Date: 09/07/2025

Lab Practical #06:

Study Client-Server Socket programming - TCP & UDP

Practical Assignment #06:

1. **Write a C/Java code for TCP Server-Client Socket Programming.**
2. **Write a C/Java code for UDP Server-Client Socket Programming.**

1. For TCP Server-Client:

TCP Server Program:

```java
package TCP;
import java.io.*;
import java.net.*;
public class Server {
    private Socket s = null;
    private ServerSocket ss = null;
    private DataInputStream in = null;
    public Server(int port) {
        try {
            ss = new ServerSocket(port);
            System.out.println("Server started");
            System.out.println("Waiting for a client ...");
            s = ss.accept();
            System.out.println("Client accepted");
            in = new DataInputStream(
                    new BufferedInputStream(s.getInputStream()));
            String m = "";
            while (!m.equals("Over")) {
                try {
                    m = in.readUTF();
                    System.out.println(m);
                } catch (IOException i) {
                    System.out.println(i);
                }
            }
            System.out.println("Closing connection");
            s.close();
            in.close();
        } catch (IOException i) {
            System.out.println(i);
        }
    }
```

Date: 09/07/2025

TCP Client Program:

```java
package TCP;
import java.io.*;
import java.net.*;
public class Client {
    private Socket s = null;
    private DataInputStream in = null;
    private DataOutputStream out = null;

    public Client(String addr, int port){
        try {
            s = new Socket(addr, port);
            System.out.println("Connected");

            // Takes input from terminal
            in = new DataInputStream(System.in);

            // Sends output to the socket
            out = new DataOutputStream(s.getOutputStream());
        }
        catch (UnknownHostException u) {
            System.out.println(u);
            return;
        }
        catch (IOException i) {
            System.out.println(i);
            return;
        }
        String m = "";
        while (!m.equals("Over")) {
            try {
                m = in.readLine();
                out.writeUTF(m);
            }
            catch (IOException i) {
                System.out.println(i);
            }
        }
        try {
            in.close();
            out.close();
            s.close();
        }
        catch (IOException i) {
            System.out.println(i);
        }
    }
```

Date: 09/07/2025

2. For UDP Server-Client:

UDP Server Program:

```java
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
public class udpBaseServer_2{
    public static void main(String[] args) throws IOException{
        DatagramSocket ds = new DatagramSocket(1234);
        byte[] receive = new byte[65535];
        DatagramPacket DpReceive = null;
        while (true){
            DpReceive = new DatagramPacket(receive, receive.length);
            ds.receive(DpReceive);
            System.out.println("Client:-" + data(receive));
            if (data(receive).toString().equals("bye")){
                System.out.println("Client sent bye.....EXITING");
                break;
            }
            receive = new byte[65535];
        }
    }
    public static StringBuilder data(byte[] a){
        if (a == null)
            return null;
        StringBuilder ret = new StringBuilder();
        int i = 0;
        while (a[i] != 0)
        {
            ret.append((char) a[i]);
            i++;
        }
        return ret;
    }
}
```

Date: 09/07/2025

UDP Client Program:

```java
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;
public class udpBaseClient_2{
    public static void main(String args[]) throws IOException{
        Scanner sc = new Scanner(System.in);
        DatagramSocket ds = new DatagramSocket();
        InetAddress ip = InetAddress.getLocalHost();
        byte buf[] = null;
        while (true){
            String inp = sc.nextLine();
            buf = inp.getBytes();
            DatagramPacket DpSend = new DatagramPacket(buf, buf.length, ip, 1234);
            ds.send(DpSend);
            if (inp.equals("bye"))
                break;
        }
    }
}
```