Experiment No : 9

Name : Chinnakrishna N Neelakanteswara Achary
Class : SE-13 Batch-A
Roll No : 1
Subject : Python Lab

1) **Aim :** To create sockets for information exchange between client and server.

2) **Software Used :** VS Code.

3) **Theory :**

   1. **Socket:** Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection.

   2. **gethostname():**
      - The Python function socket.gethostname() returns the host name of the current system under which the Python interpreter is executed.
      - This Python function can be combined with socket.gethostbyname()to get the IP address of the local host.

   3. **bind():**The bind() method of Python's socket class assigns an IP address and a port number to a socket instance. The bind() method is used when a socket needs to be made a server socket. As server programs listen on published ports, it is required that a port and the IP address to be assigned explicitly to a server socket.

   4. **listen():** A server has a listen() method which puts the server into listen mode. This allows the server to listen to incoming connections.

   5. **accept():** accept() waits for incoming connection attempts and blocks until a client tries to connect. Note that accept() returns a

tuple containing the new socket object and the remote address. It'll simply not return if there is not client trying to connect.

6. <u>recv():</u>The recv method receives up to buffersize bytes from the socket. When no data is available, it blocks until at least one byte is available or until the remote end is closed. When the remote end is closed and all data is read, it returns an empty byte string.

7. <u>decode():</u> decode() is a method specified in Strings in Python 2. This method is used to convert from one encoding scheme, in which argument string is encoded to the desired encoding scheme. This works opposite to the encode. It accepts the encoding of the encoding string to decode it and returns the original string.

8. <u>encode():</u> The encode() method returns an encoded version of the given string.

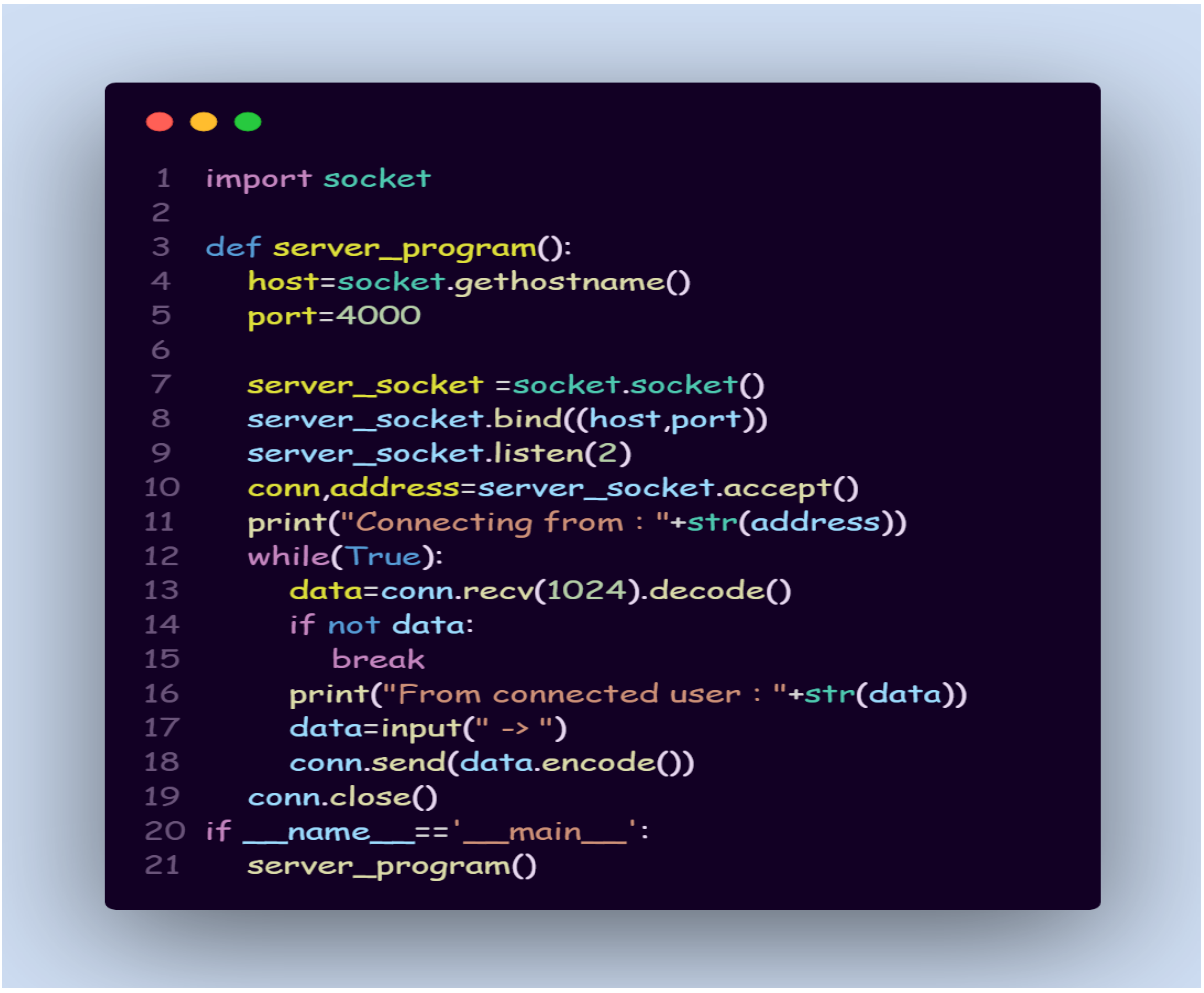
9. <u>send():</u>The send()method of Python's socket class is used to send data from one socket to another socket. The send()method can only be used with a connected socket. That is, send() can be used only with a TCP based socket and it can not be used with UDP socket.

