**Applied Distributed System**

**Coit 13229**

Student Name : Md Afsar Uddin Salman

Student Id : 12190848

**Table of Contents**

[1.0 Usage Guide 1](#_Toc161319281)

[1.1 Prerequisites 1](#_Toc161319282)

[1.2 Building and Running the Application 1](#_Toc161319283)

[2.0 TCP Application Testing Screenshots 2](#_Toc161319284)

[2.1 TCP Server Start 2](#_Toc161319285)

[2.2 TCP Client Start 2](#_Toc161319286)

[2.3 Member Registeration 3](#_Toc161319287)

[2.4 File Contents 6](#_Toc161319288)

[3.0 UDP Application Testing Screenshots 7](#_Toc161319289)

[3.1 UDP Server Start 7](#_Toc161319290)

[3.2 UDP Client Start 7](#_Toc161319291)

# Usage Guide

In this section, we will see the instructions on how to build and run the client server application that is made for the fitness club.

## Prerequisites

Before building and running the client-server application for the fitness club, ensure that you have the following prerequisites installed and configured:

* **Java Development Kit (JDK):** Ensure that you have JDK installed on your system. The application is written in Java and requires JDK to compile and run.
* **Access to the FitClubNetworkApp Repository:** You will need to have access to the FitClubNetworkApp repository containing the client-server application source code.

## Building and Running the Application

Follow these steps to build and run the client-server application:

* Download the <https://github.com/idg-salman/FitClubNetworkApp> repository from GitHub to your local machine.
* Extract the downloaded file to the folder of your choice.

The application has two parts; register member part and manager part – where each part is a complete client/server application.

To execute the register member part, follow these steps:

* Open the command prompt or Apache NetBeans IDE.
* Navigate to the directory containing the FitClubNetworkApp files.
* Compile the Java files using the command javac TCPServer.java TCPClient.java
* Run the TCP server by executing java TCPServer
* Run the TCP client by executing java TCPClient
* Follow the prompts to enter your membership details for registration.

To execute the manage part, follow these steps:

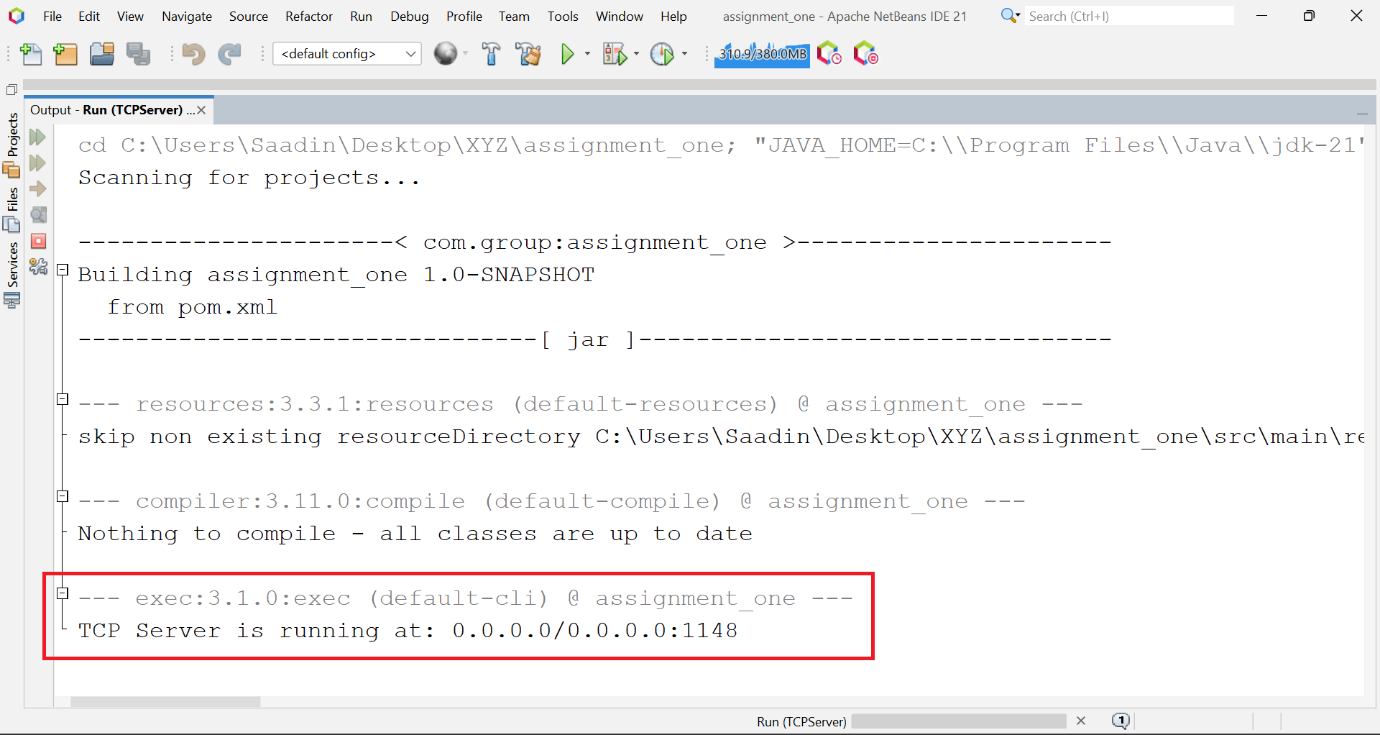
* Open the terminal or command prompt or Apache NetBeans IDE
* Navigate to the directory containing the FitClubNetworkApp files.
* Compile the Java files using the command javac UDPServer.java UDPClient.java
* Run the UDP server by executing java UDPServer
* Run the UDP client by executing java UDPClient memberlistObject

