

**Name : Kansara Anjali | Class : A | Branch : Cyber Security  
Semester: 5 | Enrollment No: 23162171032|Batch:52**

**Institute of Computer Technology  
B. Tech Computer Science and Engineering**

**Sub:CN  
Practical 7**

**Aim:** To implement Socket Programming

**Scenario:**

An organization named Albert Enterprise has established two departments for better performance of the company, as each department will be having some specific set of tasks to perform. So, this will reduce the time and increase the efficiency of the work. As both the departments are dependent on each other, they need to communicate more frequently. To solve the problem, the IT department has suggested the option to create a chat application using socket programming which will work only in the office premises. So, help the IT professionals to create the chat application.

Make sure that the application has the below mentioned features:

- 1) Department 1 will be set as the SERVER while department 2 will be set as a CLIENT device.
- 2) The message received by CLIENT or SERVER must be displayed with a time stamp.

3) If any of the devices irrespective of CLIENT or SERVER has sent the message that the “quit”, then connection should be closed on both the ends.

4) There is no restriction on the protocol selection, you can use UDP or TCP. Justify the reason for selection of the specific protocol.

### Expected Output:

```
-----  
SERVER
```

```
-----  
SERVER is listening..  
Connection accepted from ('192.168.1.6', 53792)  
CLIENT [2024-10-06 21:13:08]: Hi I am Client 1  
ENTER TEXT: I am Server  
CLIENT [2024-10-06 21:13:32]: My task is done  
ENTER TEXT: quit  
CLIENT [2024-10-06 21:13:36]:
```

```
-----  
CLIENT
```

```
-----  
Hello there! msg from SERVER  
ENTER TEXT: Hi I am Client 1  
SERVER [2024-10-06 21:13:17]: I am Server  
ENTER TEXT: My task is done  
SERVER [2024-10-06 21:13:36]: quit
```

### Server Code:

```
import socket  
  
from datetime import datetime
```

```
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server.bind(("0.0.0.0", 5050))

server.listen(1)


print("----- SERVER -----")

print("SERVER is listening..")


conn, addr = server.accept()

print("Connection accepted from", addr)


# Send welcome message to client

conn.send("Hello there! msg from SERVER".encode())


while True:

    msg = conn.recv(1024).decode()

    timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    print(f"CLIENT [{timestamp}]: {msg}")

    if msg.lower() == "quit":

        break

    text = input("ENTER TEXT: ")

    conn.send(text.encode())
```

```
        if text.lower() == "quit":  
            break  
  
conn.close()  
server.close()
```

## Client Code:

```
import socket  
  
from datetime import datetime  
  
client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
client.connect(("127.0.0.1", 5050))  
  
print("----- CLIENT -----")  
  
# Receive welcome msg from server  
msg = client.recv(1024).decode()  
print(msg)  
  
while True:  
    text = input("ENTER TEXT: ")
```

```
client.send(text.encode())

if text.lower() == "quit":
    break

msg = client.recv(1024).decode()

timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

print(f"SERVER [{timestamp}]: {msg}")

if msg.lower() == "quit":
    break

client.close()
```

Output:

```
server.py
1 import socket
2 from datetime import datetime
3
4 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 server.bind(("0.0.0.0", 5050))
6 server.listen(1)
7
8 print("----- SERVER -----")
9 print("SERVER is listening..")
10
11 conn, addr = server.accept()
12 print("Connection accepted from", addr)
13
14 # Send welcome message to client
15 conn.send("Hello there! msg from SERVER".encode())
16
17 while True:
```

```
client.py
1 import socket
2 from datetime import datetime
3
4 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 client.connect(("127.0.0.1", 5050))
6
7
8 print("----- CLIENT -----")
9
10 # Receive welcome msg from server
11 msg = client.recv(1024).decode()
12 print(msg)
13
14 while True:
15     text = input("ENTER TEXT: ")
16     client.send(text.encode())
17
```

```
PS D:\SEM 5\CH>
PS D:\SEM 5\CH> py server.py
-----
OSError: [WinError 10048] Only one usage of each socket address (protocol/network address/port) is normally permitted
PS D:\SEM 5\CH> py server.py
-----
SERVER
SERVER is listening..
Connection accepted from ('127.0.0.1', 49916)
CLIENT [2025-11-18 09:22:06]: Hello there!
ENTER TEXT: i am server
CLIENT [2025-11-18 09:22:30]: i am client here
ENTER TEXT: how is preparation going on?
CLIENT [2025-11-18 09:23:20]: Nice What about you?
ENTER TEXT: fine !
CLIENT [2025-11-18 09:24:20]: ok
ENTER TEXT: quit
PS D:\SEM 5\CH>
```

```
PS D:\SEM 5\CH> py client.py
-----
CLIENT
Hello there! msg from SERVER
ENTER TEXT: Hello there!
SERVER [2025-11-18 09:22:18]: i am server
ENTER TEXT: i am client here
SERVER [2025-11-18 09:23:04]: how is preparation going on?
ENTER TEXT: Nice What about you?
SERVER [2025-11-18 09:23:36]: fine !
ENTER TEXT: ok
SERVER [2025-11-18 09:24:27]: quit
PS D:\SEM 5\CH> quit
```

## Conclusion:

The socket-based chat application successfully enables real-time communication between the two departments of Albert Enterprise. Department 1 operates as the SERVER while Department 2 functions as the CLIENT, and all messages are sent with proper timestamps. The application also ensures that if either side sends the message “quit”, the connection is terminated safely for both ends.