# Instructions – A4 – Women's Soccer

**Overview:**
You'll be writing a python program that asks the user to enter a home women's soccer team and the number of games they'll play in a season. It will then ask about the teams they play against and randomly generate scores for games they play. It will then display information about the home team's wins, losses, and overall performance.

**Libraries Required:**
- import random

**Logical Flow:**

- Ask for the name of your home team (like "BYU")
- Then, ask for the number of games your home team will play in their season (Advice: I recommend keeping the number pretty low, like 2 games, when testing it so it is quicker to test).
- Then, for each game that your home team will play:
    - Ask the name of the away team (e.g. "Utah State") and include which number game this will be for. For example, for the first game it would look like:
        - "Enter the name of the away team for game 1: "
    - For the second game, it would look like:
        - Enter the name of the away team for game 2:
        - And so on
    - After entering in the away team name, randomly generate scores between 0 and 5 (inclusive) for the home team and the away team.
        - The easiest way to do this is by importing the random library. There are a few different methods you could use to do this (see Chapter 9.6 for reference)
        - In the case that there is a tie between the home and the away teams, keep generating new scores for the home and away teams until there isn't a tie.
    - If the home team has a higher score, keep track that they won a game.
        - Later in your code, you'll need to know the total number of games won and lost, so you can either also store the number lost, or just calculate losses by subtracting the number of wins from the total number or games.
    - If the home team won, store the name of the away team in a list of teams your home team won against. If the home team lost, store the name of the away team in a list of teams your home team lost against.
        - So, you'll have two separate lists. Just put the team name in one or the other list.
    - Print out the name of the home team's name and their score, as well as the away team's name and their score. If the away team's name were "UVU" it might look this this:
        - "BYU's score: 3 UVU's score: 1"
- Do this however many times the user inputted for the number of games the home team would play in the season. (E.g. if they inputted 3 for the number of games, go through the above logic 3 times). For full credit, you must use a loop to do this. A for loop is probably easier in this situation, but a while loop could work too.
- Once the season is over (all games played) display the following info:
    - Print out: "Teams won against:"

        - Then print out the name of each team with a tab before it in the list of teams that the home team won against. (see the example output to see how that would look)
    - Print out “Teams lost against:”
        - Then print out the name of each team with a tab before it in the list of teams that the home team lost against. (see the example output to see how that would look)
    - Print out “Final season record” followed by your home team name and their record, which is the number of wins, with a dash, then the number of losses. For example:
        - “Final season record: BYU 4-2”
    - After all of this, print out a final message based on the record of the home team.
        - If they won at least 75% of their games, then print out “Qualified for the NCAA Women's Soccer Tournament”.
        - If the team won at least 50% but less than 75% then print out “You had a good season”.
        - Otherwise print out “Your team needs to practice!”.

Upload just the python file to Learning Suite. This means you should upload the .py file that you made. You will lose points if you copy/paste your code directly to Learning Suite, or upload something like a word file instead of the .py file.

**Example Output:**
If you entered “BYU” for the team name and “4” for the number of games played, it might look something like this:

```
Enter the name of your team (the home team): BYU
Enter the number of teams that BYU will play (1-10): 4

Enter the name of the away team for game 1: UVU
BYU's score: 1. UVU's score: 5

Enter the name of the away team for game 2: Utah State
BYU's score: 1. Utah State's score: 0

Enter the name of the away team for game 3: University of Utah
BYU's score: 5. University of Utah's score: 4

Enter the name of the away team for game 4: SUU
BYU's score: 2. SUU's score: 5

Teams won against:
    Utah State
    University of Utah

Teams lost against:
    UVU
    SUU

Final season record: 2 - 2
You had a good season
```

Since you'll be generating random values for the scores, it may look different each time, even with the same inputs.

**Rubric**

| Requirement | Points | Notes |
|---|---|---|
| Asks user for home team name, number of games | 10 | · -5 for each missing |
| Loops the correct number of times based on input of user | 10 | · -5 if it doesn't loop the correct number of times<br>· -10 if they didn't use a loop |
| Asks the user for away team names | 5 | · -2 if it doesn't display the number of the game when asking for the user input |
| Generates random scores for each team between 1 and 5 | 10 | · -2 if it generates scores other than 1-5 |
| No tie scores allowed | 10 | · -7 if they attempt the logic, but do it incorrectly |
| Displays score of each game with the team names | 10 | · -5 if it doesn't include the team names.<br>· -5 if it doesn't display the scores |
| Displays names of teams home team won and lost against. | 15 | · -5 if they just directly print an entire list instead of each individual away team name<br>· -5 if they try to print the team names, but it prints the wrong team names under the "won against" or "lost against" sections.<br>· -1 if the away team names aren't tabbed in. |
| Displays final record of home team | 10 | · -4 if they didn't calculate the wins correctly<br>· -3 if they don't display the right number of losses |
| Displays final message based on the home team's record | 15 | · -2 if there is an error like using > instead of >=, etc.<br>· -7 if there is some other major error in the if statement logic so the correct statement doesn't print out. |
| includes comments: name and description at the top, and comments throughout | 3 | · -1 if it didn't include their name<br>· -1 if it didn't include a project description<br>· -2 if it didn't include general comments. Err on the side of leniency |
| file is turned in as a .py file | 2 | |
| | | |
| ***Total*** | 100 | |