Untitled-3

```java
import java.io.*;
import java.net.*;
import java.util.*;

public class ChatServer {
    private static Set<ClientHandler> clientHandlers = new HashSet<>();

    public static void main(String[] args) {
        try (ServerSocket serverSocket = new ServerSocket(12345)) {
            System.out.println("Chat server started. Waiting for clients...");

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("New client connected.");
                ClientHandler clientHandler = new ClientHandler(clientSocket);
                clientHandlers.add(clientHandler);
                new Thread(clientHandler).start();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    static class ClientHandler implements Runnable {
        private Socket clientSocket;
        private PrintWriter out;
        private BufferedReader in;
        private String username;

        public ClientHandler(Socket socket) {
            this.clientSocket = socket;
        }

        public void run() {
            try {
                out = new PrintWriter(clientSocket.getOutputStream(), true);
                in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

                out.println("Enter your username:");
                username = in.readLine();
                broadcastMessage(username + " has joined the chat!", null);

                String message;
                while ((message = in.readLine()) != null) {
                    if (message.startsWith("@")) {
                        String[] tokens = message.split(" ", 2);
                        if (tokens.length == 2) {
                            sendPrivateMessage(tokens[0].substring(1), tokens[1]);
                        }
                    } else {
                        broadcastMessage(username + ": " + message, this);
                    }
                }
            } catch (IOException e) {
                e.printStackTrace();
            } finally {
                try {
                    clientHandlers.remove(this);
                    if (username != null) {
                        broadcastMessage(username + " has left the chat.", null);
                    }
                    clientSocket.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }

        private void broadcastMessage(String message, ClientHandler excludeClient) {
            for (ClientHandler client : clientHandlers) {
                if (client != excludeClient) {
                    client.out.println(message);
                }
            }
        }

        private void sendPrivateMessage(String recipient, String message) {
            for (ClientHandler client : clientHandlers) {
                if (client.username.equals(recipient)) {
                    client.out.println("[Private] " + username + ": " + message);
                    out.println("[Private to " + recipient + "]: " + message);
                    return;
                }
            }
            out.println("User " + recipient + " not found.");
        }
    }
}

```

Untitled-4

```java
import java.io.*;
import java.net.*;

public class ChatClient {
    private static final String SERVER_IP = "localhost"; // Replace with your server IP
    private static final int SERVER_PORT = 12345;

    public static void main(String[] args) {
        try (Socket socket = new Socket(SERVER_IP, SERVER_PORT);
             BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
             PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
             BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in))) {

            System.out.println("Connected to chat server.");
            new Thread(new IncomingMessagesHandler(in)).start();

            String userInput;
            while ((userInput = stdIn.readLine()) != null) {
                out.println(userInput);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private static class IncomingMessagesHandler implements Runnable {
        private BufferedReader in;

        public IncomingMessagesHandler(BufferedReader in) {
            this.in = in;
        }

        public void run() {
            try {
                String incomingMessage;
                while ((incomingMessage = in.readLine()) != null) {
                    System.out.println(incomingMessage);
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```