# TCP Application Testing Screenshots

## TCP Server Start

Before, we start the client application for registration, we should have server application running. To start the server application for registration, we need to run following commands:

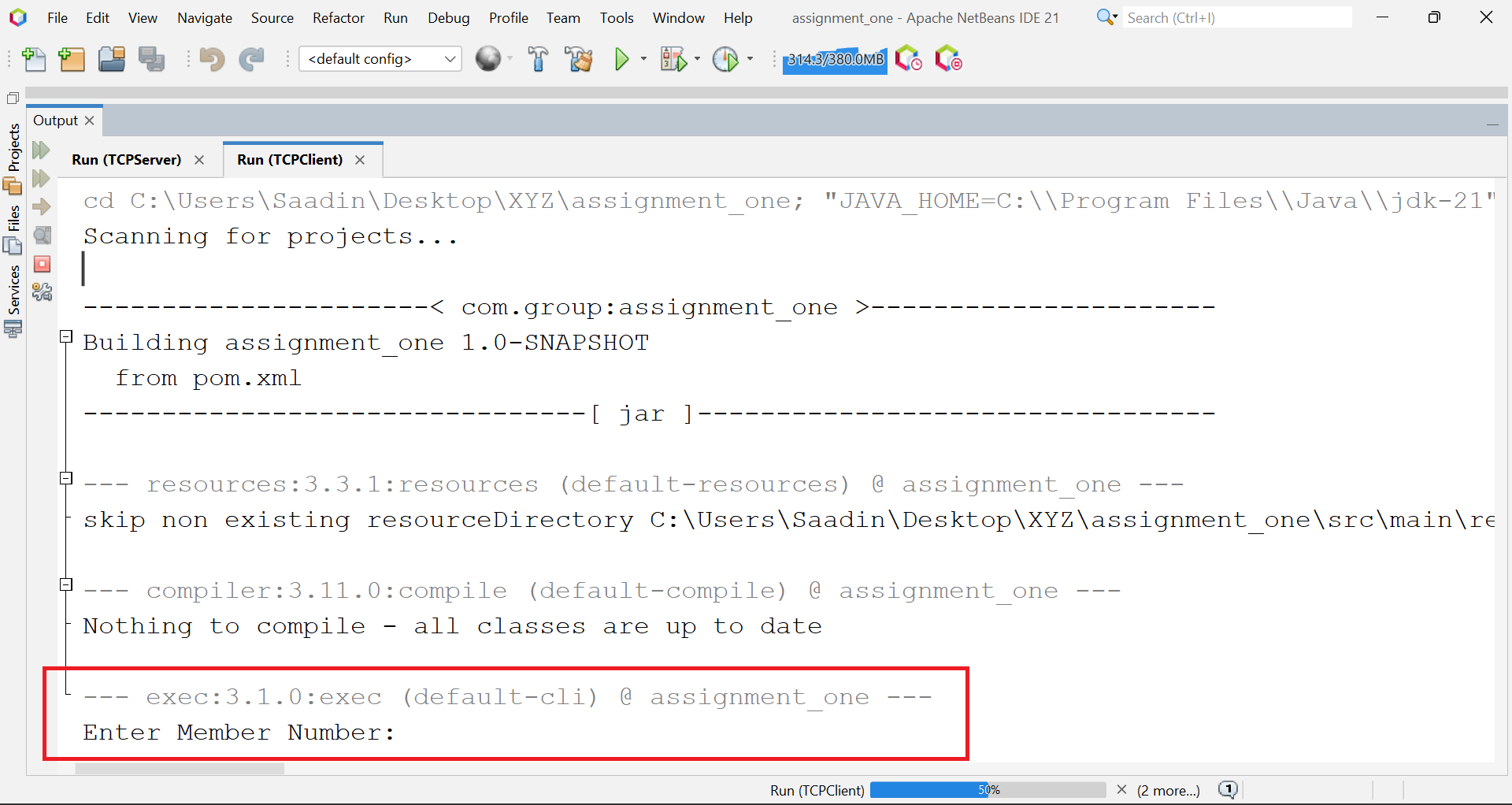
* javac TCPServer.java
