HelloServer

import java.io.IOException;  
import java.io.InputStream;  
import java.net.ServerSocket;  
import java.net.Socket;  
  
public class HelloServer {  
  
 public static void main(String[] args) {  
 try {  
 ServerSocket serverSocket = new ServerSocket(5556); // Choose any available port  
  
 System.*out*.println("Server is running and waiting for a connection...");  
  
 Socket clientSocket = serverSocket.accept(); // Wait for a client to connect  
 System.*out*.println("Client connected: " + clientSocket);  
  
 // Read data from the client  
 InputStream input = clientSocket.getInputStream();  
 byte[] buffer = new byte[1024];  
 int bytesRead = input.read(buffer);  
  
 if (bytesRead != -1) {  
 String receivedData = new String(buffer, 0, bytesRead);  
 System.*out*.println("Received data from client: " + receivedData);  
 }  
  
 // Close the connection  
 clientSocket.close();  
 serverSocket.close();  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

HelloClient

import java.io.IOException;  
import java.io.OutputStream;  
import java.net.Socket;  
  
public class HelloClient {  
  
 public static void main(String[] args) {  
 try {  
 Socket socket = new Socket("localhost", 5556); // Connect to the server  
  
 // Send data to the server  
 String sendData = "hello\_client";  
 OutputStream output = socket.getOutputStream();  
 output.write(sendData.getBytes());  
  
 // Close the connection  
 socket.close();  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

**class Server**

import java.io.IOException;  
import java.io.InputStream;  
import java.io.OutputStream;  
import java.net.ServerSocket;  
import java.net.Socket;  
import java.util.ArrayList;  
import java.util.List;  
  
public class Server {  
 private static List<ClientHandler> *clients* = new ArrayList<>();  
  
 public static void main(String[] args) {  
 try {  
 ServerSocket serverSocket = new ServerSocket(5555); // Choose any available port  
  
 System.*out*.println("Server is running and waiting for clients...");  
  
 while (true) {  
 Socket clientSocket = serverSocket.accept();  
 System.*out*.println("New client connected: " + clientSocket);  
  
 ClientHandler clientHandler = new ClientHandler(clientSocket);  
 *clients*.add(clientHandler);  
 new Thread(clientHandler).start();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 static class ClientHandler implements Runnable {  
 private Socket clientSocket;  
 private InputStream input;  
 private OutputStream output;  
  
 public ClientHandler(Socket socket) {  
 try {  
 this.clientSocket = socket;  
 this.input = socket.getInputStream();  
 this.output = socket.getOutputStream();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 @Override  
 public void run() {  
 try {  
 while (true) {  
 byte[] buffer = new byte[1024];  
 int bytesRead = input.read(buffer);  
  
 if (bytesRead == -1) {  
 break; // Client disconnected  
 }  
  
 String message = new String(buffer, 0, bytesRead);  
 System.*out*.println("Received message: " + message);  
  
 // Broadcast the message to all connected clients  
 broadcast(message);  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 try {  
 clientSocket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 private void broadcast(String message) throws IOException {  
 for (ClientHandler client : *clients*) {  
 if (client != this) {  
 client.output.write(message.getBytes());  
 }  
 }  
 }  
 }  
}

**class Client**

import java.io.IOException**;**import java.io.InputStream**;**import java.io.OutputStream**;**import java.net.Socket**;**import java.util.Scanner**;**public class Client {  
 public static void main(String[] args) {  
 try {  
 Socket socket = new Socket("localhost"**, 5555**)**;** // Connect to the server  
  
 OutputStream output = socket.getOutputStream()**;** InputStream input = socket.getInputStream()**;** Scanner scanner = new Scanner(System.*in*)**;** new Thread(() -> {  
 try {  
 while (true) {  
 byte[] buffer = new byte[**1024**]**;** int bytesRead = input.read(buffer)**;** if (bytesRead == -**1**) {  
 System.*out*.println("Server has disconnected.")**;** System.*exit*(**0**)**;** }  
  
 String message = new String(buffer**, 0,** bytesRead)**;** System.*out*.println("Received message: " + message)**;** }  
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
 }).start()**;** while (true) {  
 System.*out*.print("Enter a message: ")**;** String message = scanner.nextLine()**;** output.write(message.getBytes())**;** }  
 } catch (IOException e) {  
 e.printStackTrace()**;** }  
 }  
}