Ex1 a

Save as : chatserver.py

import time, socket, sys

print('Setup Server...')

time.sleep(1)

#Get the hostname, IP Address from socket and set Port

soc = socket.socket()

host\_name = socket.gethostname()

ip = socket.gethostbyname(host\_name)

port = 1234

soc.bind((host\_name, port))

print(host\_name, '({})'.format(ip))

name = input('Enter name: ')

soc.listen(1) #Try to locate using socket

print('Waiting for incoming connections...')

connection, addr = soc.accept()

print("Received connection from ", addr[0], "(", addr[1], ")\n")

print('Connection Established. Connected From: {}, ({})'.format(addr[0], addr[0]))

#get a connection from client side

client\_name = connection.recv(1024)

client\_name = client\_name.decode()

print(client\_name + ' has connected.')

print('Press [bye] to leave the chat room')

connection.send(name.encode())

while True:

message = input('Me > ')

if message == '[bye]':

message = 'Good Night...'

connection.send(message.encode())

print("\n")

break

connection.send(message.encode())

message = connection.recv(1024)

message = message.decode()

print(client\_name, '>', message)

client

save as : chatclient.py

import time, socket, sys

print('Client Server...')

time.sleep(1)

#Get the hostname, IP Address from socket and set Port

soc = socket.socket()

shost = socket.gethostname()

ip = socket.gethostbyname(shost)

#get information to connect with the server

print(shost, '({})'.format(ip))

server\_host = input('Enter server\'s IP address:')

name = input('Enter Client\'s name: ')

port = 1234

print('Trying to connect to the server: {}, ({})'.format(server\_host, port))

time.sleep(1)

soc.connect((server\_host, port))

print("Connected...\n")

soc.send(name.encode())

server\_name = soc.recv(1024)

server\_name = server\_name.decode()

print('{} has joined...'.format(server\_name))

print('Enter [bye] to exit.')

while True:

message = soc.recv(1024)

message = message.decode()

print(server\_name, ">", message)

message = input(str("Me > "))

if message == "[bye]":

message = "Leaving the Chat room"

soc.send(message.encode())

print("\n")

break

soc.send(message.encode())

echo server

save as: echos.py

import socket

# next create a socket object

s = socket.socket()

print ("Socket successfully created")

# reserve a port on your computer in our

# case it is 12345 but it can be anything

port = 12345

# Next bind to the port

# we have not typed any ip in the ip field

# instead we have inputted an empty string

# this makes the server listen to requests

# coming from other computers on the network

s.bind(('', port))

print ("socket binded to %s" %(port))

# put the socket into listening mode

s.listen(5)

print ("socket is listening")

# a forever loop until we interrupt it or

# an error occurs

while True:

c, addr = s.accept()

print ('Got connection from', addr )

c.send('Thank you for connecting'.encode())

c.close()

break

client

save as :echoc.py

import socket

# Create a socket object

s = socket.socket()

# Define the port on which you want to connect

port = 12345

# connect to the server on local computer

s.connect(('127.0.0.1', port))

# receive data from the server and decoding to get the string.

print (s.recv(1024).decode())

# close the connection

s.close()

ftp

server

import socket

import sys

HOST = ""

PORT = 12345

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.bind((HOST, PORT))

s.listen(5)

print("Listening ...")

conn, addr = s.accept()

print(" Client connected: ", addr)

f = open("file\_received.txt", "wb")

while True:

data = conn.recv(4096)

if not data:

break

f.write(data)

f.close()

print("Download complete!")

conn.close()

print("Client disconnected")

sys.exit(0)

client

import socket

import sys

HOST = "localhost"

PORT = 12345

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.connect((HOST, PORT))

print(" Connected with Server")

f\_send = "file\_to\_send.txt"

f=open(f\_send, "rb")

print("Sending file...")

data = f.read()

s.sendall(data)

s.close()

print("Disconnected")

sys.exit(0)