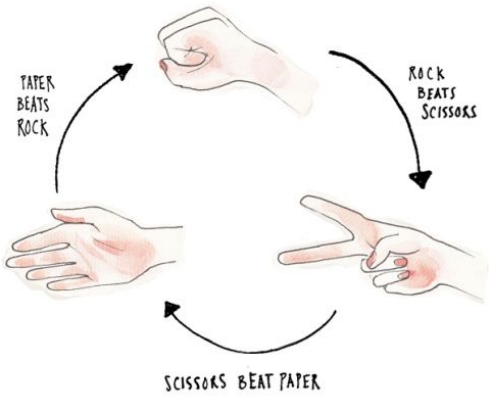
Assignment 8 (Ch15) - Not mandatory

Published | Edit | ⋮



https://i.pinimg.com/originals/c4/a3/02/c4a302fc7080660d3a1da435e1da5f47.jpg

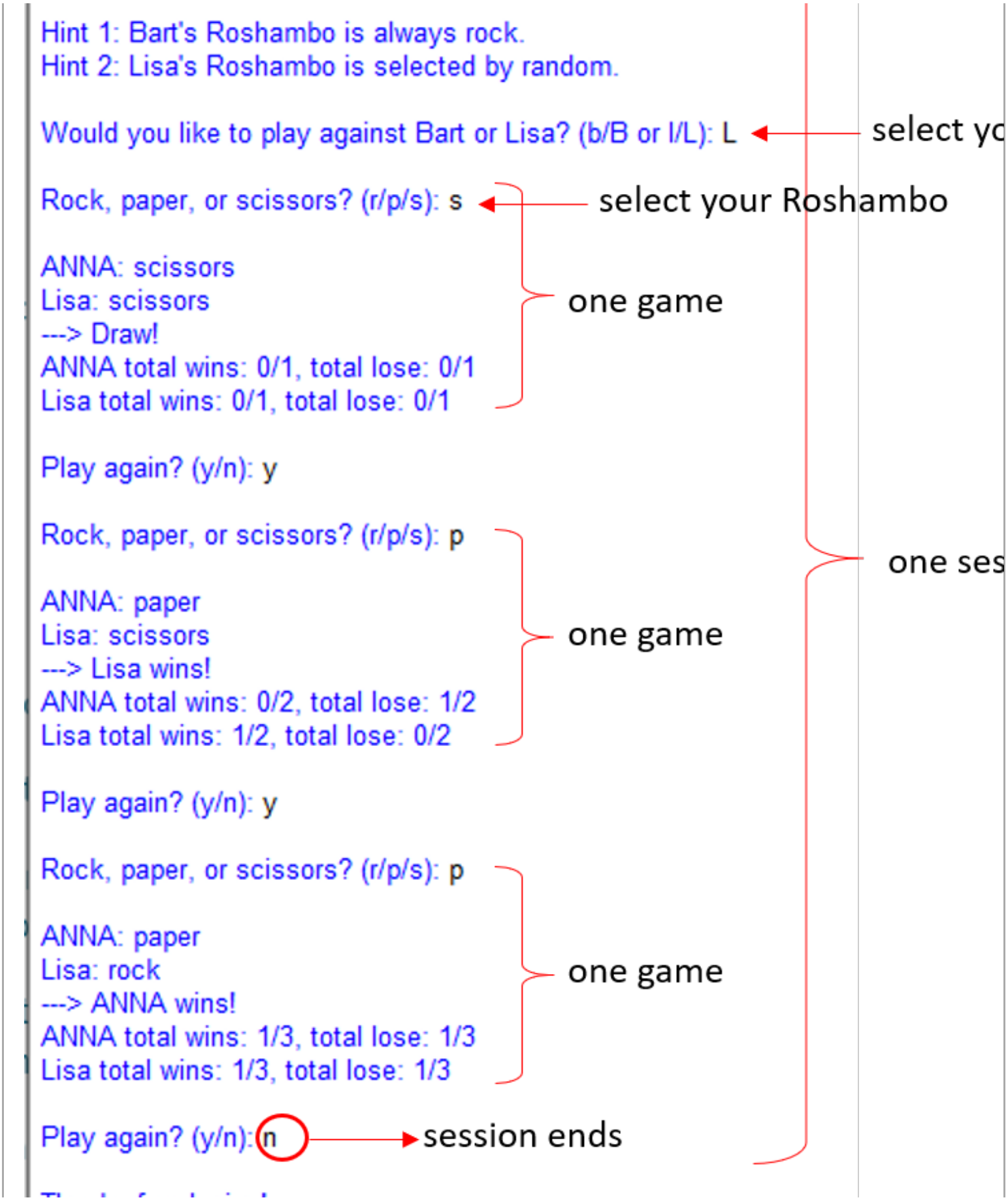
Roshambo (aka Rock-Paper-Scissors) is a traditional two-person hand game. See **here (http://playworks.org/game-library/ro-sham-bo-or-rock-paper-scissors/)** in case you don't know how to play it.

