Network Plus Training Center

Python for Network Engineers

Lab Manual

Developed By: Waleed Adlan – CCIE #41999/HCIE#14518

Lab 1

Basic Functions Data Types & Operators

Task 1:
Make a python program to write 'Hello World' Both as Script and in interactive mode

For the interactive mode

print('Hello World')

For the script

Open a file using IDLE, then write

print ('Hello World') then save the file, for example name it 1stprogram.py

Then from the command prompt, move to the same directory where you save the file

Cd /File_path → File Path is the directory where you saved the file

Then issue the following command

Python 1stprogram.py

Task 2:
Make the following math operation in Interactive mode

- Addition 5+10
- Subtraction 12-10
- Multiplication 3*12

- Division-Float 20/3
- Division-Integer 20//3
- Division-Remainder 20%3
- Combining Operation 12*(5+1)

5+10

12-10

3*12

20/3

20//3

20%3

Task 3:
Use Interactive mode to assign a value string variable using single quotes and print that variable. Then repeat it again using double quotes for another variable and print it.

X = ‘Network Plus’
print(X)

Y = “International Exam”
print(Y)

Task 4:
Use print function to print Network’s Courses

print (“Network’s Courses”)

Or

print (‘Network\’s Courses’)

Task 5:
Use print function to print the following maintaining separate line for each sentence as shown below:
Our network consists of:
Three Routers
One Firewall
Ten Switches

print(‘Our network Consists of:\nThree Routers \nOne Firewall \nTen Switches’)

Or

print(“””Our network consists of:
Three Routers
One Firewall
Ten Switches”””)

Task 6:
Use print function with r option to print the following:
NetworkPlus \n Classes

print(r”NetworkPlus\n Classes”)

Task 7:
Use print function and Multiplication print ‘Network Plus’ five times

print (5 * ‘Network Plus ’)

Task 8:

print ‘Network Plus Building’ by making a string variable has the value of ‘Network Plus’, Then make another string variable its value is the first value concatenated with ‘Building’ String. Then print the second variable

```
X= ‘Network Plus’
Y= X + ‘Building’
print(Y)
```

Task 9:
Use Input Function to allow the user receiving a message of “Please Enter your Input”, then print out whatever the user has entered

```
X=Input(‘Please Enter your Input’)
print(X)
```

Task 10:
Create a list named Router_IP_Addresses which is having three string elements: 192.168.1.1,192.168.2.1,192.168.3.1. Then print the whole list using print function

```
Router_IP_Addresses = [‘192.168.1.1’,’192.168.2.1’,’192.168.3.1’]
print(Router_IP_Addresses)
```

Task 11:
In the previously created list, print the following
‘The first element in this list is’ , then print the first element

```
print(‘The first element in this list is ’,Router_IP_Addresses[0])
```

Task 12:
Append a new element to the list, which is having a value of ‘192.168.4.1’, Then reprint the list

```
Router_IP_Addresses.append(‘192.168.4.1’)
print(Router_IP_Addresses)
```

Task 13:
Remove the first element in the list, and then reprint the list

```
Router_IP_Addresses.pop(0)
print(Router_IP_Addresses)
```

Task 14:
Remove the element in the list with the value '192.168.4.1', and then reprint the list again.

```
Router_IP_Addresses.remove('192.168.4.1')
print(Router_IP_Addresses)
```

Task 15:
Make a string variable that has the value 'Network Plus'. Then use the list concept to print the following
The second letter
Three First letters
All Letters from Letter Four
Letters between Letter Two and Six

```
X= 'Network Plus'

print(X[1])
print(X[:3])
print(X[3:])
print(X[1:6])
```

Task 16:
Create a Tuple that contains three elements 'R1','R2','R3. Then reprint it

```
Hostnames=('R1','R2','R3')

print(Hostnames)
```

Task 17:
Create a python dictionary that has three keys R1,R2,R3 with values of 192.168.1.1,192.168.2.1&192.168.3.1 respectively. Then print the dictionary

```
Routers={“R1”:”192.168.1.1”,”R2”:”192.168.2.1”,”R3”:”192.168.3.1”}
print(Routers)
```

Task 18:
print the value of an element of the above created list that has the key of R1

```
print(Routers[“R1”])
```

Task 19:
Add a new item to the above created dictionary with a value of “192.168.4.1” and key of “R4”. Then reprint the dictionary

```
Routers[“R4”]=’192.168.4.1’
print(Routers)
```

Task 20:
Remove the item from the dictionary whose key is R3, and reprint the dictionary

```
Routers.pop(“R3”)
print(Routers)
```

Lab 2

Condition Statements & Loops

Task 1:
Create a for loop that prints the following list elements:
Routers=[“R1”,”R2”,”R3”]

```
Routers=[“R1”,”R2”,”R3”]
For i in Routers:
    print(i)
```

Task 2:
Create a for loop for a list containing numbers between 0 and 10. The script should be print the list element if it is less than, greater than or equal 4

```
Numbers=[0,1,2,3,4,5,6,7,8,9,10]
for i in Numbers:
  if i<4:
    print(i," is less than 4")
  elif i>4:
    print(i," is greater than 4")
  else:
    print(i," is equal to 4")
```

Task 3:
Create a python script that finds out the prime numbers in a list of numbers between 2&10

```
Numbers=[2,3,4,5,6,7,8,9,10]
for num in Numbers:
  is_prime=True
  if num>1:
    for dev in range(2,num):
      if num%dev==0:
        is_prime=False
        break
```

```
    if is_prime:
        print(num,"is a prime number")
    else:
        print(num,"is not a prime number")
```

Task 4:

