CECS 277 – Lab 2 – Functions

Rock-Paper-Scissors

Write a program that allows a user to play Rock-Paper-Scissors against the computer. Have the program keep score of how many times each has won a round. Your program should have a main method that has a loop that repeats the game until the user chooses to quit and option 1 should also have a loop that repeats until the user chooses to go back to the main menu. Display the final score before exiting.

Write the following functions:

1. `weapon_menu()` - Prompts the user to input their choice: (R)ock, (P)aper, (S)cissors, or (B)ack. Checks that the user's input is an R, P, S, or B, displays the user's choice, and then returns the inputted value.
2. `comp_weapon()` - Randomly chooses the computer's throw, displays the computer's choice, and returns an R, P, or S.
3. `find_winner(player, comp)` – Passes in the player's and computer's weapon choices (R, P, or S), compares the two weapons, displays the result, and then returns who is the winner of that round (0=Tie, 1=Player, 2=Computer) based on the following rules.
    a. Rock crushes Scissors
    b. Scissors cuts Paper
    c. Paper covers Rock
4. `display_scores(p_score, c_score)` – Passes in the player and computer scores and displays them to the console.

Example Output (user input is in italics)**:**

```
RPS Menu:
1. Play game
2. Show Score
3. Quit
1
Choose your weapon:
R. Rock
P. Paper
S. Scissors
B. Back
P
You chose Paper
Computer chose Paper
Tie
Choose your weapon:
R. Rock
P. Paper
S. Scissors
B. Back
S
You chose Scissors
Computer chose Rock
Computer wins
Choose your weapon:
R. Rock
P. Paper
S. Scissors
B. Back
B
RPS Menu:
1. Play game
2. Show Score
3. Quit
2
Player = 0
Computer = 1
```

```
RPS Menu:
1. Play game
2. Show Score
3. Quit
```

```
3
Final Score:
Player = 0
Computer = 1
```

Notes:

1. Place your name, the date, and a brief description of the program in a comment block at the top of your program.
2. Use the check_input module provided on Canvas to check the user's input in the main menu (not the weapon menu).
3. Use the random module to randomly choose the computer's weapon.
4. Please do not use global variables. Pass values as parameters to your functions instead.
5. Do not create any extra functions or add any extra parameters.
6. Do not call your functions or main recursively.
7. Please read through the Coding Standards document provided on Canvas for guidance on how to name your variables and to format your program.
8. Use docstrings (triple-quote style) to document each of your four functions. Document all parameters and return values.
9. Add brief comments (# style) within your functions to describe sections of code.
10. Thoroughly test your program before submitting/demoing.
    a. Make sure your main code is in a main function.
    b. Make sure each of your functions passes in the correct parameters and returns the correct values.
    c. Make sure the computer's throw returns a random weapon choice (R, P, or S).
    d. Make sure that the correct winner is returned given the rules.
    e. Make sure that the points are awarded to the correct player after winning a round.
    f. Make sure that the points displayed are correct.
    g. Make sure that the game doesn't go back to the main menu until the user chooses to go back.
    h. Validate all user input (main menu is 1-3, weapon menu is R, P, S, B).

Rock Paper Scissors – Time estimate: 3 hours

Rock Paper Scissors **10 points**	Correct. 2 points	A minor mistake. 1.5 points	A few mistakes. 1 point	Several mistakes. 0.5 points	No attempt. 0 points
Weapon Choice Function: 1. Has weapon_menu function to get user's input. 2. Repeatedly checks for validity that the input is an R, P, S, or B. 3. Displays user’s weapon choice. 4. Returns the valid input.					
CompWeapon & Display Funcs: 1. Has display_scores function. 2. Passes in the player and computer scores and displays them. 3. Has comp_weapon function. 4. Randomly chooses the computer's weapon and displays it. 5. Returns either a R, P, or S.					
Find Winner Function: 1. Has find_winner function. 2. Passes in the player and computer weapon choices. 3. Uses if statements to compare weapons and display the results. 4. Returns the winner (0=Tie, 1=Player, 2=Computer).					
Main & Output: 1. Repeats main menu until user quits. 2. Has loop to repeat option 1. 3. Correctly calls functions. 4. Displays user & comp throws correctly. 5. Displays results & scores correctly. 6. Correctly validates all user input. 7. Not recursively called.					
Code Formatting: 1. Main code is in a main function. 2. Correct spacing. 3. Meaningful variable names. 4. No global variables. 5. Correctly documented.					