SFWRTECH 3PR3:

Procedural and Objective Oriented Programming Concepts

(Assignment #1)

Student Name: Dojae Kim

Student Number: 400420323

Professor Name: Dr. Seshasai Srinivasan

**Objective**

The purpose of this Assignment 1:

1. To give students practice writing python code.

2. To learn how to use the case statement (if-elif-else) to solve given conditions.

3. To understand how to import module from python library.

**Introduction**

Rock, Paper, Scissors, Spock, Lizard is the upgraded version of Rock, Paper, Scissors. This game that is generally played by two people and the main idea is to make hand shapes that represent a rock, paper, scissors, spock, and lizard. The program will produce a random outcome. Based on each player’s outcome that would determine who is the winner.

\* Basic rule table of Rock Paper Scissors Spock Lizard

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Hand Shape** | **Beat** | **Lose** | **Tie** |
| 1 | Rock | Scissors, Lizard | Paper, Spock | Rock |
| 2 | Paper | Rock, Spock | Scissors, Lizard | Paper |
| 3 | Scissors | Paper, Lizard | Rock, Spock | Scissors |
| 4 | Spock | Scissors, Rock | Paper, Lizard | Spock |
| 5 | Lizard | Spock, Paper | Rock, Scissors | Lizard |

**Input Specification**

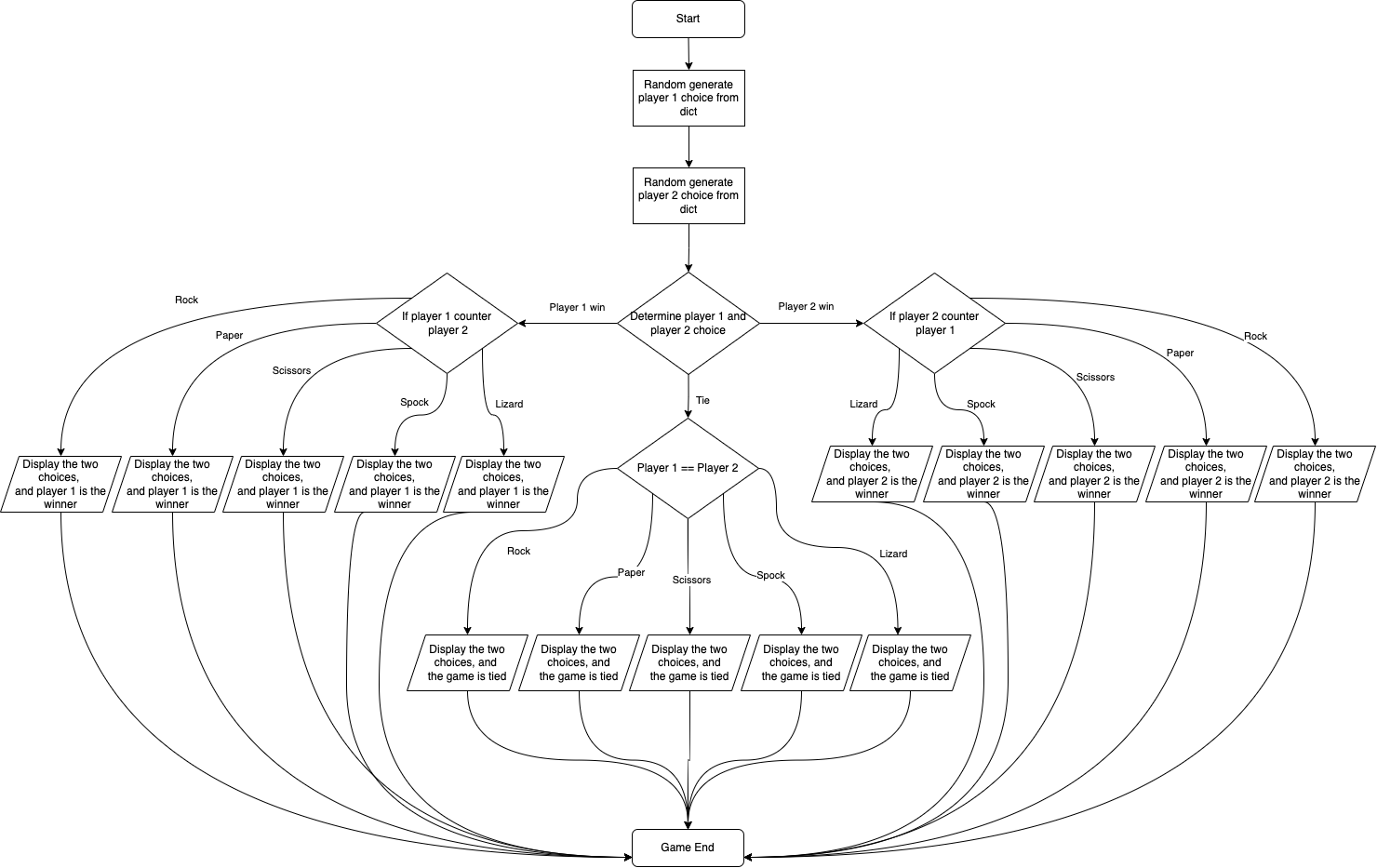
What are the shapes of Rock, Paper, Scissors, Spock, and Lizard?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shapes of Rock Paper Scissors Spock Lizard** | | | | |
|  |  |  |  |  |
| Rock | Paper | Scissors | Spock | Lizard |

