

Quiz 1

Course: Computer Networks Lab

Topic: Socket Programming (TCP & UDP)

Duration: 1 Hour

Total Marks: 10

INSTRUCTIONS

1. This is a practical lab assessment
 2. You will write and execute socket programs during the lab session
 3. Test your programs and demonstrate them to the instructor
 4. Submit your source code files before leaving the lab
 5. You may refer to syntax documentation only
 6. Write clean, well-commented code
-

QUESTION 1: TCP ECHO SERVER & CLIENT (5 Marks)

Problem Statement:

Create a TCP-based client-server application where the client sends messages to the server and the server echoes them back.

Requirements:

SERVER PROGRAM (2.5 marks):

- Create a TCP server that listens on **port 8080**
- Accept a connection from one client
- Receive **THREE messages** from the client
- Echo each message back to the client
- Display each received message on the server console
- Close the connection after processing 3 messages

CLIENT PROGRAM (2.5 marks):

- Create a TCP client that connects to the server on **localhost:8080**

- Send the following three messages to the server:
 1. "Hello Server"
 2. "This is message 2"
 3. "Goodbye"
- Receive and display each echoed message
- Close the connection after all messages are sent

Expected Output:

Server Console:

```
Server listening on port 8080...
Client connected
Received: Hello Server
Received: This is message 2
Received: Goodbye
Connection closed
```

Client Console:

```
Connected to server
Sent: Hello Server | Received: Hello Server
Sent: This is message 2 | Received: This is message 2
Sent: Goodbye | Received: Goodbye
Connection closed
```

Deliverables:

- File 1: tcp_server.py
- File 2: tcp_client.py

QUESTION 2: UDP MESSAGE EXCHANGE (5 Marks)

Problem Statement:

Create a UDP-based client-server application where the client sends messages and the server responds with acknowledgments.

Requirements:

SERVER PROGRAM (2.5 marks):

- Create a UDP server that binds to **port 9090**
- Receive messages from the client
- For each message received:

- Display the message on server console
 - Send back an acknowledgment: "ACK: [original message]"
- Keep the server running continuously

CLIENT PROGRAM (2.5 marks):

- Create a UDP client
- Send **THREE** messages to the server at **localhost:9090**:
 1. "Message 1"
 2. "Message 2"
 3. "Message 3"
- Receive and display the acknowledgment for each message
- Close the socket after all messages are sent

Expected Output:

Server Console:

```
UDP Server running on port 9090...
Received from client: Message 1
Received from client: Message 2
Received from client: Message 3
```

Client Console:

```
Sending: Message 1
Received ACK: ACK: Message 1

Sending: Message 2
Received ACK: ACK: Message 2

Sending: Message 3
Received ACK: ACK: Message 3
```

Deliverables:

- File 1: `udp_server.py`
 - File 2: `udp_client.py`
-