Import java.io.\*;

Import java.net.\*;

Import java.util.\*;

Public class ChatServer {

Private static Set<PrintWriter> clientWriters = new HashSet<>();

Public static void main(String[] args) throws IOException {

Int port = 12345; // Define the port number

System.out.println(“Chat server is running on port “ + port);

Try (ServerSocket serverSocket = new ServerSocket(port)) {

While (true) {

New ClientHandler(serverSocket.accept()).start();

}

}

}

Private static class ClientHandler extends Thread {

Private Socket socket;

Private PrintWriter out;

Public ClientHandler(Socket socket) {

This.socket = socket;

}

Public void run() {

Try (BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()))) {

Out = new PrintWriter(socket.getOutputStream(), true);

Synchronized (clientWriters) {

clientWriters.add(out);

}

String message;

While ((message = in.readLine()) != null) {

System.out.println(“Received: “ + message);

Synchronized (clientWriters) {

For (PrintWriter writer : clientWriters) {

Writer.println(message);

}

}

}

} catch (IOException e) {

System.err.println(“Error handling client: “ + e.getMessage());

} finally {

Try {

Socket.close();

} catch (IOException e) {

System.err.println(“Failed to close socket: “ + e.getMessage());

}

Synchronized (clientWriters) {

clientWriters.remove(out);

}

System.out.println(“Client disconnected.”);

}

}

}

}

Client code

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class ChatClient {

public static void main(String[] args) {

String serverAddress = "localhost"; // Change this to the server's IP address if needed

int port = 12345;

try (Socket socket = new Socket(serverAddress, port);

BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

Scanner scanner = new Scanner(System.in)) {

System.out.println("Connected to the chat server.");

// Thread to listen for incoming messages

Thread listener = new Thread(() -> {

try {

String message;

while ((message = in.readLine()) != null) {

System.out.println(message);

}

} catch (IOException e) {

System.err.println("Disconnected from server.");

}

});

listener.start();

// Main thread to send messages

while (scanner.hasNextLine()) {

String message = scanner.nextLine();

out.println(message);

}

} catch (IOException e) {

System.err.println("Error connecting to the server: " + e.getMessage());

}

}

}