Java Project

Chat Application

Server side:-

import java.net.\*;  
import java.io.\*;  
class Server {  
 ServerSocket server;  
 Socket socket;  
 BufferedReader br;  
 PrintWriter out;  
  
  
 //constructor..  
 public Server()  
 {  
 try{  
 server = new ServerSocket(7777);  
 System.*out*.println("Server is ready to accept the connection");  
 System.*out*.println("Waiting...");  
 socket=server.accept();  
  
 br=new BufferedReader(new InputStreamReader(socket.getInputStream()));  
  
 out=new PrintWriter(socket.getOutputStream());  
  
 startReading();  
 startWriting();  
  
 } catch (Exception e){  
 e.printStackTrace();  
 }  
  
 }  
  
 public void startReading()  
 {  
 // thread- read karka deta rahega  
 Runnable r1=()->{  
 System.*out*.println("reader started..");  
  
 try {  
 while(true)  
 {  
  
 String msg = br.readLine();  
 if (msg.equals("exit")) {  
 System.*out*.println("Client terminated the chat");  
 socket.close();  
 break;  
 }  
 System.*out*.println("Client :" + msg);  
  
  
 }  
 }catch(Exception e){  
 // e.printStackTrace();  
 System.*out*.println("connection closed");  
 }  
 };  
  
 new Thread(r1).start();  
  
 }  
  
 public void startWriting()  
 {  
 //thread- data user laga and then send karaga client ko  
 Runnable r2=()->{  
 System.*out*.println("Writer Started..");  
  
 try {  
 while ( !socket.isClosed()) {  
  
  
 BufferedReader br1 = new BufferedReader(new InputStreamReader(System.*in*));  
  
 String content = br1.readLine();  
  
  
 out.println(content);  
 out.flush();  
 if(content.equals("exit"))  
 {  
 socket.close();  
 break;  
 }  
  
  
 }  
 }catch(Exception e){  
 // e.printStackTrace();  
 System.*out*.println("connection closed");  
 }  
 System.*out*.println("connection closed");  
  
 };  
 new Thread(r2).start();  
  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("hi this server");  
 new Server();  
 }  
  
}

Client Side:-

import java.net.\*;  
import java.io.\*;  
  
public class Client {  
  
 Socket socket;  
 BufferedReader br;  
 PrintWriter out;  
  
 public Client()  
 {  
 try{  
 System.*out*.println("Sending request to server");  
  
 socket=new Socket("127.0.0.1",7777);  
 System.*out*.println("connection done. ");  
  
 br=new BufferedReader(new InputStreamReader(socket.getInputStream()));  
  
 out=new PrintWriter(socket.getOutputStream());  
  
 startReading();  
 startWriting();  
  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
 }  
 public void startReading()  
 {  
 // thread- read karka deta rahega  
 Runnable r1=()->{  
 System.*out*.println("reader started..");  
 try{  
 while(true)  
 {  
  
 String msg = br.readLine();  
 if (msg.equals("exit")) {  
 System.*out*.println("Server terminated the chat");  
 socket.close();  
  
 break;  
 }  
 System.*out*.println("Server :" + msg);  
 }  
  
 }catch(Exception e){  
 //e.printStackTrace();  
 System.*out*.println("connection closed");  
 }  
 };  
  
 new Thread(r1).start();  
  
 }  
  
  
 public void startWriting()  
 {  
 //thread- data user laga and then send karaga client ko  
 Runnable r2=()->{  
 System.*out*.println("Writer Started..");  
  
 try{  
 while ( !socket.isClosed())  
 {  
  
  
 BufferedReader br1=new BufferedReader(new InputStreamReader(System.*in*));  
  
 String content= br1.readLine();  
  
  
  
 out.println(content);  
 out.flush();  
 if(content.equals("exit"))  
 {  
 socket.close();  
 break;  
 }  
  
 }  
 System.*out*.println("connection closed");  
 }catch(Exception e){  
 e.printStackTrace();  
 }  
  
 };  
 new Thread(r2).start();  
  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("this is Client");  
 new Client();  
 }  
}