Import java.io.\*;

Import java.net.\*;

Import java.util.\*;

Public class ChatServer {

Private static final int PORT = 1234;

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

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

ServerSocket serverSocket = new ServerSocket(PORT);

System.out.println(“Server started. Waiting for clients…”);

While (true) {

Socket clientSocket = serverSocket.accept();

System.out.println(“New client connected: “ + clientSocket);

New ClientHandler(clientSocket).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.out.println(“Error handling client: “ + e.getMessage());

} finally {

Try {

Socket.close();

} catch (IOException e) { }

Synchronized (clientWriters) {

clientWriters.remove(out);

}

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

}

}

}

}

import java.io.\*;

Import java.net.\*;

Public class ChatClient {

Private static final String SERVER\_IP = “localhost”;

Private static final int SERVER\_PORT = 1234;

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

Socket socket = new Socket(SERVER\_IP, SERVER\_PORT);

System.out.println(“Connected to chat server.”);

New ReadThread(socket).start();

New WriteThread(socket).start();

}

Static class ReadThread extends Thread {

Private BufferedReader in;

Public ReadThread(Socket socket) throws IOException {

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

}

Public void run() {

Try {

String response;

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

System.out.println(response);

}

} catch (IOException e) {

System.out.println(“Disconnected from server.”);

}

}

}

Static class WriteThread extends Thread {

Private PrintWriter out;

Private BufferedReader consoleInput;

Public WriteThread(Socket socket) throws IOException {

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

consoleInput = new BufferedReader(new InputStreamReader(System.in));

}

Public void run() {

Try {

String userInput;

While ((userInput = consoleInput.readLine()) != null) {

Out.println(userInput);

}

} catch (IOException e) {

System.out.println(“Error writing to server.”);

}

}

}

}