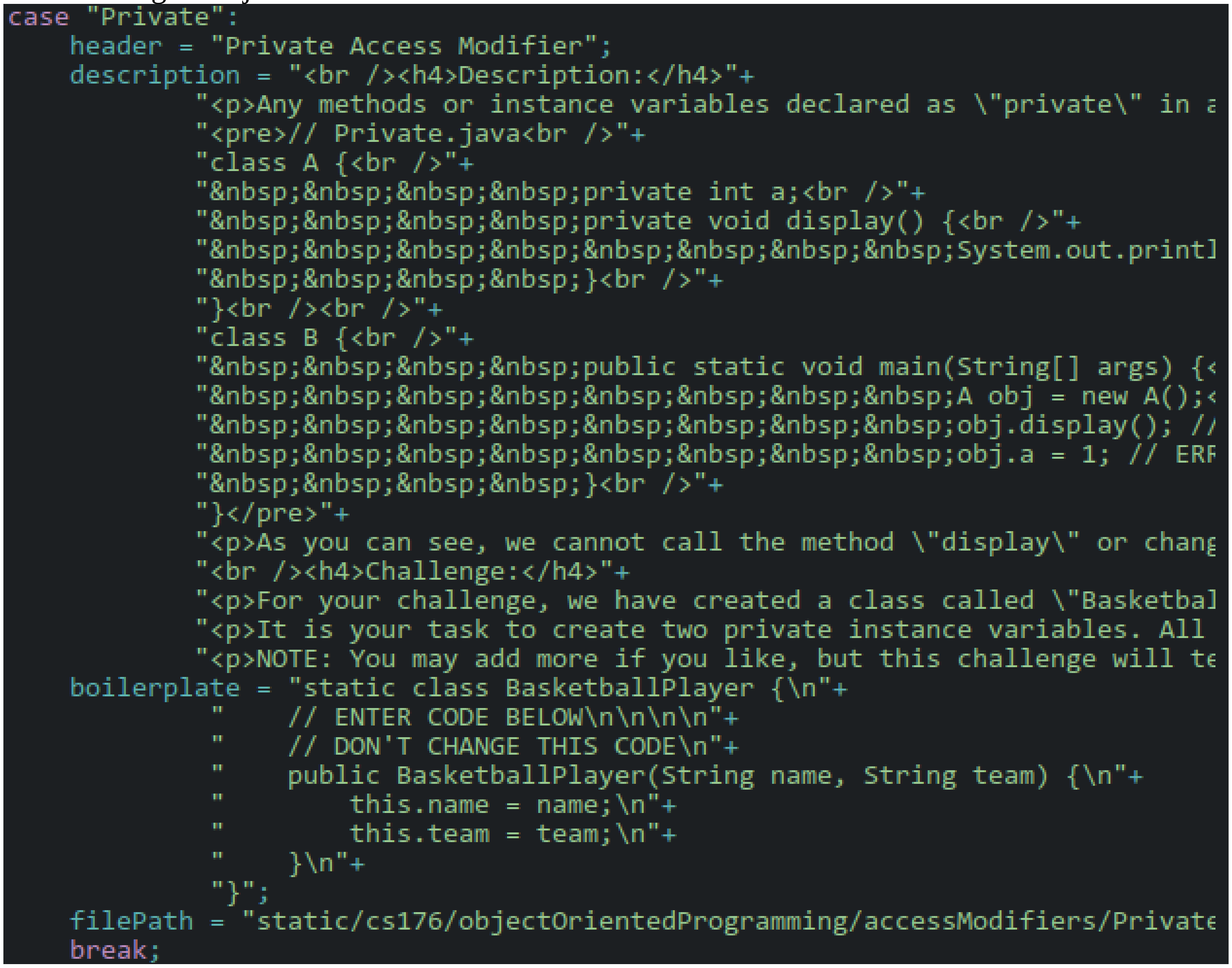
Adding a new challenge

Look at src/main/resources/static/(Course Name) to see how challenges are structured. You can first test your challenge in another project, and then copy/paste it into a new file under src/main/resources/static/(Course Name)/(Challenge Name).txt, make sure not to include the package declaration.

