

Self-Learning Project Report

PROGRAMMING(PYTHON)

Group Leader Name: Abdulrahman Almyman

University ID:44

Section Number: 72005

Teacher Name: Mohammed ashrafuddin

Date:10/11/2019 1441/3/13

No.	Name	University Id
1	Majeed aldakheelallah	44
2	Abdulmajeed al shareif	44
3	Khalid alshiluwyy	44

ACKNOWLEDGEMENT

*we would like to thanks our teacher Mr. Mohammed Ashrafuddin
for his guidance and support during our project work.*

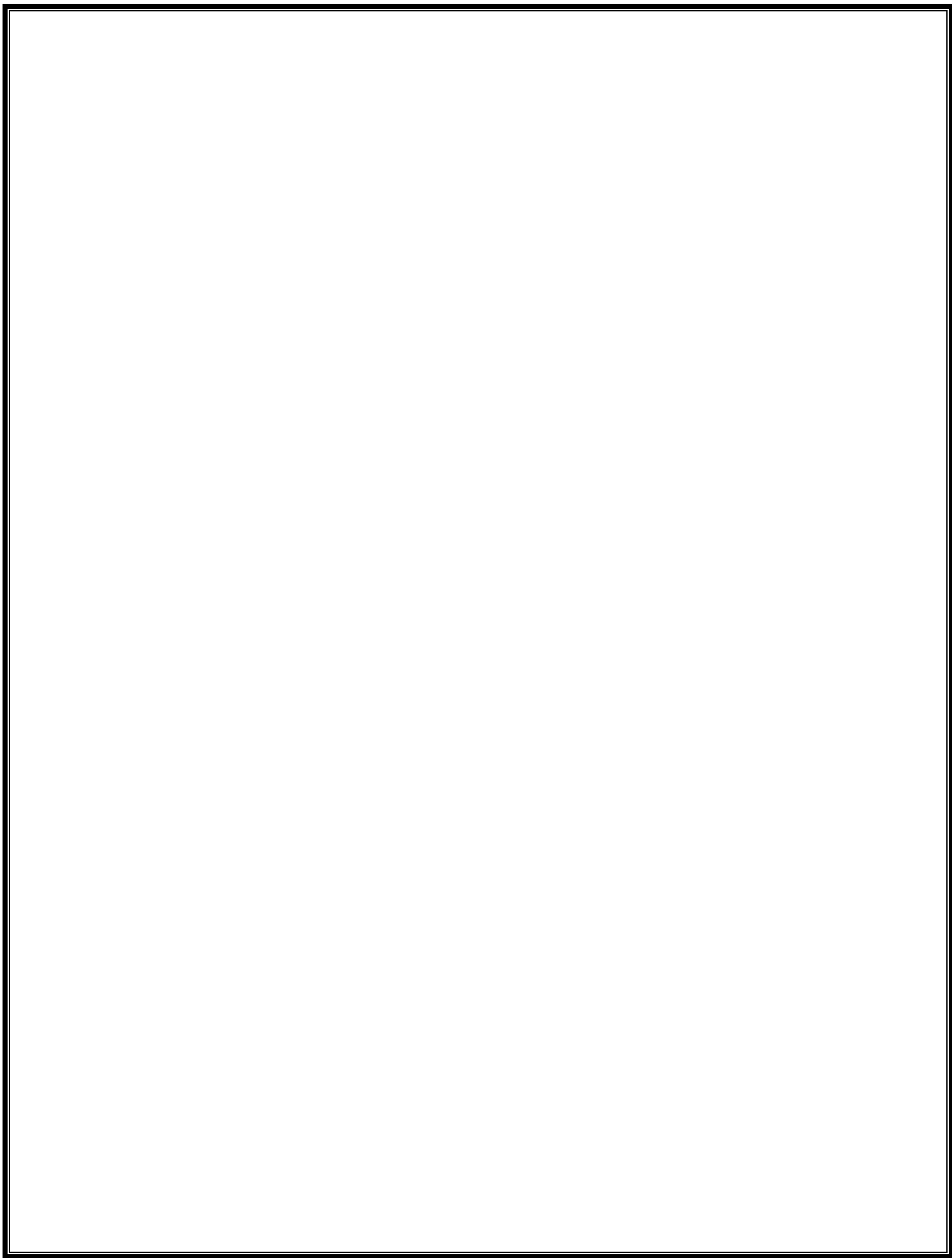
ABSTRACT

This is our self-learning project we selected the topic “Programming(Python)”. We are going to show “Rock-Paper-scissor” game in this project by using IDLE(Python).

Table of Contents

Contents

INTRODUCTION	6
Algorithm	6
Flowchart	6
Programming Languages.....	6
Features of Programming Languages.....	7
Simplicity.....	7
Naturalness.....	7
Efficiency	7
Compactness	7
Locality	7
Procedure	8
Step 1	8
Algorithm	8
Step 2	9
Flowchart	9
Step 3	10
Program in Python	10
Step 4: Result - Output of the Program.....	11
References	13

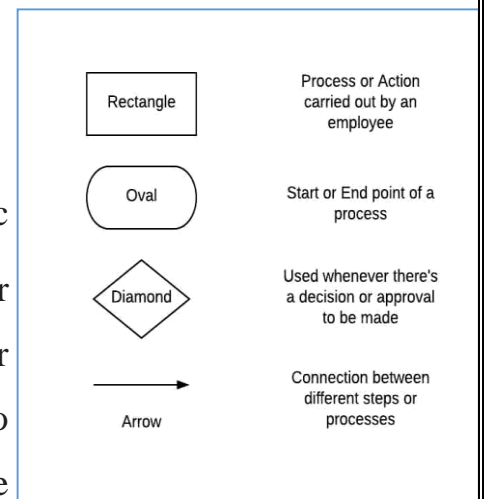


INTRODUCTION:

Algorithm: An algorithm is a step by step method of solving a problem.

It is commonly used for data processing, calculation and other related computer and mathematical operations. An algorithm is also used to manipulate data in various ways, such as inserting a new data item, searching for a particular item or sorting an item.

Flowchart: A flowchart is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure. The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.



Programming Languages: A programming language is a formal language, which comprises a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms.

