

Revision Tue:

While defining the team of a player, we only registered the abbreviation of the team that the player plays for. Due to this issue you can show your viewers only a limited amount of information. For that, ESPN (the broadcasting company) contacted you and required a more detailed definition for a team that the player plays for. For that you will define a class named `Team` with the following specifications:

Instance Data Members:

- `name`: Full name of the team defined.
- `abbreviation`: Abbreviation of the team that has 3 characters (e.g., OKC, LAL, NYC, CLE).
- `stadiumName`: Name of the stadium where the team plays their home games.
- `playerCount`: Number of active players registered to the team (**Tip**: You may need to modify player registration from previous stages to implement this)
- `city`: City that the team plays for.

Methods:

- **Constructor**
 - Takes `name`, `abbreviation`, `stadiumName` and `city` as input values and initializes the respective instance data members for that instance.
 - Initializes `playerCount`.
- `transferPlayer`: Adds the player to the team. Changes `playerCount` accordingly.
- `releasePlayer`: Removes the player to the team. Changes `playerCount` accordingly.
- `changeStadium`: Changes the stadium name where the team plays its home games.
- `changeOwnership`: Changes the name of the team and the city that the team plays for. Takes **new name**, **abbreviation** and **city** as inputs.
- `toString`: Returns a string representation of the team showing details for all of the data members for that team.

After defining your class, you need to change the team variable in your `Player` class definition as well as the constructor and access methods, now that that variable is no longer a string but an instance of the `Team` class. Finally, you will reproduce the following output by calling the `toString` method of the `Team` from the player that plays for that team (`print` does this by default). To access the team instance you need to modify the accessor method for the team instance in `Player` class also. Keep in mind that you also need to maintain the number of players in the team now when you change the team of the player. A sample run is provided as follows.

Sample run:

```
Team team = new Team("Seattle Supersonics", "SEA", "Sonics Arena", "Seattle");
Player playerRev1 = new Player("Kevin Durant", "PF", team, 7);
team.changeOwnership("Oklahoma City Thunder", "OKC", "Oklahoma");
System.out.println(playerRev1.getTeam());
```

```
*****
Team name: Oklahoma City Thunder
Abbreviation: OKC
Number of players: 1
Stadium name: Chesapeake Energy Arena
City: Oklahoma
*****
```