Create a script that allows you to add the values of list elements (10 elements), then print all the list, and then only print the positive elements of the list.

```
lists =[]
for i in range (11):
    x=input("Enter lists element:")
    x=int(x)
    lists.append(x)

print(lists)

for j in lists:
    if j<0:
        continue
    else:
        print(j)
```

Task 5:

Create a python script that makes incrementing stars, and every time the stars line is growing until you type exit

```
i=1
while True:
    print("* " *i)
    command = input("Press any key to continue or press 'exit' to exit:").lower()
    If command=='exit':
        print("You chose to exit the code")
        break
    else:
        i+=1
```

Lab 3

Python Functions

Task 1:
Create a python function that prints 'Hello World', whenever it get called

```
def first ():
   Print('Hello World')

first()
```

Task 2:
Create a python function that accepts a string parameter, and then prints Hello followed by the passed parameter

```
def greet(name):
   print("Hello",name)

greet("Waleed")
greet("Ali")
```

Task 3:
Create a python function that accepts two numbers and add them together and return the result

```
def add(a,b):
    Return a+b

Result=add(5,3)
print(Result)
```

Lab 4

File Handling in Python

Task 1:
Create a python script that writes in a file the following sentence

“Hello this is my first text file in python.

I will add more details”

```
with open(“lab.txt”,”w”) as file:
   File.write(“Hello this is my first text file in python \nI will add more details”)
```

Task 2:

Create a python script that reads the above created file and print it in the screen

```
With open(“lab.txt”,”r”) as file:
   Content=file.read()
print(content)
```

Task 3:
In the above created file make extra spaces and extra new lines manually. Then use strip() function to remove any extra spaces and lines and reprint the file

```
with open(“lab.txt”,”r”) as file:
        for lines in file:
                Content=lines.strip()
                print(Content)
```

Task 4:
Use readlines() function to print each line as separate element in a list for the above created file

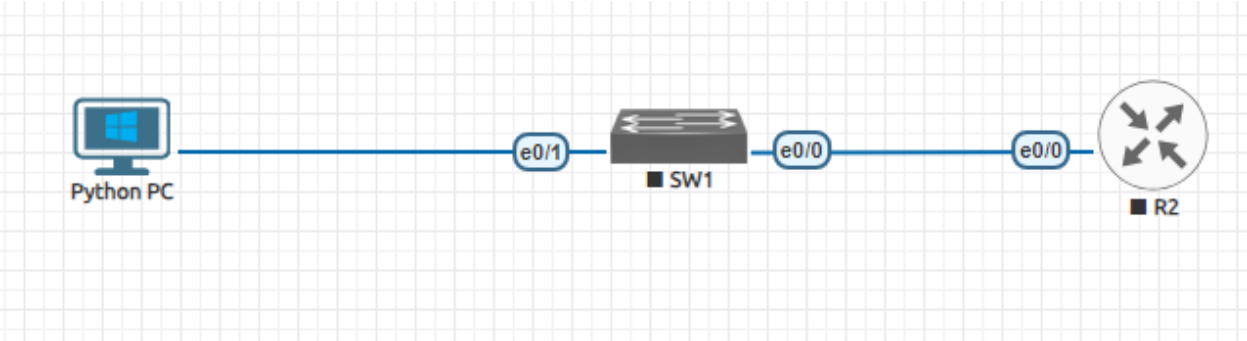
```
with open("lab.txt","r") as file:
        Content=file.readlines()
print(content)
```

Task 5:
Append another line to the above file. The new line contains the following syntax "Appending another line", then print the modified file

```
with open("lab.txt","a") as file:
        file.write("\nAppending another line")
With open("lab.txt","r") as file:
         For lines in file:
                 Print(lines.strip())
```

Lab 5

Basic Telnet to Network Devices using telnetlib



Task 1:
Connect the above network in eve-ng where python host is actually management cloud0 (used to connect to the real network). Make sure the router interface is getting IP address through dhcp. Also make sure the router is telnet enabled using local database and user will be assigned privilege 15. Then test the telnet from your local PC

!R2
Interface e0/0
No shutdown
Ip address dhcp
Exit
!
Line vty 0 4
Login local
Transport input telnet
!
Username Waleed privilege 15 password cisco

Task2:
Write a python script using telnetlib to make a telnet to the router and allow the user to write the command you want to send to cisco router. Consider the following

- The router ip address,username,password are hardcoded in the script.

```
from telnetlib import Telnet

aaa = input('Enter the Command : ')
session = Telnet('192.168.25.135')
session.write(b'Waleed\n')
session.write(b'cisco\n')
session.write(b'terminal length 0\n')
session.write(aaa.encode('ascii') + b'\n')
session.write(b'exit\n')
print(session.read_all().decode('ascii'))
```

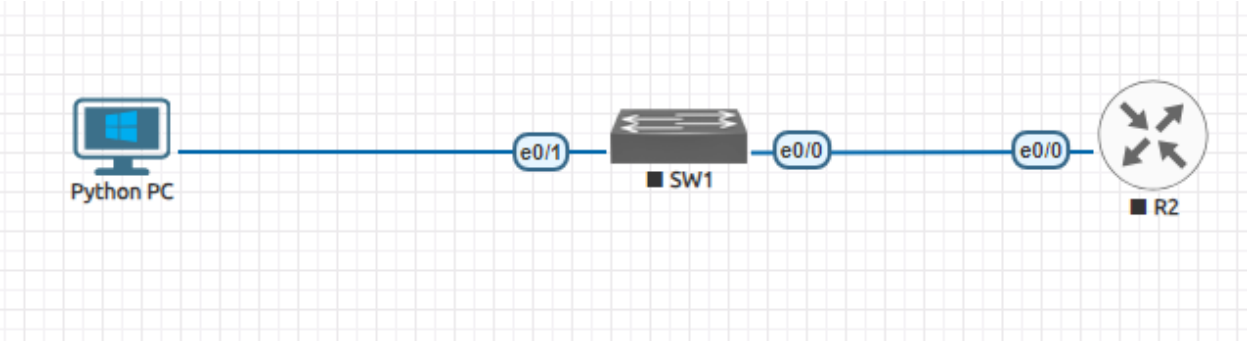
Task3:

Make a script that configure loopback interface in R2 with IP address 1.1.1.1/32, and then configure ospf putting all networks in area 0

```
from telnetlib import Telnet

session = Telnet('192.168.25.135')
session.write(b'Waleed\n')
session.write(b'cisco\n')
session.write(b'configure terminal\n')
session.write(b'interface loopback0\n')
session.write(b'ip address 1.1.1.1 255.255.255.255\n')
session.write(b'router ospf 1\n')
session.write(b'network 192.168.25.0 0.0.0.255 area 0\n')
session.write(b'network 1.1.1.1 0.0.0.0 area 0\n')
session.write(b'end\n')
session.write(b'show ip interface brief\n')
session.write(b'show runn | section ospf\n')
session.write(b'exit\n')
print(session.read_all().decode('ascii'))
```

Lab 6

Improving telnetlib Scripts



Task 1:
Connect the above network in eve-ng where python host is actually management cloud0 (used to connect to the real network). Make sure the router interface is getting IP address through dhcp. Also make sure the router is telnet enabled using local database and user will be assigned privilege 15. Then test the telnet from your local PC

!R2
Interface e0/0
No shutdown
Ip address dhcp
Exit
!
Line vty 0 4
Login local
Transport input telnet
!
Username Waleed privilege 15 password cisco

Task2:
Write a python script using telnetlib to make a telnet to the router to configure IP address for loopback interface. Then run show ip interface brief command. In this Task only username and password are hardcoded and Router IP address as well.

from telnetlib import Telnet

```
interface = input('What Interface would you like to configure : ')
IP_Add = input('Specify the IP Address : ')
Mask = input('Specify the Subnet mask : ')

aaa = Telnet('192.168.25.135')
aaa.write(b'Waleed\n')
aaa.write(b'cisco\n')
aaa.write(b'config t\n')
aaa.write(b'Interface ' + Interface.encode('ascii') + b'\n')
aaa.write(b'IP Address ' + IP_Add.encode('ascii') + b' '  + Mask.encode('ascii') +
b'\n')
aaa.write(b'no shut\n')
aaa.write(b'end\n')
aaa.write(b'sh ip int brief\n')
aaa.write(b'exit\n')
print (aaa.read_all().decode('ascii'))
```

Task 3:

Improve the previous script, by making username, password and IP address of the router is entered by the user in an interactive mode.

```
from telnetlib import Telnet
import getpass

HOST = input('Specify the Host IP : ')
USER = input('Specify the Username: ')
PASS = getpass.getpass('Specify the password: ')
Interface = input('What Interface would you like to configure : ')
IP_Add= input('Specify the IP Address : ')
Mask = input('Specify the Subnet mask : ')

aaa = Telnet(HOST)
aaa.write(USER.encode('ascii') + b'\n')
aaa.write(PASS.encode('ascii') + b'\n')
```

```
aaa.write(b'config t\n')
aaa.write(b'Interface ' + Interface.encode('ascii') + b'\n')
aaa.write(b'IP Address ' + IP_Add.encode('ascii') + b' ' + Mask.encode('ascii') +
b'\n')
aaa.write(b'end\n')
aaa.write(b'sh ip int brief\n')
aaa.write(b'exit\n')
print (aaa.read_all().decode('ascii'))
```

Task 4:

Configure the switch vlan 1 to have ip address in the same range of the router, we can configure it for example as 192.168.25.136. enable telnet using local credentials. Configure the local credentials to be as same as for the router with the same privilege. Then test the telnet to the switch from your laptop

```
!SW1
Interface vlan 1
No shutdown
Ip address 192.168.25.136 255.255.255.0
!
Line vty 0 4
Login local
Transport input telnet
Exit
!
Username Waleed privilege 15 password cisco
!
```

Task 5:

Write a python program that telnet to the switch and create vlans from vlan 2 to vlan 9

```
from telnetlib import Telnet
import getpass
```

```
host=input("please enter the ip address of the device:")
username=input("please enter the username")
password=getpass.getpass("please enter the password:")

aaa=Telnet(host)
aaa.write(username.encode("ascii") + b"\n")
aaa.write(password.encode("ascii") + b"\n")
aaa.write(b"config t\n")
aaa.write(b"vlan 2\n")
aaa.write(b"vlan 3\n")
aaa.write(b"vlan 4\n")
aaa.write(b"vlan 5\n")
aaa.write(b"vlan 6\n")
aaa.write(b"vlan 7\n")
aaa.write(b"vlan 8\n")
aaa.write(b"vlan 9\n")

aaa.write(b"end\n")
aaa.write(b"terminal length 0\n")
aaa.write(b"show vlan\n")
aaa.write(b"exit\n")
output=aaa.read_all().decode("ascii")
print(output)
```

Task 6:

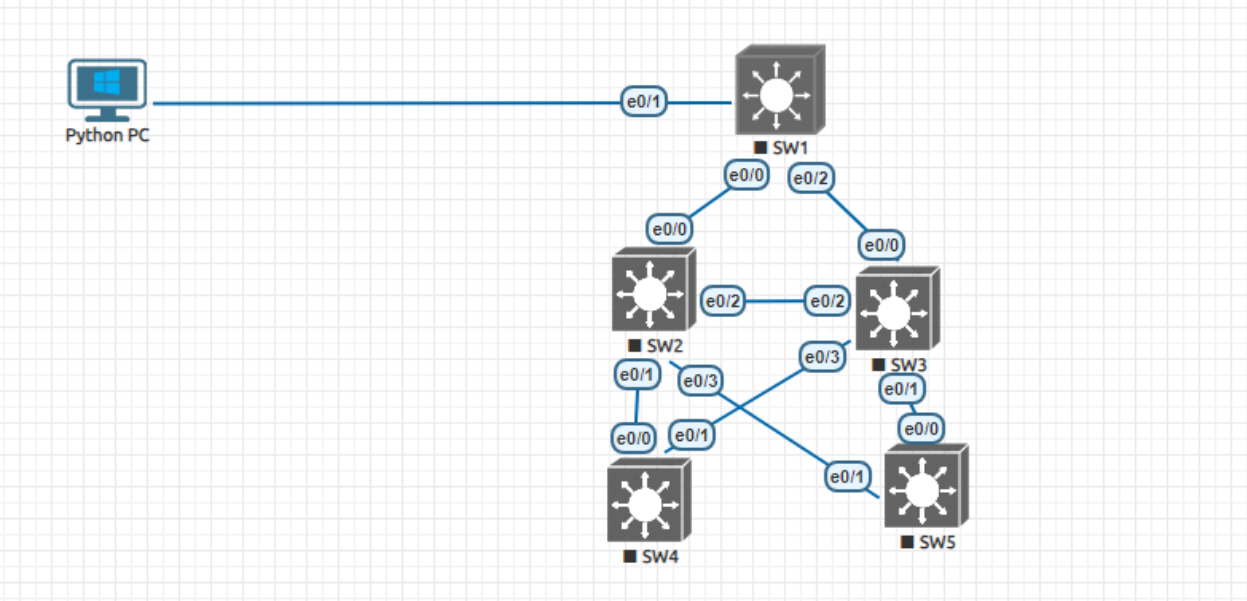
Improve the previous code by creating vlans from vlan 10 to vlan 19 using for loop, and assign them names

```
from telnetlib import Telnet
import getpass

host=input("please enter the ip address of the device:")
username=input("please enter the username")
password=getpass.getpass("please enter the password:")
```

```
aaa=Telnet(host)
aaa.write(username.encode("ascii") + b"\n")
aaa.write(password.encode("ascii") + b"\n")
aaa.write(b"configure terminal \n")
for i in range(10,20):
   x=str(i)
   aaa.write(b"vlan " + x.encode("ascii") + b"\n")
   aaa.write(b"name python_vlan_" + x.encode("ascii") + b"\n")
aaa.write(b"end\n")
aaa.write(b"terminal length 0\n")
aaa.write(b"show vlan\n")
aaa.write(b"exit\n")
output=aaa.read_all().decode("ascii")
print(output)
```

Lab 7

Configuring Multiple Vlans on Multiple Switches using telnetlib



Task 1:
Connect the above network, and make sure switches have IP addresses configured 192.168.25.136-.140. Make sure telnet is enabled, local username and password configured with the highest privilege. Then make a telnet test from your PC.

!All_SW
Username Waleed privilege 15 password cisco
Line vty 0 4
Login local
Transport input telnet
!

!SW1
Interface vlan 1
No shutdown
Ip address 192.168.25.136 255.255.255.0

```
!
!SW2
Interface vlan 1
No shutdown
Ip address 192.168.25.137 255.255.255.0
!
!SW3
Interface vlan 1
No shutdown
Ip address 192.168.25.138 255.255.255.0
!
!SW4
Interface vlan 1
No shutdown
Ip address 192.168.25.139 255.255.255.0
!
!SW5
Interface vlan 1
No shutdown
Ip address 192.168.25.140 255.255.255.0
!
```

Task 2:

Create a python script using telnetlib to configure vlan 2-10, and assign them a name

```
from telnetlib import Telnet
import getpass



username=input("Please enter the username:")
password=getpass.getpass("please enter the password:")

for i in range(136,141):
```

```
    x=str(i)
    host="192.168.25." + x
    aaa=Telnet(host)
    print("Connecting to host ",host)
    aaa.write(username.encode("ascii") + b"\n")
    aaa.write(password.encode("ascii") + b"\n")
    aaa.write(b"config t\n")
    for j in range(2,11):
        y=str(j)
        aaa.write(b"vlan " + y.encode("ascii") + b"\n")
        aaa.write(b"name python_vlan" + y.encode("ascii") + b"\n")
    aaa.write(b"end\n")
    aaa.write(b"terminal length 0\n")
    aaa.write(b"show vlan\n")
    aaa.write(b"exit\n")
    print(aaa.read_all().decode("ascii"))
    aaa.close()
```

Task 3:

Repeat the same scenario for the switches, however in this time creating vlans from 11 to 20. Also make the switch IP addresses to be listed in a file and make sure the script is reading from that file.

```
from telnetlib import Telnet
import getpass



username=input("Please enter the username:")
password=getpass("please enter the password:")

with open("switches.txt","r") as file:
    for line in file:
        host=line.strip()
        aaa=Telnet(host)
        print("Connecting to host ",host)
```

```
    aaa.write(username.encode("ascii") + b"\n")
    aaa.write(password.encode("ascii") + b"\n")
    aaa.write(b"config t\n")
    for j in range(11,21):
       y=str(j)
       aaa.write(b"vlan " + y.encode("ascii") + b"\n")
       aaa.write(b"name python_vlan" + y.encode("ascii") + b"\n")
    aaa.write(b"end\n")
    aaa.write(b"terminal length 0\n")
    aaa.write(b"show vlan\n")
    aaa.write(b"exit\n")
    print(aaa.read_all().decode("ascii"))
    aaa.close()
```

Task 4:

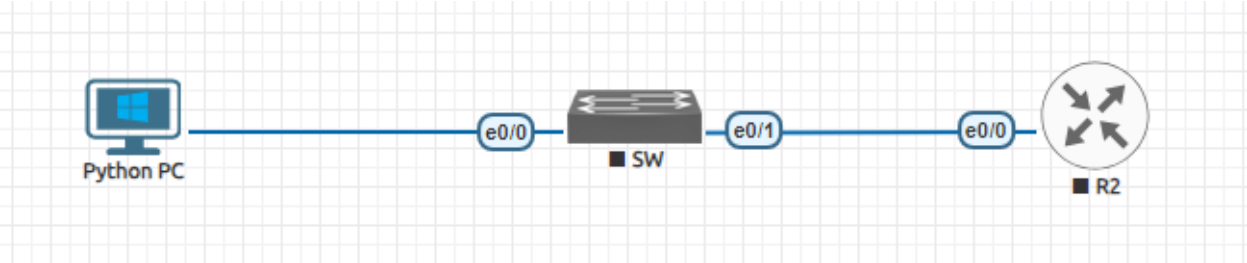
Make Backup configuration to all switches, and save them in text files.

```
from telnetlib import Telnet
import getpass



username=input("Please enter the username:")
password=input("please enter the password:")

with open("switches.txt","r") as file:
   for line in file:
      host=line.strip()
      aaa=Telnet(host)
      print("Connecting to host ",host)
      aaa.write(username.encode("ascii") + b"\n")
      aaa.write(password.encode("ascii") + b"\n")
      aaa.write(b"terminal length 0\n")
      aaa.write(b"show runn\n")
      aaa.write(b"exit\n")
      output=aaa.read_all().decode("ascii")
```

```
with open(f"Switch_{host}_config.tx","w") as file:
   file.write(output)
aaa.close()
```

Lab 8

SSH to Network Device Using Paramiko



Task 1:
Connect the above network. Make the router interface to get IP address from DHCP. Also make sure the ssh is enabled on the router. Local username and password are also configured.

!R2
Interface e0/0
No shutdown
Ip address dhcp
Exit
Hostname R2
Ip domain-name cisco.com
Crypto key generate rsa modulus 1024
Username Waleed privilege 15 password cisco
Line vty 0 4
Login local
Transport input ssh

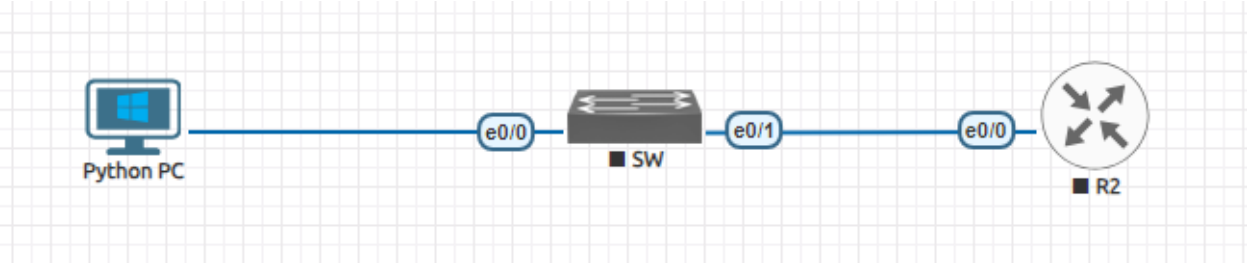
Task 2:
Use Paramiko to configure loopback interface with ip address of 1.1.1.1/32 and configure Ospf routing protocol in the router as well

import paramiko
import time

```
host="192.168.25.135"
username="Waleed"
password="cisco"

ssh_client=paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(hostname=host,username=username,password=password)

print(f"Successful Connection to {host}")
remote_connection=ssh_client.invoke_shell()
time.sleep(1)
remote_connection.send(b"configure terminal\n")
time.sleep(1)
remote_connection.send(b"interface lo0\n")
remote_connection.send(b"no shutdown\n")
remote_connection.send(b"ip address 1.1.1.1 255.255.255.255\n")
time.sleep(1)
remote_connection.send(b"router ospf 1\n")
remote_connection.send(b"network 1.1.1.1 0.0.0.0 area 0\n")
remote_connection.send(b"network 192.168.25.0 0.0.0.255 area 0\n")
remote_connection.send(b"end\n")
remote_connection.send(b"terminal length 0\n")
time.sleep(1)
remote_connection.send(b"exit\n")
output=remote_connection.recv(65535).decode("ascii")
time.sleep(1)
print(output)
ssh_client.close()
```

Lab 9

SSH to Network Device Using Netmiko



Task 1:
Connect the above network, make sure the router is getting IP address from DHCP. Then enable SSH on the router.

```
!R2
Int e0/0
No shutdown
Ip address dhcp
Exit
!
Username Waleed privilege 15 password cisco
Hostname R2
Ip domain-name cisco.com
Crypto key generate rsa modulus 1024
!
Line vty 0 4
Login local
Transport input ssh
```

Task 2:
Make a python script that send a *show ip interface brief command* to R2, and print the result. Install the netmiko first

```
From netmiko import ConnectHandler
```

```
AAA={
“device_type”:”cisco_ios”,
“ip”:”192.168.25.135”,
“username”:”Waleed”,
“password”:”cisco”
}

Connection=ConnectHandler(**AAA)
Output=Connection.send_command(“show ip interface brief”)
print(Output)
```

Task 3:

Make a python script that configure loopback interface on R2 and give it ip address of 1.1.1.1 and then configure ospf and advertising the networks of the router.

