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:

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

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)
        {
```

Date: 09 / 07 /2025

```
                System.out.println(i);
            }
        }
        System.out.println("Closing connection");

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

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

TCP Client Program:

```
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");

      in = new DataInputStream(System.in);

      out = new DataOutputStream(s.getOutputStream());
    }
    catch (UnknownHostException u) {
      System.out.println(u);
      return;
    }
    catch (IOException i) {
      System.out.println(i);
      return;
```

Date: 09 / 07 /2025

```
        }

        String m = "";

        while (!m.equals("Over")) {
            try {
                out.writeUTF(m);
            }
            catch (IOException i) {
                System.out.println(i);
            }
        }

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

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

2. For UDP Server-Client:

UDP Server Program:

```
package UDP;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class Server
{
    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;
  }
}
```

UDP Client Program:

```
package UDP;


import java.util.*;

import java.io.*;

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;


public class Client

{
    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;
        }
    }
}
```