10. <u>connect():</u> The connect() method of Python's socket module, connects a TCP(Transmission Control Protocol) based client socket to a TCP based server socket. The connect() function initiates a 3-way handshake with the server socket and establishes the connection.

11. <u>close():</u> The close() method of a file object flushes any unwritten information and closes the file object, after which no more writing can be done. Python automatically closes a file when the reference object of a file is reassigned to another file. It is a good practice to use the close() method to close a file

4) <u>Snapshots of code and outputs :</u>

<u>Socket server code:</u>

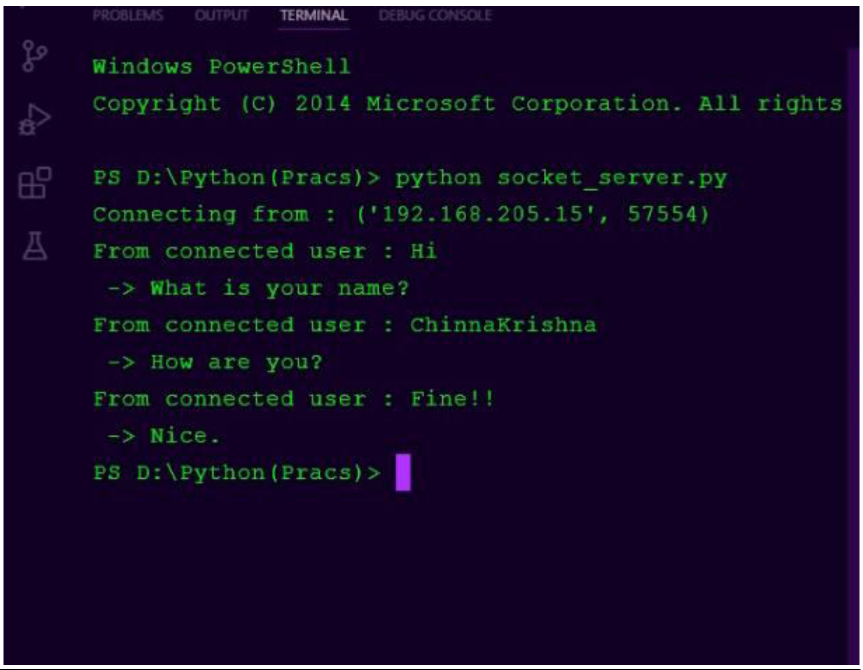


```python
import socket

def server_program():
    host=socket.gethostname()
    port=4000

    server_socket =socket.socket()
    server_socket.bind((host,port))
    server_socket.listen(2)
    conn,address=server_socket.accept()
    print("Connecting from : "+str(address))
    while(True):
        data=conn.recv(1024).decode()
        if not data:
            break
        print("From connected user : "+str(data))
        data=input(" -> ")
        conn.send(data.encode())
    conn.close()
if __name__=='__main__':
    server_program()
```

