Date: 11/07/2025

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:

```
import java.io.*;
import java.net.*;

public class Server {
   public static void main(String args[]) throws IOException
   {
      ServerSocket serverSocket = new ServerSocket(9090);
      System.out.println("Server is running and waiting for client connection...");

      Socket clientSocket = serverSocket.accept();
      System.out.println("Client connected!");

BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
      PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

      String message = in.readLine();
      System.out.println("Client says: " + message);

      out.println("Message received by the server.");
      clientSocket.close();      serverSocket.close();
   }
}
```

Date: 11/07/2025

TCP Client Program:

```
import java.io.*;
import java.net.*;

public class Client {
   public static void main(String args[]) throws IOException
   {
      Socket socket = new Socket("localhost", 9090);

      PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

      BufferedReader in = new BufferedReader(new
      InputStreamReader(socket.getInputStream()));

      out.println("Hello from client!");

      String response = in.readLine();
      System.out.println("Server says: " + response);

      socket.close();
   }
}
```

Date: 11/07/2025

2. For UDP Server-Client:

UDP Server Program:

```
import java.net.*;

public class UDPServer {
   public static void main(String[] args) {
      try {

         DatagramSocket socket = new DatagramSocket(5000);
         System.out.println("UDP Server is running...");

         byte[] buffer = new byte[1024];

         while (true) {

            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            socket.receive(request);

            String clientMsg = new String(request.getData(), 0, request.getLength());
            System.out.println("Client: " + clientMsg);


            if (clientMsg.equalsIgnoreCase("bye")) {
               System.out.println("Server closing...");
               break;
            }

            String responseMsg = "Received: " + clientMsg;
            byte[] responseData = responseMsg.getBytes();

            DatagramPacket response = new DatagramPacket(
```

Date: 11/07/2025

```
                responseData,
                responseData.length,
                request.getAddress(),
                request.getPort()
            );

            socket.send(response);
        }

        socket.close();
    } catch (Exception e) {
        System.out.println("Server error: " + e.getMessage());
    }
  }
}
```

UDP Client Program:

```
import java.net.*;
import java.io.*;

public class UDPClient {
  public static void main(String[] args) {
    try {
      DatagramSocket socket = new DatagramSocket();
      InetAddress serverAddress = InetAddress.getByName("127.0.0.1");

      BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
      byte[] buffer = new byte[1024];

      String message;
      while (true) {
        System.out.print("You: ");
```

Date: 11/07/2025

```
            message = userInput.readLine();


            byte[] data = message.getBytes();
            DatagramPacket packet = new DatagramPacket(data, data.length, serverAddress,
5000);
            socket.send(packet);

            if (message.equalsIgnoreCase("bye")) {
               break;
            }


            DatagramPacket response = new DatagramPacket(buffer, buffer.length);
            socket.receive(response);

            String responseMsg = new String(response.getData(), 0, response.getLength());
            System.out.println("Server: " + responseMsg);
         }

         socket.close();
         System.out.println("Client disconnected.");
      } catch (Exception e) {
         System.out.println("Client error: " + e.getMessage());
      }
   }
}
```