Features of Programming Languages

Simplicity: A good programming language must be simple and easy to learn and use. It should provide a programmer with a clear, simple and unified set of concepts, which can be easily grasped.

Naturalness: A good language should be natural for the application area, for which it has been designed.

Efficiency: Programs written in a good programming language are efficiently translated into machine code, are efficiently executed, and acquire as little space in the memory as possible.

Compactness: In a good programming language, programmers should be able to express intended operations concisely.

Locality: A good programming language should be such that while writing a programmer concentrate almost solely on the part of the program around the statement currently being worked with.

Procedure

Step 1: Algorithm:

Step1: start.

Step2: Declare variables and constants

Step3: Input player-choice.

Step4: comp_choice \leftarrow random (1,3)

Step5: decision, if player_choice= comp_choice

Step6: if yes then output "draw" then stop.

Step7: if no then player_choice=1?

step7.1: if yes comp_choice = 2?

step7.2: if yes winner \square "player" then winner=" player"?

step7.3: if no winner \square "comp"

step8: if player_choice = 2?

step8.1: if yes then comp_choice=3?

step8.2: if yes winner \square "player" then winner="player"?

step8.3: if no winner \square "comp"

step9: if no then player_choice = 1?

step9.1: if yes winner \square "player" then winner=" player"?

step9.2: if no winner \square "comp"

step10: if winner =" player"?