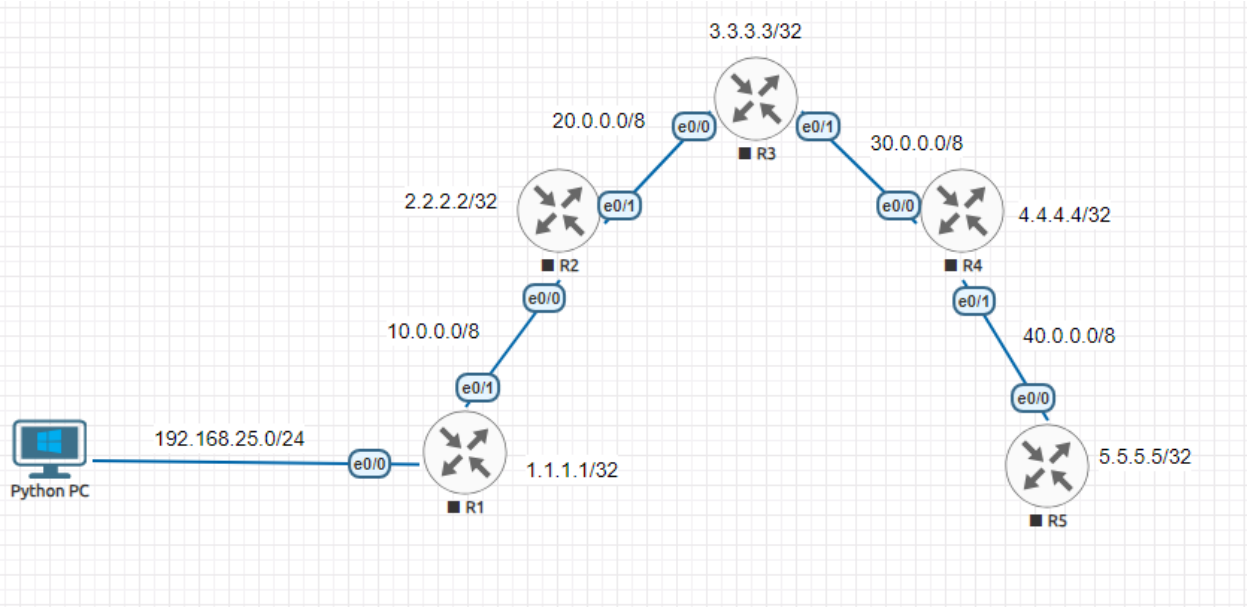
```
From netmiko import ConnectHandler

AAA={
“device_type”:”cisco_ios”,
“ip”:”192.168.25.135”,
“username”:”Waleed”,
“password”:”cisco”
}

Connection=ConnectHandler(**AAA)
Commands=[“interface lo0”,”ip address 1.1.1.1 255.255.255.255”,”router ospf 1”,
“network 192.168.25.0 0.0.0.255 area 0”,”network 1.1.1.1 0.0.0.0 area 0”]

Connection.send_config_set(commands)

Output1=Connection.send_command(“show ip interface brief”)
Output2=Connection.send_command(“show runn | section ospf”)
print(Output1)
Print(Output2)
```

Task 4:
Make a python script that repeats the same previous lab by adding another loopback interface with ip address 2.2.2.2, and reconfigure the ospf. However in this task use non-hardcoded values for IP address, username and password.

```
From netmiko import ConnectHandler
Import getpass
Host=input("Please enter the Hostname:")
User=input("Please enter the username:")
Password=getpass.getpass("Please enter the password:")


AAA={
"device_type":"cisco_ios",
"ip":Host,
"username":User,
"password":Password
}

Connection=ConnectHandler(**AAA)
Commands=["interface lo1","ip address 2.2.2.2 255.255.255.255","router ospf 1",
"network 2.2.2.2 0.0.0.0 area 0"]

Connection.send_config_set(commands)

Output1=Connection.send_command("show ip interface brief")
Output2=Connection.send_command("show runn | section ospf")
print(Output1)
Print(Output2)
```

Lab 10

Collecting Interface Information from Multiple Routers



Task 1:
Connect the above network, ensure ip addresses are assigned as depicted in the figure. Routing protocol is enabled to ensure the connectivity, and ssh is enabled per device

!All_Routers
Username Waleed privilege 15 password cisco
Ip domain-name cisco.com
Crypto key generate rsa modulus 1024
!
Line vty 0 4
Login local
Transport input ssh

!R1
Hostname R1
Int e0/0
No shut

```
Ip address dhcp
!
Int e0/1
No shut
Ip address 10.0.0.1 255.0.0.0
!
Interface lo0
Ip add 1.1.1.1 255.255.255.255
!
Router eigrp 100
Network 10.0.0.0
Network 192.168.25.0
Network 1.0.0.0
No auto summary


!R2
Hostname R2
Int e0/0
No shut
Ip address 10.0.0.2 255.0.0.0
!
Int e0/1
No shut
Ip address 20.0.0.2 255.0.0.0
!
Interface lo0
Ip add 2.2.2.2 255.255.255.255
!
Router eigrp 100
Network 10.0.0.0
Network 20.0.0.0
Network 2.0.0.0
No auto-summary


!R3
```

```
Hostname R3
Int e0/0
No shut
Ip address 20.0.0.3 255.0.0.0
!
Int e0/1
No shut
Ip address 30.0.0.3 255.0.0.0
!
Interface lo0
Ip add 3.3.3.3 255.255.255.255
!
Router eigrp 100
Network 20.0.0.0
Network 30.0.0.0
Network 3.0.0.0
No auto-summary


!R4
Hostname R4
Int e0/0
No shut
Ip address 30.0.0.4 255.0.0.0
!
Int e0/1
No shut
Ip address 40.0.0.4 255.0.0.0
!
Interface lo0
Ip add 4.4.4.4 255.255.255.255
!
Router eigrp 100
Network 30.0.0.0
Network 40.0.0.0
Network 4.0.0.0
No auto-summary
```

```
!R5
Hostname R5
Int e0/0
No shut
Ip address 40.0.0.5 255.0.0.0
!
Interface lo0
Ip add 5.5.5.5 255.255.255.255
!
Router eigrp 100
Network 40.0.0.0
Network 5.0.0.0
No auto-summary
```

