06.2 - Rock Paper Scissors

Write a program that lets the user play a game of Rock, Paper, Scissors against the computer. The program should work as follows:

1. When the program begins, the computer chooses either 'rock', 'paper', or 'scissors' at random. (Don't display the computer's choice yet.) This step should be implemented using a function named `get_computer_choice` that takes no arguments and returns the computer's choice as a string, either 'rock', 'paper', or 'scissors'.

2. Then the user is asked to make their choice, either 'rock', 'paper', or 'scissors'. If the user enters an invalid choice they should be asked to try again until they enter a valid choice. This step should be implemented using a function named `get_player_choice` that takes no arguments and returns the player's valid choice as a string, either 'rock', 'paper', or 'scissors'.

3. Once both players have chosen, their choices are displayed.

4. Next a winner is selected according to the following rules:

   - If one player chooses rock and other player chooses scissors, then rock wins (rock smashes scissors).
   - If one player chooses scissors and the other player choose paper, then scissors wins (scissors cut paper).
   - If one player chooses paper and the other play chooses rock, then paper wins (paper wraps rock).
   - If both players make the same choice, the game is played again to determine the winner.

   This step should be implemented using a function named `get_winner` that takes the computer's choice as its first argument, and the player's choice as its second argument. Both arguments should be strings. The function should return the winner as a string, either 'computer', 'player', or 'tie'.

5. Finally, the program should display the results of the contest. If it is a tie, the match should continue until one player wins.

Test your program at least 3 times and take the screenshot of all the results. Include at least one invalid user choice, one rock, one paper, and one scissor in your test. A sample of the output is shown below. Your program should match the formatting of the sample but your results will be different because the computers choices should be generated randomly. User input in the sample has been highlighted in Pappy's Purple to distinguish it from the program's output, but your user input does not need to be colored. Save your program as `rock_paper_scissors_login.py`, where `login` is your Purdue login. Then submit it along with a screenshot showing **all 4** test conditions.

Terminal

```
$ python rock_paper_scissors_login.py
Choose rock, paper, or scissors: papper
You made an invalid choice. Please try again.
Choose rock, paper, or scissors: rock
  The computer chose paper, and you chose rock.
  paper beats rock
  You lost.  Better luck next time.
Thanks for playing.
$ python rock_paper_scissors_login.py
Choose rock, paper, or scissors: paper
  The computer chose rock, and you chose paper.
  paper beats rock
  You won the game!
Thanks for playing.
$ python rock_paper_scissors_login.py
Choose rock, paper, or scissors: scissors
  The computer chose scissors, and you chose scissors.
  It's a tie. Starting over.

Choose rock, paper, or scissors: rock
  The computer chose rock, and you chose rock.
  It's a tie. Starting over.

Choose rock, paper, or scissors: paper
  The computer chose scissors, and you chose paper.
  scissors beats paper
  You lost.  Better luck next time.
Thanks for playing.
```