except that it replaces n with 11 (the number of players on a Soccer team).

Output Format

When executed, your completed code should print the following:

```
Generic Sports
Each team has n players in Generic Sports
Soccer Class
```

Author	Shafaet
Difficulty	Easy
Max Score	10
Submitted By	206936

NEED HELP?

View discussions
View editorial
View top submissions

RATE THIS CHALLENGE

MORE DETAILS

Download problem statement
Download sample test cases
Suggest Edits

```
Each team has 11 players in Soccer Class
```

Change Theme Language Java 7

```
1  import java.util.*;
2  class Sports{
3
4      String getName(){
5          return "Generic Sports";
6      }
7
8      void getNumberOfTeamMembers(){
9          System.out.println( "Each team has n players in " +
   getName() );
10     }
11 }
12
13 class Soccer extends Sports{
14     @Override
15     String getName(){
16         return "Soccer Class";
17     }
18     @Override
19     void getNumberOfTeamMembers(){
20         System.out.println( "Each team has 11 players in " +
   getName() );
21     }
22
23     // Write your overridden getNumberOfTeamMembers method
   here
24
25 }
```

Line: 19 Col: 9

Upload Code as File Test against custom input Run Code Submit Code

You have earned 10.00 points!
You are now 30 points away from the 3rd star for your java badge.

0% 50/80

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0

Compiler Message

Success

Expected Output Download

```
1 Generic Sports
2 Each team has n players in Generic Sports
3 Soccer Class
4 Each team has 11 players in Soccer Class
```