

Homework #4 Introduction to Computing

Due: February 20, 2010 (Saturday at 11:59 PM)

Rock, Wall, Paper, Scissors Program

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. When two people play, each person pumps an arm up and down three times in unison while counting to three (“1”, “2”, “3”) out loud. On “3”, both players independently shape their hand to resemble either:



rock



paper



scissors

The winner is determined by the following rules:

“Rock smashes scissors”	If one player chooses rock and the other player chooses scissors, then the player with rock wins.
“Scissors cut paper”	If one player chooses scissors and the other player chooses paper, then the player with scissors wins.
“Paper covers rock”	If one player chooses paper and the other player chooses rock, then the player with paper wins.

If both players make the same choice, the process is repeated until there is a clear winner.

On the computer, we’ll need to substitute the text “rock”, “paper”, or “scissors” for the hand gestures. Your program should allow a human player to play rock-paper-scissors against the computer repeatedly. After each game, prompt the user with a question like: “Do you want to play again (Y/N)?”. When they ultimately quit, your program should output the following statistics:

- the number of times the human player won
- the number of times that the computer won
- the percentage of times that the computer won
- the average number of “gestures” needed to determine a winner overall

For each game the general algorithm you’ll need to follow is:

1. Have your program randomly chose one of the strings “rock”, “paper”, or “scissors” by using the “choice” function from the “random” module. At the top of your program, you’ll need to import the “random” module and using the choice function, Then, whenever you want to randomly select one of those string, call the choice function as:

```
computersChoice = random.choice(["rock", "paper", "scissors"])
```

Don’t display the computer’s choice yet.

2. Prompt the user to enter their choice of “rock”, “paper”, or “scissors” at the keyboard (use `input_raw`)

3. Display the computer’s choice and declare a winner based on the above rules.

ADDITIONAL REQUIREMENTS:

Before you start writing your program, (re)read section 6.2 on Problem Solving with Top-Down Design, because I want you to actually design your program. Split your program up into well-defined functions by doing a top-down design and developing a structure chart.

When you write your program, be sure to use:

- meaningful variable names with good style (i.e., useCamelCase or use_underscores)
- docstring comments at the start of the program **and** after the “def”inition of each function which describing what they do
- a main function (see lab4 Part C) located at the top of program with a call to it at the bottom of the file to start execution
- global constants where appropriate with good style (ALL_CAPS_AND_UNDERSCORES).

Save your program in a file called rockPaperScissors.py

Submit your homework electronically at http://www.cs.uni.edu/~schafer/submit/which_course.cgi

The steps for the homework submission system are:

1. Write, debug, and test your program. Save it in a file called rockPaperScissors.py
2. Log on to the submission system at: http://www.cs.uni.edu/~schafer/submit/which_course.cgi
(It is very likely that you will get some security certificate warnings when trying to use this. You may add an exception and accept the existing security certificate.) Use the same AD-ITS User name and password you use to log on the lab computers.
3. Select the course and section number of "810:051, Intro to Computer, Fienup". Click the "Continue".
4. Select the homework that you wish to submit: "HW 4: Rock, Paper, Scissors". Click "Continue"
5. Specify how many extra files you want to submit. Just leave it at 0. Click the "Continue" button.
6. Upload your program by Browsing and selecting your rockPaperScissors.py file. Click the "Continue" button.
7. The next page reports on the status of the upload(s). You can always continue to upload a better version of the program until the deadline. The newer file will replace an older file of the same name.