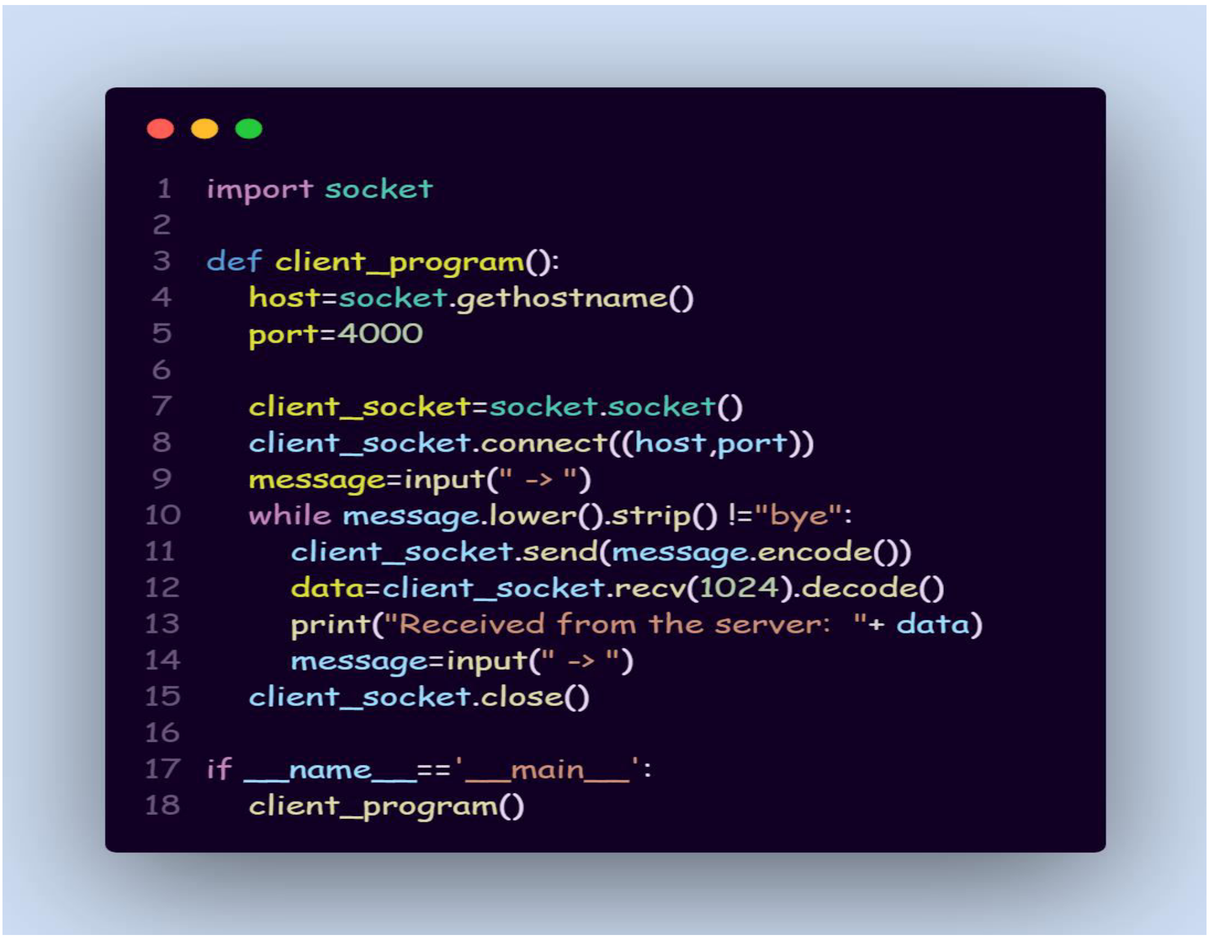
<u>Output of server :</u>



```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights

PS D:\Python(Pracs)> python socket_server.py
Connecting from : ('192.168.205.15', 57554)
From connected user : Hi
 -> What is your name?
From connected user : ChinnaKrishna
 -> How are you?
From connected user : Fine!!
 -> Nice.
PS D:\Python(Pracs)>
```