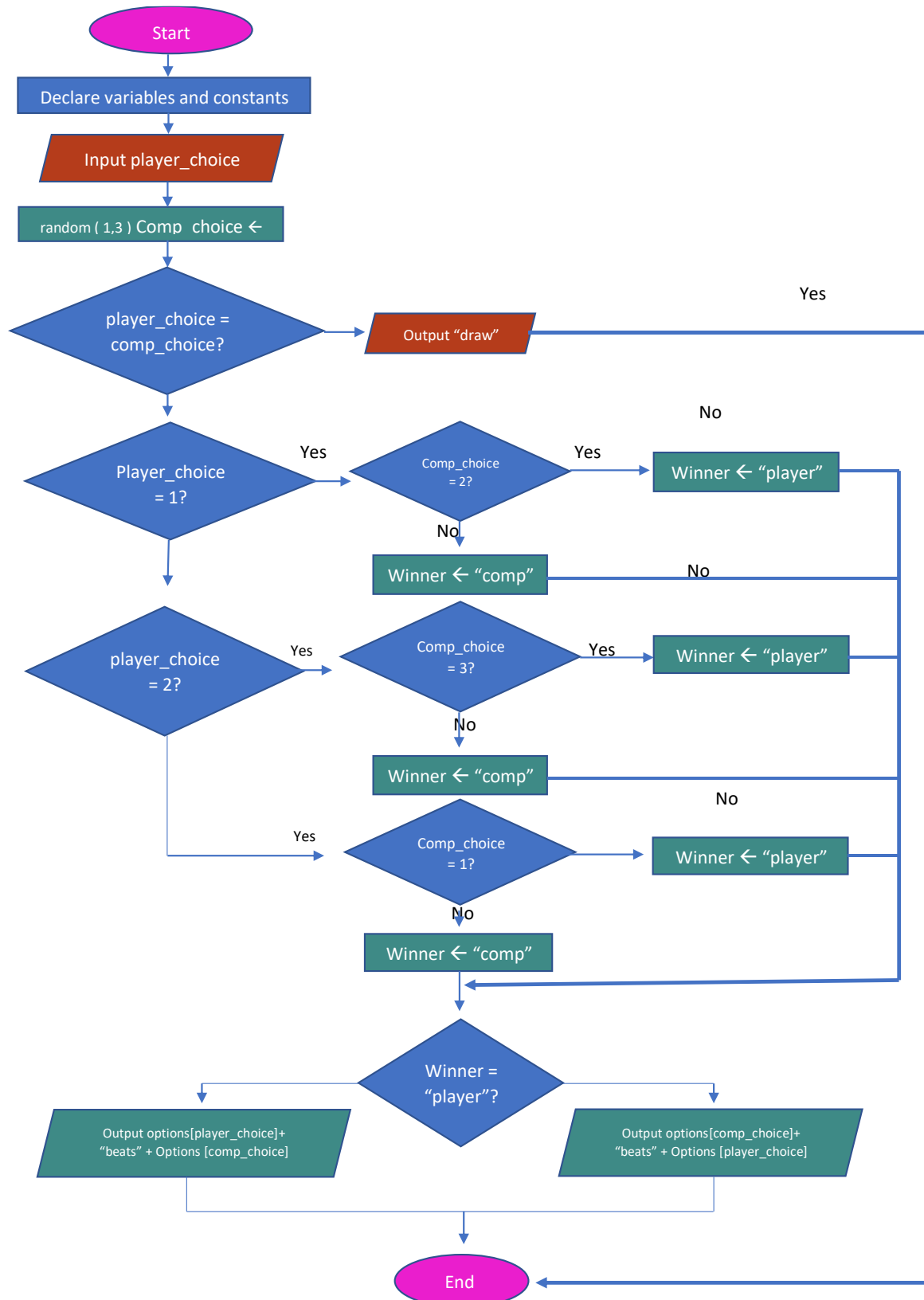
**step10.1: if no then output options [comp_choice] +" beats"+options
[player_choice] then end.**

**Step10.2: if if yes then output options [player_choice] +"beats"+options
[comp_choice] then end then end.**

Step11: stop

Step 2: Draw a flowchart for the selected algorithm.

Flowchart:



Step 3: Write a program to get desired output based on algorithm and flowchart created.