Task 2:

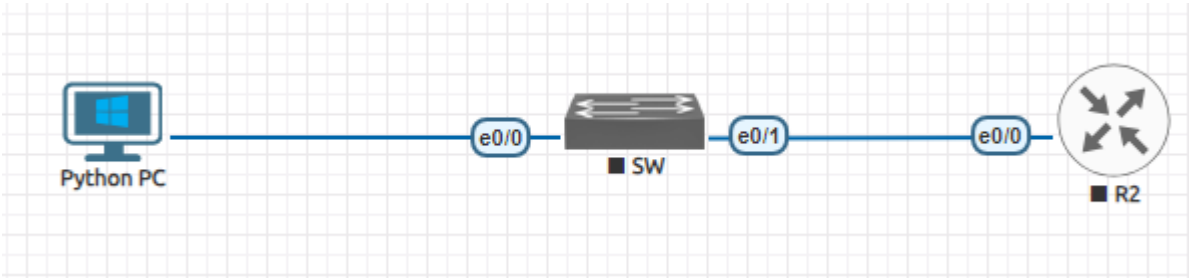
Make a text file that contains the ip addresses of routers loopback interfaces. Then make a python script that uses this file to collect the interface information.

```
from netmiko import ConnectHandler
import getpass

username=input("Please enter username:")
password=getpass.getpass("Please enter password:")

with open("devices.txt","r") as routers:
   for IP in routers:
      AAA={
         "device_type":"cisco_ios",
         "ip":IP,
         "username":username,
         "password":password
      }
      connection=ConnectHandler(**AAA)
      print(f"Connecting to {IP}")
```

```
output=connection.send_command("show ip interface brief")
print(output)
print()
connection.disconnect()
```

Lab 11

Making a Configuration Backup of a Single Router



Task 1:
Connect the above network, ensure Router is getting IP address from DHCP. Ensure SSH is enabled on the router.

```
!R2
Hostname R2
Username Waleed privilege 15 password cisco
Crypto key generate rsa modulus 1024
Ip domain-name cisco.com
Line vty 0 4
Login local
Transport input ssh
!
Interface e0/0
Ip address dhcp
```

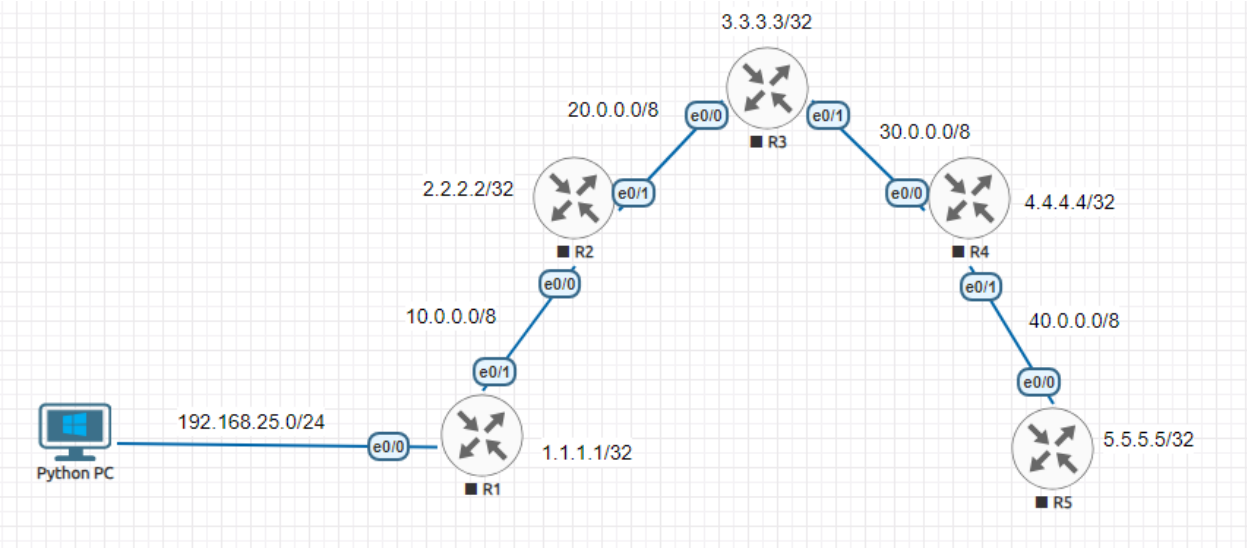
Task 2:
Make a python script that creates a configuration backup of R2, and saves it in text file

```
from netmiko import ConnectHandler
import getpass

username=input("Please Enter username:")
password=getpass.getpass("Please Enter password:")
```

```
AAA={
   "device_type":"cisco_ios",
   "ip":"192.168.25.135",
   "username":username,
   "password":password
}

connection=ConnectHandler(**AAA)
hostname=connection.send_command("show runn | inc hostname")
x=hostname.split()
device=x[1]
print(f"Backing up {device}")
FFF=device+"_Backup.txt"
CCC=connection.send_command("show runn")
with open(FFF,"w") as file:
   file.write(CCC)
   file.write("\n")
print(device + " Backup Successfully")
connection.disconnect()
```

Lab 12

Making a Configuration Backup of Multiple Routers & Applying Extra Configurations



Task 1:

Connect the above network, ensure ip addresses are assigned as depicted in the figure. Routing protocol is enabled to ensure the connectivity, and ssh is enabled per device