Socket client code :



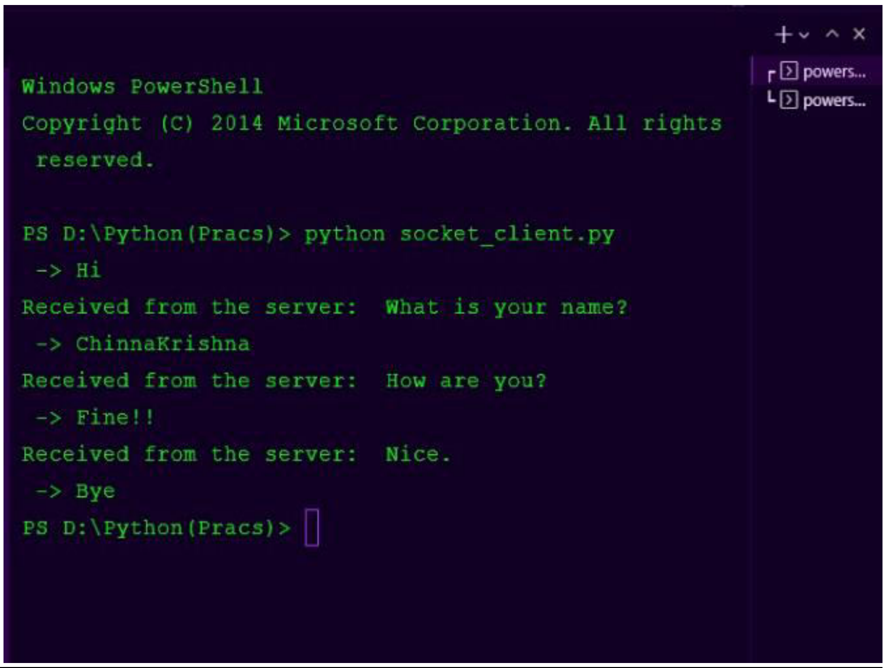
```python
import socket

def client_program():
    host=socket.gethostname()
    port=4000

    client_socket=socket.socket()
    client_socket.connect((host,port))
    message=input(" -> ")
    while message.lower().strip() !="bye":
        client_socket.send(message.encode())
        data=client_socket.recv(1024).decode()
        print("Received from the server:  "+ data)
        message=input(" -> ")
    client_socket.close()

if __name__=='__main__':
    client_program()
```

Output of client :



```
Windows PowerShell
Copyright (C) 2014 Microsoft Corporation. All rights
 reserved.

PS D:\Python(Pracs)> python socket_client.py
 -> Hi
Received from the server:  What is your name?
 -> ChinnaKrishna
Received from the server:  How are you?
 -> Fine!!
Received from the server:  Nice.
 -> Bye
PS D:\Python(Pracs)> 
```

5) **<u>Conclusions :</u> By performing this experiment the concept of socket is cleared.**