**Experiment No. 3**

**Master.java**

import java.io.\*;

import java.util.\*;

import java.net.\*;

public class Master

{

public static void main(String args[])throws Exception

{

String send="", r="";

Socket MyClient = new Socket("192.168.0.106",25);

System.out.println("Connected as Master");

DataInputStream din=new DataInputStream(MyClient.getInputStream());

DataOutputStream dout = new DataOutputStream(MyClient.getOutputStream());

Scanner sc = new Scanner(System.in);

do

{

System.out.print("Message('close' to stop): ");

send = sc.nextLine();

dout.writeUTF(send);

dout.flush();

}while(!send.equals("stop"));

dout.close();

din.close();

MyClient.close();

}

}

**Slave.java**

import java.io.\*;

import java.util.\*;

import java.net.\*;

public class Slave

{

public static void main(String args[])throws Exception

{

String r="";

Socket MyClient = new Socket("192.168.0.106",25);

System.out.println("Connected as Slave");

DataInputStream din=new DataInputStream(MyClient.getInputStream());

do

{

r=din.readUTF();

System.out.println("Master says: " + r);

}while(!r.equals("stop"));

din.close();

MyClient.close();

}

}

**Server.java**

import java.util.\*;

import java.io.\*;

import java.net.\*;

public class Server {

static ArrayList<ClientHandler> clients;

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

//Server server = new Server();

ServerSocket MyServer = new ServerSocket(25);

clients = new ArrayList<ClientHandler>();

Socket ss = null;

Message msg = new Message();

int count = 0;

while(true) {

ss = null;

try {

ss = MyServer.accept();

DataInputStream din =new DataInputStream(ss.getInputStream());

DataOutputStream dout=new DataOutputStream(ss.getOutputStream());

ClientHandler chlr = new ClientHandler(ss, din, dout, msg);

Thread t = chlr;

if (count > 0)

clients.add(chlr);

count++;

//System.out.println(threads);

t.start();

}

catch(Exception E){

continue;

}

}

}

}

class Message{

String msg;

public void set\_msg(String msg){

this.msg = msg;

}

public void get\_msg(){

System.out.println("\nNEW GROUP MESSAGE: " + this.msg);

for(int i = 0; i < Server.clients.size(); i++){

try{

System.out.print("Client: " + Server.clients.get(i).ip + "; ");

Server.clients.get(i).out.writeUTF(this.msg);

Server.clients.get(i).out.flush();

}

catch(Exception e){

System.out.print(e);

}

}

}

}

class ClientHandler extends Thread{

DataInputStream in;

DataOutputStream out;

Socket socket;

int sum;

float res;

boolean conn;

Message msg;

String ip;

public ClientHandler(Socket s, DataInputStream din, DataOutputStream dout, Message msg) {

this.socket = s;

this.in = din;

this.out = dout;

this.conn = true;

this.msg = msg;

this.ip = (((InetSocketAddress) this.socket.getRemoteSocketAddress()).getAddress()).toString().replace("/","");

}

public void run(){

while(conn == true){

try{

String input = this.in.readUTF();

// System.out.println("From host " + this.ip + ':' + input);

// String msg = "From host " + this.ip + ':' + input;

this.msg.set\_msg(input);

this.msg.get\_msg();

}

catch(Exception E){

conn = false;

System.out.println(E);

}

}

closeConn();

}

public void closeConn(){

try{

this.out.close();

this.in.close();

this.socket.close();

}

catch(Exception E){

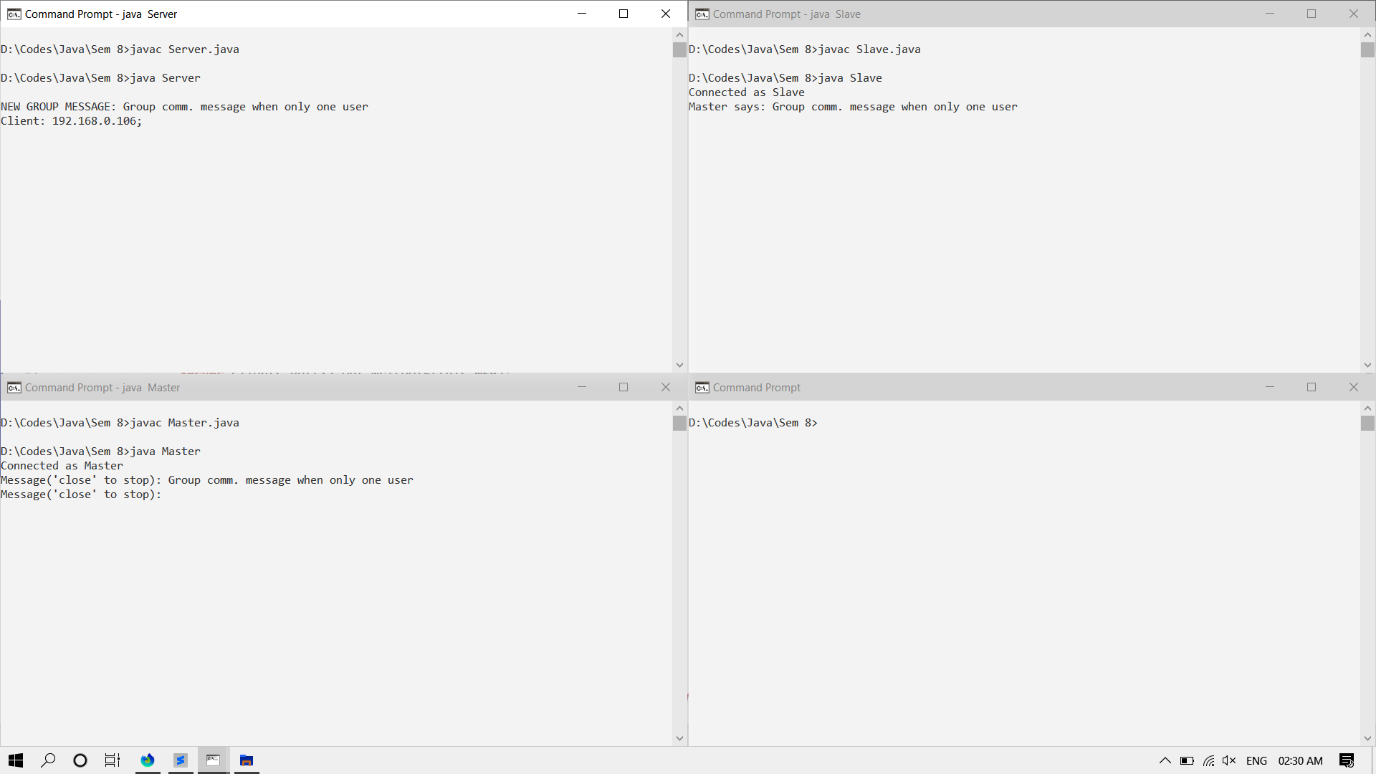
System.out.println(E);