**Output Specification (Expected)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Player 1** | **Player 2** | **Result (Output)** |
| 1 | Rock | Rock | Tie |
| 2 | Rock | Paper | Player 2 Wins |
| 3 | Rock | Scissors | Player 1 Wins |
| 4 | Rock | Spock | Player 2 Wins |
| 5 | Rock | Lizard | Player 1 Wins |
| 6 | Paper | Rock | Player 1 Wins |
| 7 | Paper | Paper | Tie |
| 8 | Paper | Scissors | Player 2 Wins |
| 9 | Paper | Spock | Player 1 Wins |
| 10 | Paper | Lizard | Player 2 Wins |
| 11 | Scissors | Rock | Player 2 Wins |
| 12 | Scissors | Paper | Player 1 Wins |
| 13 | Scissors | Scissors | Tie |
| 14 | Scissors | Spock | Player 2 Wins |
| 15 | Scissors | Lizard | Player 1 Wins |
| 16 | Spock | Rock | Player 1 Wins |
| 17 | Spock | Paper | Player 2 Wins |
| 18 | Spock | Scissors | Player 1 Wins |
| 19 | Spock | Spock | Tie |
| 20 | Spock | Lizard | Player 2 Wins |
| 21 | Lizard | Rock | Player 2 Wins |
| 22 | Lizard | Paper | Player 1 Wins |
| 23 | Lizard | Scissors | Player 2 Wins |
| 24 | Lizard | Spock | Player 1 Wins |
| 25 | Lizard | Lizard | Tie |

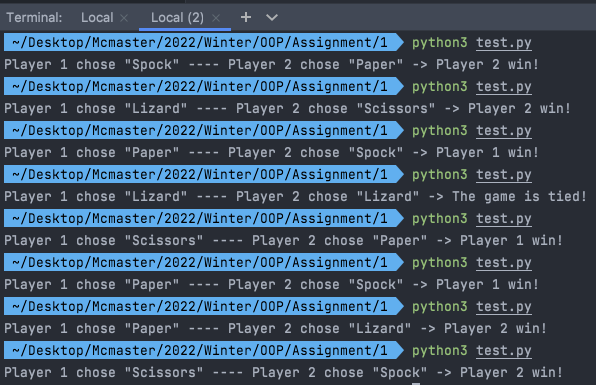
**Flow Chart**



**Source Code**

# Student name: Dojae Kim  
# Student number: 400420323  
# Student email: kim408@mcmaster.ca  
# Lecture: SFWRTECH 3PR3  
# Assignment1  
  
"""  
Rock Paper Scissors Spock Lizard game. The program will have two players and each of them produce a random outcome,  
i.e., Rock, paper, scissor, spock, or lizard. Based on each player’s outcome, determine the winner.  
The rules of the game are:  
 Spock beats scissors and rock, but loses to paper and lizard.  
 Lizard beats Spock and paper, but loses to rock and scissors.  
 Rock beats scissors and lizard, but loses to paper and Spock.  
 Paper beats rock and Spock, but loses to scissors and lizard.  
 Scissors beats paper and lizard, but loses to rock and Spock.  
"""  
  
import random  
  
hand\_shapes = {1: "Rock", 2: "Paper", 3: "Scissors", 4: "Spock", 5: "Lizard"}  
  
# The hand shapes will be randomly generated  
player\_1 = random.randint(1, 5)  
player\_2 = random.randint(1, 5)  
  
# If both players choose the same hand shape, the game is tied  
if player\_1 == player\_2:  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> The game is tied!')  
  
elif (player\_1 == 1 and player\_2 == 3) or (player\_1 == 1 and player\_2 == 5):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 1 win!')  
  
elif (player\_1 == 3 and player\_2 == 1) or (player\_1 == 5 and player\_2 == 1):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 2 win!')  
  
elif (player\_1 == 2 and player\_2 == 1) or (player\_1 == 2 and player\_2 == 4):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 1 win!')  
  
elif (player\_1 == 1 and player\_2 == 2) or (player\_1 == 4 and player\_2 == 2):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 2 win!')  
  
elif (player\_1 == 3 and player\_2 == 2) or (player\_1 == 3 and player\_2 == 5):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 1 win!')  
  
elif (player\_1 == 2 and player\_2 == 3) or (player\_1 == 5 and player\_2 == 3):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 2 win!')  
  
elif (player\_1 == 4 and player\_2 == 3) or (player\_1 == 4 and player\_2 == 1):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 1 win!')  
  
elif (player\_1 == 3 and player\_2 == 4) or (player\_1 == 1 and player\_2 == 4):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 2 win!')  
  
elif (player\_1 == 5 and player\_2 == 4) or (player\_1 == 5 and player\_2 == 2):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 1 win!')  
  
elif (player\_1 == 4 and player\_2 == 5) or (player\_1 == 2 and player\_2 == 5):  
 print(f'Player 1 chose "{hand\_shapes[player\_1]}" ---- Player 2 chose "{hand\_shapes[player\_2]}" -> Player 2 win!')

**Sample Output**



**Conclusion**

Through this assignment1, I learned that there are many types of python operators and conditional statements. I was also able to learn how to import random module to generate random choices from dictionary. As a tool, it is extremely important to have that knowledge and understanding how to build up the case statement for every programmer.