!All_Routers
Username Waleed privilege 15 password cisco
Ip domain-name cisco.com
Crypto key generate rsa modulus 1024
!
Line vty 0 4
Login local
Transport input ssh

!R1
Hostname R1
Int e0/0
No shut
Ip address dhcp

```
!
Int e0/1
No shut
Ip address 10.0.0.1 255.0.0.0
!
Interface lo0
Ip add 1.1.1.1 255.255.255.255
!
Router eigrp 100
Network 10.0.0.0
Network 192.168.25.0
Network 1.0.0.0
No auto-summary


!R2
Hostname R2
Int e0/0
No shut
Ip address 10.0.0.2 255.0.0.0
!
Int e0/1
No shut
Ip address 20.0.0.2 255.0.0.0
!
Interface lo0
Ip add 2.2.2.2 255.255.255.255
!
Router eigrp 100
Network 10.0.0.0
Network 20.0.0.0
Network 2.0.0.0
No auto-summary


!R3
Hostname R3
```

```
Int e0/0
No shut
Ip address 20.0.0.3 255.0.0.0
!
Int e0/1
No shut
Ip address 30.0.0.3 255.0.0.0
!
Interface lo0
Ip add 3.3.3.3 255.255.255.255
!
Router eigrp 100
Network 20.0.0.0
Network 30.0.0.0
Network 3.0.0.0
No auto-summary


!R4
Hostname R4
Int e0/0
No shut
Ip address 30.0.0.4 255.0.0.0
!
Int e0/1
No shut
Ip address 40.0.0.4 255.0.0.0
!
Interface lo0
Ip add 4.4.4.4 255.255.255.255
!
Router eigrp 100
Network 30.0.0.0
Network 40.0.0.0
Network 4.0.0.0
No auto-summary
```

```
!R5
Hostname R5
Int e0/0
No shut
Ip address 40.0.0.5 255.0.0.0
!
Interface lo0
Ip add 5.5.5.5 255.255.255.255
!
Router eigrp 100
Network 40.0.0.0
Network 5.0.0.0
No auto-summary
```

Task 2:

Create a python script that makes a backup configuration for the five routers and saves the configuration in a text file

```
from netmiko import ConnectHandler
import getpass

username=input("Please Enter username:")
password=getpass.getpass("Please Enter password:")

with open("device.txt","r") as file:
   for IP in file:
      Router={
         "device_type":"cisco_ios",
         "ip":IP,
         "username":username,
         "password":password
      }
      connection=ConnectHandler(**Router)
      hostname=connection.send_command("show runn | inc hostname")
      x=hostname.split()
      device=x[1]
```

```
print(f"Backing up {device}")
FFF=device+"_Backup.txt"
CCC=connection.send_command("show runn")
with open(FFF,"w") as file:
   file.write(CCC)
   file.write("\n")
print(device + " Backup Successfully")
connection.disconnect()
```

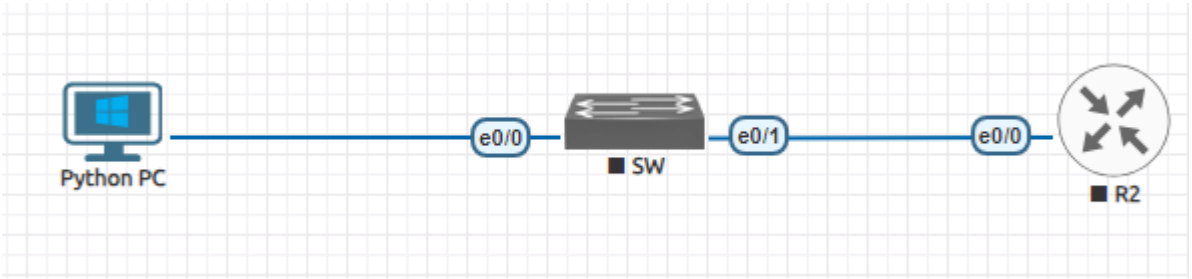
Task 3:

Create a python script that applies configuration to the five routers. The configuration is pulled from text files. The configuration should include applying OSPF configuration for the five routers

```
from netmiko import ConnectHandler
import getpass

username=input("Please Enter username:")
password=getpass.getpass("Please Enter password:")

with open("device.txt","r") as file:
   for IP in file:
      Router={
         "device_type":"cisco_ios",
         "ip":IP,
         "username":username,
         "password":password
      }
      connection=ConnectHandler(**Router)
      hostname=connection.send_command("show runn | inc hostname")
      x=hostname.split()
      device=x[1]
      print(f"Configuring {device}")
      FFF=device+".txt"
      connection.send_config_from_file(FFF)
      print(f"{device} is configured")
      connection.disconnect()
```

Lab 13

Using NAPALM to Collect Router's Information



Task 1:
Connect the above network, ensure Router is getting IP address from DHCP. Ensure SSH is enabled on the router.

!R2
Hostname R2
Username Waleed privilege 15 password cisco
Crypto key generate rsa modulus 1024
Ip domain-name cisco.com
Line vty 0 4
Login local
Transport input ssh
!
Interface e0/0
Ip address dhcp

Task 2:
Make a python script that connects to R2 using Napalm and gets R2 device information, Interface IP information and the version Information

```
from napalm import get_network_driver

driver=get_network_driver("ios")

device=driver(hostname="192.168.25.135",username="Waleed",password="cisco
```

```
")
device.open()
print("Connected Successfully")
output1=device.get_facts()
output2=device.get_interfaces_ip()
output3=device.cli(["show ip interface brief","show version"])
print(output1)
print()
print(output2)
print()
print(output3)
device.close()
print("Disconnected Successfully")
```