**Lab Practical #07:**

Study Client-Server Socket programming - TCP & UDP

**Practical Assignment #07:**

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

**TCP Server Program:**

**import** **java**.**io**.**BufferedInputStream**;

**import** **java**.**io**.**DataInputStream**;

**import** **java**.**net**.**\***;

**import** **java**.**io**.**\***;

**public** **class** Server {

**private** **Socket** socket = **null**;

**private** **ServerSocket** server = **null**;

**private** **DataInputStream** in = **null**;

**public** Server(**int** port) {

**try** {

            server = **new** ServerSocket(port);

            System.out.println("Server started...");

            System.out.println("Waiting for a client...");

            socket = server.accept();

            System.out.println("Client accepted");

            in = **new** DataInputStream(**new** BufferedInputStream(socket.getInputStream()));

**String** line = "";

**while** (!line.equals("Over")) {

**try** {

                    line = in.readUTF();

                    System.out.println(line);

                } **catch** (**IOException** e) {

                    System.out.println(e);

                }

            }

            System.out.println("Closing connection");

            socket.close();

            in.close();

        } **catch** (**IOException** e) {

            System.out.println(e);

        }

    }

**public** **static** **void** main(**String** args[]) {

**Server** server = **new** Server(5000);

    }

}

**TCP Client Program:**

**import** **java**.**net**.**\***;

**import** **java**.**io**.**\***;

**public** **class** Client {

**private** **Socket** socket = **null**;

**private** **BufferedReader** input = **null**;

**private** **DataOutputStream** out = **null**;

**public** Client(**String** address, **int** port) {

**try** {

            socket = **new** Socket(address, port);

            System.out.println("Connnected");

            input = **new** BufferedReader(**new** InputStreamReader(System.in));

            out = **new** DataOutputStream(socket.getOutputStream());

        } **catch** (**UnknownHostException** e) {

            System.out.println("unknownHost :: " + e);

        } **catch** (**IOException** e) {

            System.out.println("ioException :: " + e);

        }

**String** line = "";

**while** (!line.equals("Over")) {

**try** {

                line = input.readLine();

                out.writeUTF(line);

            } **catch** (**IOException** e) {

                System.out.println("ioException :: " + e);

            }

        }

**try** {

            input.close();

            out.close();

            socket.close();

        } **catch** (**IOException** e) {

            System.out.println("ioException :: " + e);

        }

    }

**public** **static** **void** main(**String** args[]) {

**Client** client = **new** Client("127.0.0.1", 5000);

    }

}

1. **For UDP Server-Client:**

**UDP Server Program:**

**import** **java**.**io**.**IOException**;

**import** **java**.**net**.**DatagramPacket**;

**import** **java**.**net**.**DatagramSocket**;

**import** **java**.**net**.**InetAddress**;

**import** **java**.**net**.**SocketException**;

**public** **class** Server {

**public** **static** **void** main(**String**[] args) **throws** **IOException** {

**DatagramSocket** ds = **new** DatagramSocket(1234);

**byte**[] receive = **new** **byte**[65535];

**boolean** flag = **true**;

**DatagramPacket** DpReceive = **null**;

        System.out.println("Server started...");

        System.out.println("Waiting for a client...");

**while** (**true**) {

            DpReceive = **new** DatagramPacket(receive, receive.length);

            ds.receive(DpReceive);

**if** (flag) {

                System.out.println("Client accepted");

                flag = **false**;

            }

            System.out.println("Client:-" + data(receive));

**if** (data(receive).toString().equals("Over")) {

                System.out.println("Closing connection");

**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:**

**import** **java**.**io**.**IOException**;

**import** **java**.**net**.**DatagramPacket**;

**import** **java**.**net**.**DatagramSocket**;

**import** **java**.**net**.**InetAddress**;

**import** **java**.**util**.**Scanner**;

**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("Over"))

**break**;

        }

    }

}