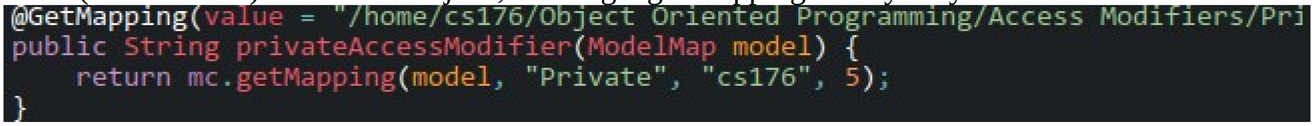
Inside (Course Name)ChallengeText.java, create a new case for your challenge name, and then make sure it has a header, description, and file path with a break statement at the end. Boilerplate is optional code that you want to show up in the text area when the user loads the challenge.
CS176ChallengeText.java:



```
case "Private":
    header = "Private Access Modifier";
    description = "<br /><h4>Description:</h4>"+
            "<p>Any methods or instance variables declared as \"private\" in a
            "<pre>// Private.java<br />"+
            "class A {<br />"+
            "&nbsp;&nbsp;&nbsp;&nbsp;private int a;<br />"+
            "&nbsp;&nbsp;&nbsp;&nbsp;private void display() {<br />"+
            "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;System.out.printl
            "&nbsp;&nbsp;&nbsp;&nbsp;}<br />"+
            "}<br /><br />"+
            "class B {<br />"+
            "&nbsp;&nbsp;&nbsp;&nbsp;public static void main(String[] args) {<
            "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;A obj = new A();<
            "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;obj.display(); //
            "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;obj.a = 1; // ERR
            "&nbsp;&nbsp;&nbsp;&nbsp;}<br />"+
            "}</pre>"+
            "<p>As you can see, we cannot call the method \"display\" or chang
            "<br /><h4>Challenge:</h4>"+
            "<p>For your challenge, we have created a class called \"Basketbal
            "<p>It is your task to create two private instance variables. All
            "<p>NOTE: You may add more if you like, but this challenge will te
    boilerplate = "static class BasketballPlayer {\n"+
             "    // ENTER CODE BELOW\n\n\n\n"+
             "    // DON'T CHANGE THIS CODE\n"+
             "    public BasketballPlayer(String name, String team) {\n"+
             "        this.name = name;\n"+
             "        this.team = team;\n"+
             "    }\n"+
             "}";
    filePath = "static/cs176/objectOrientedProgramming/accessModifiers/Private
    break;
```

Creating get and post mappings

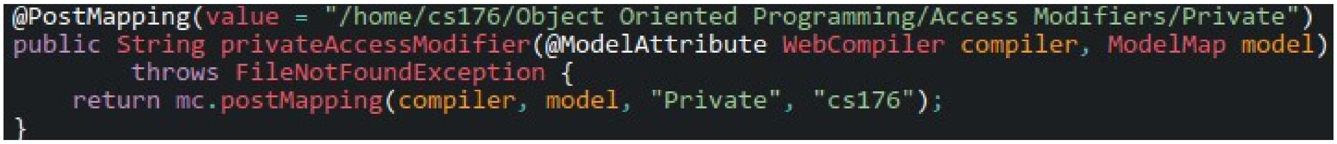
Inside (Cours Name)GetController.java, creating a get mapping is very easy:



```
@GetMapping(value = "/home/cs176/Object Oriented Programming/Access Modifiers/Pri
public String privateAccessModifier(ModelMap model) {
    return mc.getMapping(model, "Private", "cs176", 5);
}
```

You’re passing in a model, the challenge name, challenge course, and amount of points to award the user when completing the challenge.

Post mappings are a bit more work:



```
@PostMapping(value = "/home/cs176/Object Oriented Programming/Access Modifiers/Private")
public String privateAccessModifier(@ModelAttribute WebCompiler compiler, ModelMap model)
        throws FileNotFoundException {
    return mc.postMapping(compiler, model, "Private", "cs176");
}
```