FAQ: Why do we use CricketCoach class instead of Coach Interface?

Question:

For the CricketCoach example with Setter Injection, why do we use the CricketCoach class instead of the Coach interface?

Answer:

The getTeam() method is only defined in the CricketCoach class. It is not part of the Coach interface.

As a result, you would need the following code:

```
CricketCoach theCricketCoach = context.getBean("myCricketCoach", CricketCoach.class);
```

The Coach interface has two methods: getDailyWorkout and getDailyFortune

The CricketCoach class has four methods: getDailyWorkout, getDailyFortune, getTeam and setTeam

---

When you retrieve a bean from the Spring container using the Coach interface:

```
Coach theCricketCoach = context.getBean("myCricketCoach", Coach.class);
```

You only have access to the methods defined in the Coach interface: getDailyWorkout and getDailyFortune. Even though the actual implementation has additional methods, you only have visibility to methods that are defined at the Coach interface level.

When you retrieve a bean from the Spring container using the CricketCoach class:

```
CricketCoach theCricketCoach = context.getBean("myCricketCoach", CricketCoach.class);
```

You have access to the methods defined in the Coach interface: getDailyWorkout and getDailyFortune.

ALSO, you have access to the additional methods defined in the CricketCoach class: getTeam, setTeam.

The bottom line is it depends on how you retrieve the object and assign it ... that determines the visibility you have to the methods.