

Assignment 7 – Functions

Learning Objective

Define and call functions to simulate a game of rock-paper-scissors.

Assignment Description

Write a program that allows the user to play rock-paper-scissors against the computer. Your code will randomly choose an integer from 0 to 2 (inclusive), which will represent the computer's choice with 0 for rock, 1 for paper, and 2 for scissors. The user will enter an integer for their choice.

A winner is selected based on the following rules:

- Rock smashes scissors (If one player chooses rock and the other chooses scissors, then the player who chooses rock wins).
- Scissors cut paper (If one player chooses scissors and the other chooses paper, then the player who chooses scissors wins).
- Paper covers rock (If one player chooses paper and the other chooses rock, then the player who chooses paper wins).
- If both players make the same choice, then it is a tie.

The game continues as long as the player wants to play another round. When the player decides to exit the program, display the score results which includes how many times the player won, how many times the computer won, and the number of ties.

Steps

1. In PyCharm (Community Edition), open an existing project (such as ITP115) or create a new project.
 - If you open an existing project, then create a new directory (probably under the Assignments directory) named **a7_last_first** where *last* is your last/family name and *first* is your preferred first name. Use lowercase letters.
 - If you create a new project, then name it **a7_last_first** where *last* is your last/family name and *first* is your preferred first name. Use lowercase letters.

2. In the project or directory, create a new Python file called **assignment7.py**. At the top of the file, put comments in the following format and replace the name, email, and section with your actual information:

```
# Name, USC email
# ITP 115, Spring 2022
# Section: number or nickname
# Assignment 7
# Description:
# Describe what this program does.
```

3. Import the random module.
4. Define the **displayRules()** function.
 - o Parameters (input): None
 - o Return value (output): None
 - o Displays the game rules to the user:

```
Let's play Rock Paper Scissors.
The rules of the game are:
    Rock smashes Scissors
    Scissors cut Paper
    Paper covers Rock
    If both the choices are the same, it's a tie
```

5. Define the **getComputerNum()** function.
 - o Parameters (input): None
 - o Return value (output): an integer that is a random number between 0 and 2.
 - o Use the random module to get a random integer between 0 and 2.
6. Define the **getPlayerNum()** function.
 - o Parameters (input): None
 - o Return value (output): an integer that represents the user's choice
 - o Display the following message to the user:

```
Enter 0 for Rock, 1 for Paper, or 2 for Scissors
```

- o Get an integer from the user using "> " for the prompt. Make sure that user enters a 0, 1, or 2. If they don't, then keep looping until they enter a valid integer.

The graders will test by entering integers. You do not need to worry about the user entering a non-number.

```
> -1  
> 3  
> 1
```

7. Define the **playRound(computerNum, playerNum)** function.
- Parameter 1: an integer representing the computer's choice
 - Parameter 2: an integer representing the player's choice
 - Return value: an integer the represents if there was a tie or the winner:
 - Return -1 if the computer won the round
 - Return 1 if the player won the round
 - Return 0 if there is a tie
 - This method contains the game logic. It simulates the game and determines a winner. Use the logic in the Assignment Description above.

8. Define the **continueGame()** function.
- Parameters: None
 - Return value: a bool (Boolean), which is True or False (not a string)
 - Ask the user if they want to continue. Allow them to enter upper or lower case letters.

```
Do you want to continue playing (y or n)? y
```

- Return True if the user enters "y" or "Y". Otherwise, return False.
9. Define and call the **main()** function. This function will not have any parameters nor a return value.
- Create variables to keep track of the number of ties, wins by the player, and wins by the computer. You will update them in the while loop based on the return value of the **playRound()** function.
 - Create a while loop that runs as long as the user want to continue the game.
 - In the loop, display the menu, get the player's number, get the computer's number, and play a round by calling the appropriate functions (that you already

defined). Create variables for the return values and use them when calling the other functions.

- Since the `playRound()` function will return the result of who won the game, update the appropriate variable and print who won or if there was a tie. Display the appropriate message:

| |
|-----------------------------------|
| You win! |
| Computer wins. |
| You and the computer tied. |

- Call the `continueGame()` function to ask the user if they want to continue, and use their response to control the while loop.
- After the while loop, display the final results (i.e. number of ties, number of player wins, and number of computer wins).

| |
|---|
| You won 2 game(s). |
| The computer won 1 game(s). |
| You tied with the computer 0 times(s). |

10. Be sure to comment your code. This means that there should be comments throughout your code. Put a comment block before each function stating the parameters, return values, and what that function does. Points will be deducted for not having comments.
11. Follow coding conventions. You should use lowerCamelCase or snake_case for variable names. You are welcome to create any variables that you need.
12. Test the program. Look at the Sample Output below. Assignments that do not run are subject to 20% penalty.
13. Prepare your submission:
 - Find the **a7_last_first** folder on your computer and compress it. This cannot be done within PyCharm.
 - On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
 - On Mac OS, use **Finder** to select the folder. Right click and select the Compress "FolderName" option. This will create a zip file.
14. Upload the zip file to your Blackboard section:

- On Blackboard, navigate to the appropriate item.
- Click on the specific item for this assignment.
- Click on the **Browse Local Files** button and select the zip file.
- Click the **Submit** button.

Grading

- This assignment is worth 35 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.

| Item | Points |
|------------------|-----------|
| displayRules() | 2 |
| getComputerNum() | 3 |
| getPlayerNum() | 5 |
| playRound() | 10 |
| continueGame() | 5 |
| main() | 10 |
| Total | 35 |

Sample Output

```
Let's play Rock Paper Scissors.
The rules of the game are:
    Rock smashes Scissors
    Scissors cut Paper
    Paper covers Rock
    If both the choices are the same, it's a tie
Enter 0 for Rock, 1 for Paper, or 2 for scissors
> 2
You win!
Do you want to continue playing (y or n)? Y
```

```
Let's play Rock Paper Scissors.
The rules of the game are:
    Rock smashes Scissors
    Scissors cut Paper
    Paper covers Rock
    If both the choices are the same, it's a tie
Enter 0 for Rock, 1 for Paper, or 2 for scissors
> -1
> 3
> 1
Computer wins.
Do you want to continue playing (y or n)? y
```

```
Let's play Rock Paper Scissors.
The rules of the game are:
    Rock smashes Scissors
    Scissors cut Paper
    Paper covers Rock
    If both the choices are the same, it's a tie
Enter 0 for Rock, 1 for Paper, or 2 for scissors
> 0
You win!
Do you want to continue playing (y or n)? Q
```

```
You won 2 game(s).
The computer won 1 game(s).
You tied with the computer 0 times(s).
```