Experiment. NO: Date :	2. Study of Socket programming and client-server model.

Aim : To implement client-server model using socket programming.

Algorithm:

client : 1. create a client socket & connect it to the servers port number.
2. Retrieve its own IP address using builtin function
3. send its address to the server.
4. Display the connected message & give the message to be transmitted.
5. close the streams and sockets
6. stop.

sender :

```
import java.net.*;
import java.io.*;
public class Client{
    private Socket  skt = null;
    DataInputStream input = null;
    DataOutputStream  out = null;
    public client(String address, int port){
        try{
            skt = new Socket(address, port);
        System.out.println("connected");
        input = new DataInputStream(System.in);
        out = new DataOutputStream(skt.getOutputStream());
       }catch(Exception e){}
       String line = "";
       while(!line.equals("over")){
          try{
              line = input.readLine(); out.writeUTF(line);
        }catch(Exception e){}
```

```
        finally{
            input.close(); out.close(); skt.close();
        }
    }
    public static void main(String a[]){
            Client c = new Client("127.0.0.1", 5000);
        }
    }
```

Receiver

```
import java.net.*;
import java.io.*;
public class Server{
    Socket  skt =null;
    ServerSocket ss = null;
    DataInputStream  in = null;
    public Server(int port){
      try{
        ss = new ServerSocket(port);
        System.out.println("server started In waiting for client..");
        skt = ss.accept();
        System.out.println("Client accepted");
        in = new DataInputStream(new BufferedInputStream(
                                    skt.getInputStream()));
        String line ="";
        while(!line.equals("Over")) {
            try{
                line =in.readUTF(); System.out.println(line);
            }catch(Exception e){}
        finally {
```

Client output:

```
connected
This is the message from client
Goodbye server
```

Server output:

```
Server Started
waiting for a client...
Client accepted
This is the message from client
Goodbye server
```

```
skt.close(); pn.close();
}
}catch(Exception e){}
public static void main(String a[]){
    Server s = new Server(5000);
}
}
```

Result: Thus the study of Socket programming and implementation of client-server model was done successfully.