Date: / /

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:

```
// TCP Server-side Programming
import java.net.*;
import java.io.*;

public class SocketServer {

    // Initialize socket and input stream
    private Socket s = null;
    private ServerSocket ss = null;
    private DataInputStream in = null;

    // Constructor with port
    public SocketServer(int port) {

        // Starts server and waits for a connection
        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");

// Takes input from the client socket
in = new DataInputStream(
    new BufferedInputStream(s.getInputStream()));

String m = "";

// Reads message from client until "Over" is sent
while (!m.equals("Over"))
{
    try
    {
        m = in.readUTF();
        System.out.println(m);

    }
    catch(IOException i)
    {
        System.out.println(i);
    }
}
System.out.println("Closing connection");

// Close connection
s.close();
```

Date: / /

```
            in.close();
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
    }

    public static void main(String args[])
    {
        SocketServer s = new SocketServer(5000);
    }
}
```

TCP Client Program:

```
// TCP  Client-side Programming
import java.io.*;
import java.net.*;

public class SocketClient {

    // Initialize socket and input/output streams
    private Socket s = null;
    private DataInputStream in = null;
    private DataOutputStream out = null;

    // Constructor to put IP address and port
    public SocketClient(String addr, int port)
    {
```

Date: / /

```
// Establish a connection
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 to read message from input
String m = "";

// Keep reading until "Over" is input
while (!m.equals("Over")) {
   try {
      m = in.readLine();
      out.writeUTF(m);
      }
      catch (IOException i) {
        System.out.println(i);
      }
    }

    // Close the connection
    try {
      in.close();
      out.close();
      s.close();
    }
    catch (IOException i) {
      System.out.println(i);
    }
  }

  public static void main(String[] args) {
    SocketClient c = new SocketClient("127.0.0.1", 5000);
  }
}
```

Date: / /

2. For UDP Server-Client:

UDP Server Program:

```
// UDPServer.java

import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class UDPServer {
  public static void main(String[] args) {
    final int PORT = 12345;
    byte[] buffer = new byte[1024];

    try (DatagramSocket serverSocket = new DatagramSocket(PORT)) {
      System.out.println("UDP Server is running on port " + PORT);

      while (true) {
        // Receive packet
        DatagramPacket request = new DatagramPacket(buffer, buffer.length);
        serverSocket.receive(request);

        String clientMessage = new String(request.getData(), 0, request.getLength());
        System.out.println("Received from client: " + clientMessage);

        // Prepare response
        String responseMessage = "Hello Client, I received your message: " + clientMessage;
        byte[] responseData = responseMessage.getBytes();
```

Date: / /

```
                // Send response
                DatagramPacket response = new DatagramPacket(
                    responseData,
                    responseData.length,
                    request.getAddress(),
                    request.getPort()
                );
                serverSocket.send(response);
              }
            } catch (Exception e) {
              e.printStackTrace();
            }
          }
        }
```

UDP Client Program:

```
// UDPClient.java

import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class UDPClient {
  public static void main(String[] args) {
    final String SERVER_ADDRESS = "localhost";
    final int SERVER_PORT = 12345;

    try (DatagramSocket clientSocket = new DatagramSocket()) {
            // Message to send
            String message = "Hello Server, this is Client!";
            byte[] sendData = message.getBytes();

            // Send packet
            InetAddress serverIP = InetAddress.getByName(SERVER_ADDRESS);
            DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, serverIP, SERVER_PORT);
            clientSocket.send(sendPacket);

            // Receive response
            byte[] receiveBuffer = new byte[1024];
            DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);
            clientSocket.receive(receivePacket);

            String serverReply = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("Received from server: " + serverReply);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```