This (optional) assignment requires you to create an object-oriented Python module call **roshambo.py** for a Roshambo game where the user can choose to compete against one of two computer players: Bart or Lisa, where Bart's Roshambo is always rock and Lisa's Roshambo is selected by random. See below for a sample session of playing this game by executing **roshambo.py**

To reduce the complexity of object design, **roshambo.py** requires only the most basic part of Python's OOP with simple inheritance. No class attributes, no private or protected attributes/methods, no properties of any kind and no object compositions are invloved.

```
Roshambo Game

Enter your name: ANNA <-------- your name
```


Hint 1: Bart's Roshambo is always rock.
Hint 2: Lisa's Roshambo is selected by random.
Would you like to play against Bart or Lisa? (b/B or l/L): L
select yo
Rock, paper, or scissors? (r/p/s): s
select your Roshambo
ANNA: scissors
Lisa: scissors
---> Draw!
ANNA total wins: 0/1, total lose: 0/1
Lisa total wins: 0/1, total lose: 0/1
one game
Play again? (y/n): y
Rock, paper, or scissors? (r/p/s): p
ANNA: paper
Lisa: scissors
---> Lisa wins!
ANNA total wins: 0/2, total lose: 1/2
Lisa total wins: 1/2, total lose: 0/2
one game
Play again? (y/n): y
Rock, paper, or scissors? (r/p/s): p
ANNA: paper
Lisa: rock
---> ANNA wins!
ANNA total wins: 1/3, total lose: 1/3
Lisa total wins: 1/3, total lose: 1/3
one game
Play again? (y/n): n
session ends
one ses

```
Thanks for playing!
>>>
```

Specifications

1. Create a class named **Player** that provides public attributes for storing the player's name, Roshambo value, and the count of wins and losses of all games in a session. Here a game *session* means from the beginning of the game (i.e., program starts) until user answers 'n' to "Play again?". This class also overrides the **__str__** method of object class to show the player's name and his/her Roshambo value like "ANNA: paper" or "Lisa: rock" when the player object is printed during a game.
2. Create a class named **Bart** that inherits the Player class. It must initialize the player's name to "Bart" and add a public **generateRoshambo** method to set the Roshambo attribute to rock.
3. Create a class named **Lisa** that inherits the Player class. It must initialize the player's name to "Lisa" and adds a public **generateRoshambo** method to randomly select rock, paper, or scissors to set the Roshambo attribute.
4. The **main()** of this module should do several tasks in the following order:
   A. Display game title and get name of the player (i.e., you).
   B. Allow the player (i.e., you) to play Roshambo against Bart or Lisa.
   C. Create a Player object to represent you and an object of Bart or Lisa as your opponent.
   D. Counts the games played in a session.
   E. Allow you to choose your Roshambo value. In case of an invalid value (i.e., any characters other than "r", "s", and "p", in upper or lower case), it displays "Invalid choice. Try again." and allows the player (i.e., you) to re-enter. Note that invalid input is not demonstrated in the above same game.
   F. Let your opponent, Bart or Lisa, generate his/her Roshambo value.
   G. Display both players' name and roshambo.
   H. Call play() (see below) to display game result like "---> Lisa wins!" or "---> Draw!" and the updated wins/losses summary of both players.
   I. Query if the player (i.e., you) wants to play again. If yes, repeat the above tasks from D. Otherwise, display "Thanks for playing!" to end the execution.
5. The **play()** function accepts two player objects and game count as input parameters. It determines who wins the game or a player draws with the other and displays the result in a form like "---> Lisa wins!" or "---> Draw!". Based on the game result, this function also updates each player's attribute of count of wins and losses before their wins/losses summary is

displayed. Note that the total wins or losses is displayed in a form like "total wins: 1/3" to indicate the player wins one game out of three. Game draw is not included in the summary.

6. To simplify the code, no exception handling by using try-except is required.

When finished, please upload your **roshambo.py** here for a possible maximum of 40 (out of zero) points.

Points	0
Submitting	a file upload
File Types	py

Due	For	Available from	Until
Apr 15	Everyone	-	-

+ Rubric