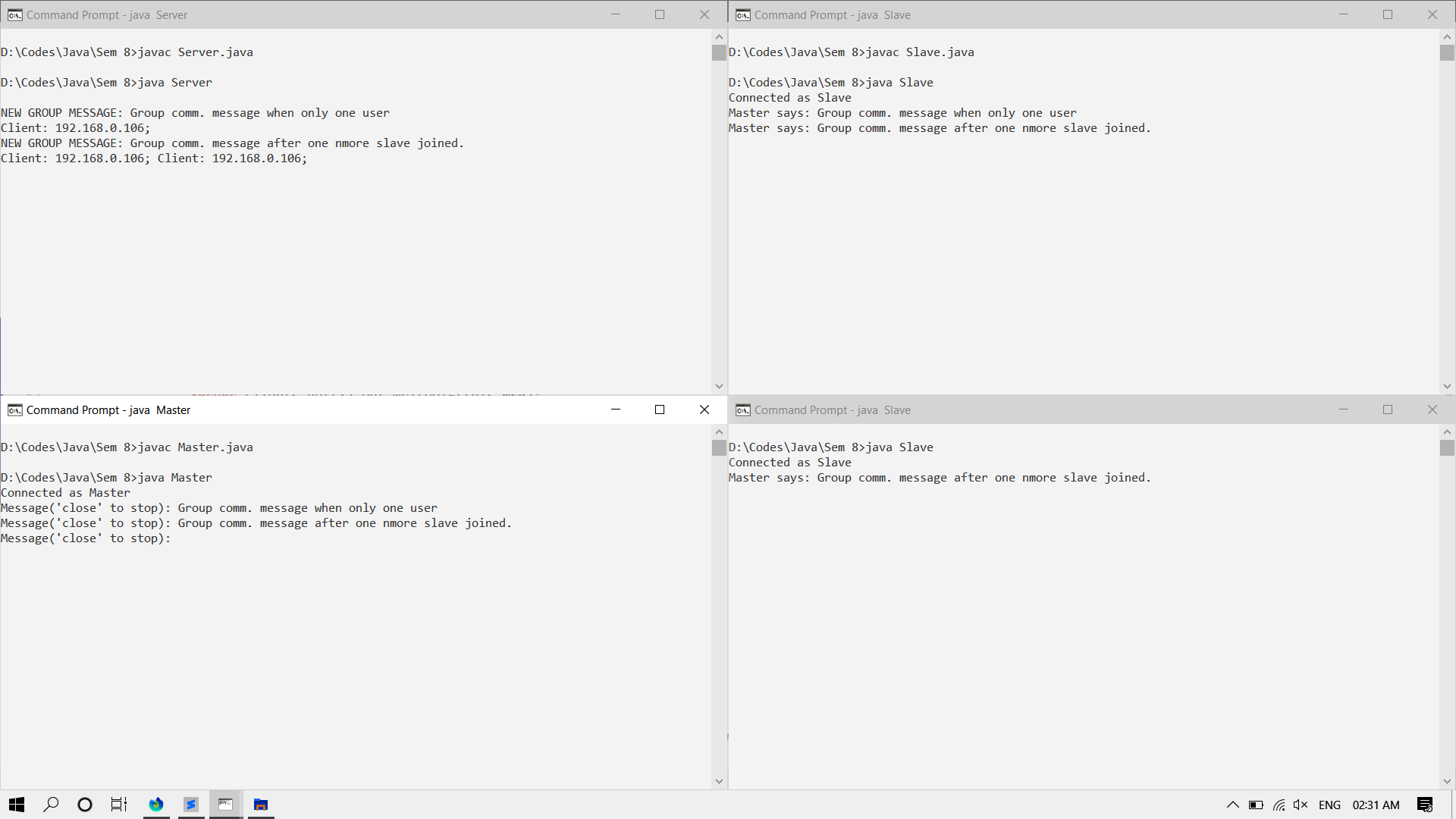
}

}

}

**Output**

****

****