Client/server

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

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

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

// initialize socket and input output streams

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

**private** DataInputStream input = **null**;

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

// constructor to put ip address and port

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

// establish a connection

**try** {

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

System.***out***.println("Connected");

// takes input from terminal

input = **new** DataInputStream(System.***in***);

// sends output to the socket

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

} **catch** (UnknownHostException u) {

System.***out***.println(u);

} **catch** (IOException i) {

System.***out***.println(i);

}

// string to read message from input

String line = "";

// keep reading until "Over" is input

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

**try** {

line = input.~~readLine~~();

out.writeUTF(line);

} **catch** (IOException i) {

System.***out***.println(i);

}

}

// close the connection

**try** {

input.close();

out.close();

socket.close();

} **catch** (IOException i) {

System.***out***.println(i);

}

}

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

Client client = **new** Client("172.16.0.138", 5000);

}

}

Server

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

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

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

//initialize socket and input stream

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

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

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

// constructor with port

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

// starts server and waits for a connection

**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");

// takes input from the client socket

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

String line = "";

// reads message from client until "Over" is sent

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

**try** {

line = in.readUTF();

System.***out***.println(line);

} **catch** (IOException i) {

System.***out***.println(i);

}

}

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

// close connection

socket.close();

in.close();

} **catch** (IOException i) {

System.***out***.println(i);

}

}

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

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

}

}

Java mail

**import** java.util.Properties;

**import** javax.mail.\*;

**import** javax.mail.internet.\*;

**class** Mailjava {

**public** **static** **void** send(**final** String from, **final** String password, String to, String sub, String msg) {

// Get properties object

Properties props = **new** Properties();

props.put("mail.smtp.host", "smtp.gmail.com");

props.put("mail.smtp.socketFactory.port", "465");

props.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");

props.put("mail.smtp.auth", "true");

props.put("mail.smtp.port", "465");

// get Session

Session session = Session.*getDefaultInstance*(props, **new** javax.mail.Authenticator() {

**protected** PasswordAuthentication getPasswordAuthentication() {

**return** **new** PasswordAuthentication(from, password);

}

});

// compose message

**try** {

MimeMessage message = **new** MimeMessage(session);

message.addRecipient(Message.RecipientType.***TO***, **new** InternetAddress(to));

message.setSubject(sub);

message.setText(msg);

// send message

Transport.*send*(message);

System.***out***.println("message sent successfully");

} **catch** (MessagingException e) {

**throw** **new** RuntimeException(e);

}

}

}

**class** SendMail { //

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

// from,password,to,subject,message

Mailjava.*send*("borraramya569@gmail.com", "ramya569", "kate.chaitanya96@gmail.com", "hello ",

"How r u?(MESSAGE)");

// change from, password and to

}

}