Program in Python:

rps.py - C:\Users\TEMP\Desktop\Project-Python\rps.py (3.6.5)

File Edit Format Run Options Window Help

```
import random
print("Winning Rules of the Rock paper scissor game as follows: \n"
      +"Rock vs paper->paper wins \n"
      + "Rock vs scissor->Rock wins \n"
      +"paper vs scissor->scissor wins \n")

while True:
    print("Enter choice \n 1. Rock \n 2. paper \n 3. scissor \n")
    choice = int(input("User turn: "))
    while choice > 3 or choice < 1:
        choice = int(input("enter valid input: "))
    if choice == 1:
        choice_name = 'Rock'
    elif choice == 2:
        choice_name = 'paper'
    else:
        choice_name = 'scissor'

    print("user choice is: " + choice_name)
    print("\nNow its computer turn.....")
    comp_choice = random.randint(1, 3)
    while comp_choice == choice:
        comp_choice = random.randint(1, 3)
    if comp_choice == 1:
        comp_choice_name = 'Rock'
    elif comp_choice == 2:
        comp_choice_name = 'paper'
    else:
        comp_choice_name = 'scissor'

    print("Computer choice is: " + comp_choice_name)

    print(choice_name + " V/s " + comp_choice_name)

    if((choice == 1 and comp_choice == 2) or
        (choice == 2 and comp_choice == 1)):
        print("paper wins => ", end = "")
        result = "paper"

    elif((choice == 1 and comp_choice == 3) or
        (choice == 3 and comp_choice == 1)):
        print("Rock wins =>", end = "")
        result = "Rock"
    else:
        print("scissor wins =>", end = "")
        result = "scissor"

    if result == choice_name:
        print("<== User wins ==>")
    else:
        print("<== Computer wins ==>")

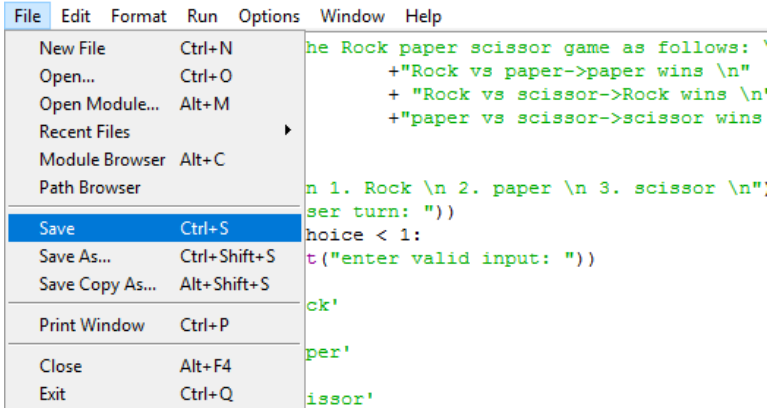
    print("Do you want to play again? (Y/N)")
    ans = input()

    if ans == 'n' or ans == 'N':
        break
print("\nThanks for playing")
```

Step 4: Result - Output of the Program.

Save and Run the Program:

rps.py - C:\Users\TEMP\Desktop\Project-Python\rps.py (3.6.5)



```
the Rock paper scissor game as follows: `
    + "Rock vs paper->paper wins \n"
    + "Rock vs scissor->Rock wins \n"
    + "paper vs scissor->scissor wins

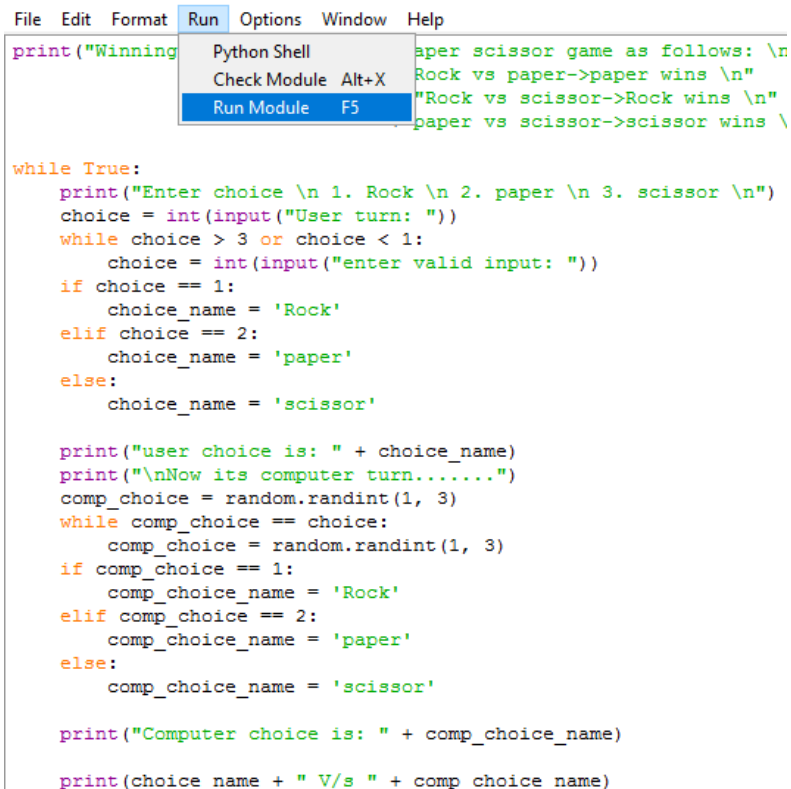
n 1. Rock \n 2. paper \n 3. scissor \n";
ser turn: ")
choice < 1:
t("enter valid input: ")

ck'
per'
issor'
```


```
print("user choice is: " + choice_name)
print("\nNow its computer turn.....")
comp_choice = random.randint(1, 3)
while comp_choice == choice:
    comp_choice = random.randint(1, 3)
if comp_choice == 1:
    comp_choice_name = 'Rock'
elif comp_choice == 2:
    comp_choice_name = 'paper'
else:
    comp_choice_name = 'scissor'

print("Computer choice is: " + comp_choice_name)
```

rps.py - C:\Users\TEMP\Desktop\Project-Python\rps.py (3.6.5)



Play with computer and choose number:

 *Python 3.6.5 Shell*

File Edit Shell Debug Options Window Help

Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:\Users\TEMP\Desktop\Project-Python\rps.py =====

Winning Rules of the Rock paper scissor game as follows:

Rock vs paper->paper wins

Rock vs scissor->Rock wins


paper vs scissor->scissor wins

Enter choice

1. Rock
2. paper
3. scissor

User turn: |

Select Operation to get the desired output:

 *Python 3.6.5 Shell*

File Edit Shell Debug Options Window Help

Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.

>>>

===== RESTART: C:\Users\TEMP\Desktop\Project-Python\rps.py =====

Winning Rules of the Rock paper scissor game as follows:

Rock vs paper->paper wins

Rock vs scissor->Rock wins

paper vs scissor->scissor wins

Enter choice

1. Rock
2. paper
3. scissor

User turn: 2

user choice is: paper

Now its computer turn.....

Computer choice is: scissor

paper V/s scissor

scissor wins ==><== Computer wins ==>

Do you want to play again? (Y/N)

y

Enter choice

1. Rock
2. paper
3. scissor

User turn: 3

user choice is: scissor

Now its computer turn.....

Computer choice is: paper

scissor V/s paper

scissor wins ==><== User wins ==>

Do you want to play again? (Y/N)

|

Result:

Using Python's integrated Development and learning Environment IDLE(Python) was very good experience. We learned a lot by using this such as how to write program in shell and code window and debug and modify our code to error free our project.

References:

<https://www.bbc.co.uk/bitesize/guides/zjqmsg/revision/4>

<https://thehelloworldprogram.com/python/python-game-rock-paper-scissors/>

<https://www.geeksforgeeks.org/python-program-implement-rock-paper-scissor-game/>