

Solution Review 2: Implementing a Sports Team!

This lesson provides the solution to the challenge, "Implementing a Sports Team!" with an explanation.

We'll cover the following



- Solution

Solution

```
1 class Player:
2     def __init__(self, ID, name, teamName):
3         self.ID = ID
4         self.name = name
5         self.teamName = teamName
6
7
8 class Team:
9     def __init__(self, name):
10        self.name = name
11        self.players = []
12
13    def getNumberOfPlayers(self):
14        return len(self.players)
15
16    def addPlayer(self, player):
17        self.players.append(player)
18
19
20 class School:
21     def __init__(self, name):
22         self.name = name
23         self.teams = []
24
25    def addTeam(self, team):
26        self.teams.append(team)
27
28    def getTotalPlayersInSchool(self):
29        length = 0
30        for n in self.teams:
31            length = length + n.getNumberOfPlayers()
```



- **Line 2-5:** Defined initializer for `Player`.
- **Line 9-11:** Defined initializer for `Team` which also contains a list of `players`.
- **Line 13-14:** `getNumberOfPlayers()` returns the length of the list, `players`.
- **Line 16-17:** `addPlayer()` appends a `Player` object in the `players` list.
- **Line 21-23:** Defined initializer for `School` which also contains a list of `teams`.
- **Line 25-26:** `addTeam()` appends a `Team` object in the `players` list.
- **Line 28-32:** `getTotalPlayersInSchool` returns the total number of players in the school by adding the *length* of all the `players` list in the `teams` list.
- **Line 35-38:** Created four `Player` objects.
- **Line 40:** Created a `Team` object, `red_team`.
- **Line 41-42:** Added two `Player` objects in the `red_team`'s `players` list.
- **Line 44:** Created a `Team` object, `blue_team`.
- **Line 45-46:** Added two `Player` objects in the `blue_team`'s `players` list.
- **Line 48:** Created a `School` object, `mySchool`.
- **Line 49-50:** Added two `Team` objects in the `mySchool`'s `teams` list.