Homework 4: Rock, Paper, Scissors!

September 2018

Updated Sept 27 (Thurs) @ 11am.
We no longer ask you to use try/except statements, as they are superfluous for this exercise.
This homework is based entirely from content in Tuesday’s Lecture (Lecture #06)

You will be implementing the game Rock, Paper, Scissors! This activity will walk you through the process of breaking this program into concrete steps.

Download our hw04.py file, which you will fill in with actual code (and the answers to our questions).

The user will be playing a game of Rock, Paper, Scissors against the computer. In your homework, you will be writing a program that accepts user input and filters it for the user’s choice (as well as determining whether the user’s choice is valid), computes a random choice for the computer player, and then determines the winner.

1  Step 1: Overview (4 pts)

1. You will need to accept user input. What function are you going to use to do this?

2. How can you store the user’s input for later use throughout the program?

3. In other parts of the program, you will be simply printing text without requesting the user’s input. How do you do this?

Write your answers at the top of hw04.py.

2  Step 2: User’s Choice (5 pts)

Let’s require that the user inputs her choice correctly spelled, and the only thing we allow is for the text to be case-insensitive. For example, if the user types “rock” or “ROCK” or “Rock” or “RocK”, etc, our program should count this as a valid entry for “rock”. However, if the user inputs anything else like “Rock!” (with punctuation) or “rocks” or “a rock”, those are all invalid entries. Along the same lines for the other choices, “paper” and “PAPER” and “Paper” are all valid for representing “paper”, but “paper!” or “papers” are all invalid.

1. Write code the accepts the user’s input and tries to assign it to a variable. However, if the user inputs something that is not a valid option of rock, paper, or scissors, your program should politely (or ridiculously angrily) tell the user that their input needs to be “rock” “paper” or “scissors” then your program can end. If they’ve input valid input, your program should continue onto the next part of code in Step 3.
3 Step 3: Computer’s Choice (0 pts, we did this for you)

Generate a random response of paper, rock, or scissor for the computer. Specifically, we will use Python’s random module. To use Python’s random number module, we need to have the line:

```python
import random
```

At the beginning of our program (which we included for you in our provided hw04.py code)

To generate a random computer choice, use the code:

```python
random.choice(['rock', 'paper', 'scissors']).
```

This calls the `choice()` function within the random module, passing in a list (a data structure we will soon learn about, which is represented by the brackets) of the possible options. After you have generated a random choice for the computer and stored it in a variable, you will compare it with the user’s choice and determine a winner. You will then print out a message with the outcome.

4 Step 4: Control Flow (8 pts)

Once you generate the computer’s choice, you will need to determine the winner. Rock beats scissors, scissors beats paper, and paper beats rock. There are nine different possible outcomes, depending on the user’s and computer’s choice.

```python
if ____________:
    if ____________:
        elif ____________:
            elif ____________:
```

For example, you will start by checking if the word “rock” is in the user’s input. If the word is in there, you can use a nested if statement to further check if the computer chose “rock”. If so, you can determine that the game is a draw. You will also need to check the other computer choices of “paper” and “scissors” in nested `elif` statements. Display (print()) to the screen who the winner is, or if there’s a draw. That’s the end of the program, with no code following it.

5 Step 5: Improving Input Versatility (3 pts)

Now edit your program to allow the user a little more versatility. Namely, instead of only allowing input be case-insensitive, also accept input anytime one of the options is found anywhere within the word. For example, if the user inputs “a rock!” or “rock!” or “Rocks” we will accept all of these as valid input for “rock” This also has pitfalls, because we will accept words like “rocky” as being valid, but that’s okay.
1. Which built-in Python function can you use?

2. Write code to handle such.

NOTE: Don’t worry about crazy cases where the user may type something with multiple options in the same input, such as “rock paper!”.

6 Handin

Complete the questions which reside at the top of the hw04.py file in comments.
Submit your hw04.py to Canvas before October 1 (Monday) at 11:59PM.