* java TCPServer



## TCP Client Start

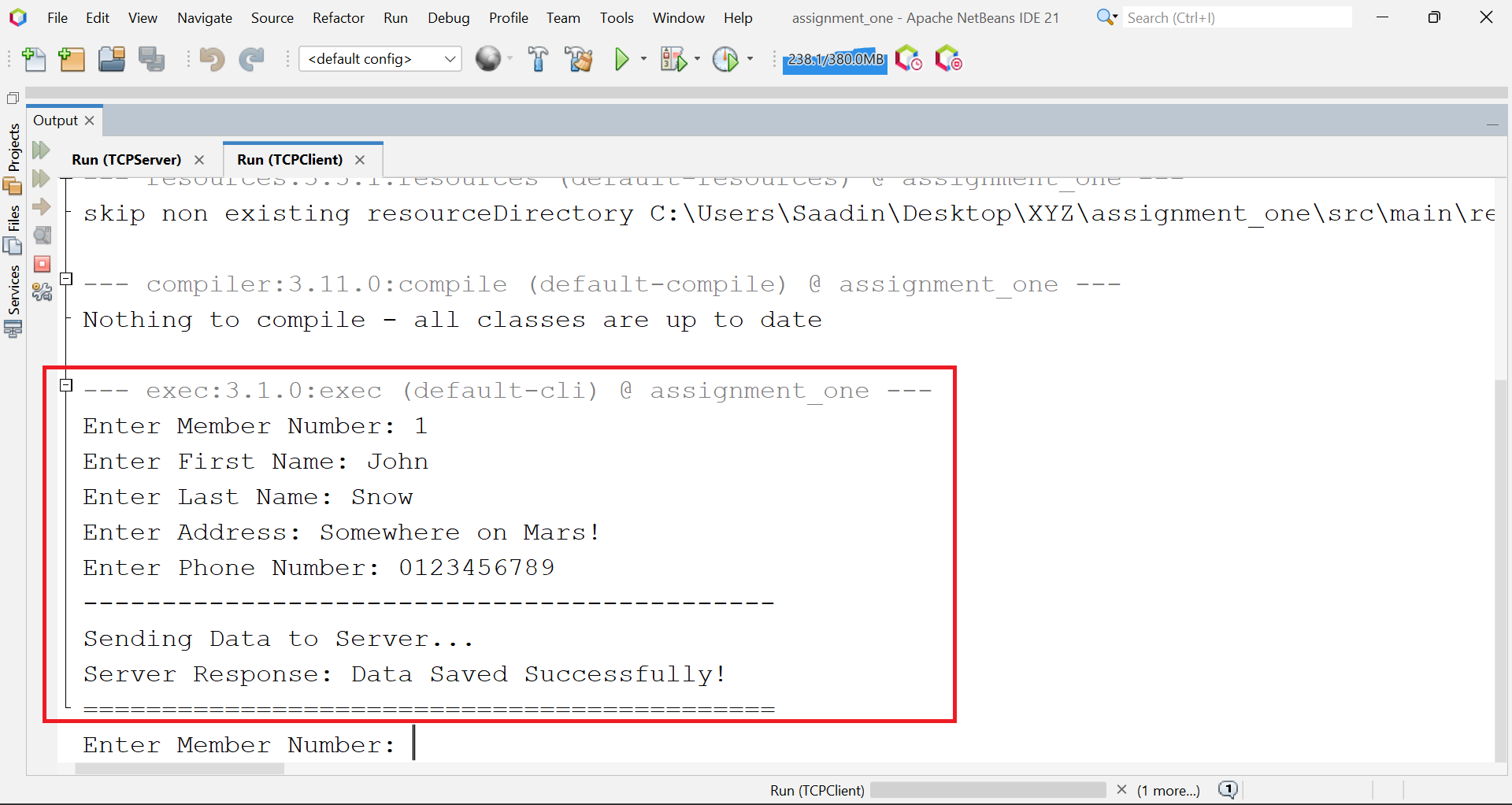
As our server application is running, now is the time to start the client application. To start the client application for registration, we need to run following commands:

* javac TCPClient.java
* java TCPClient

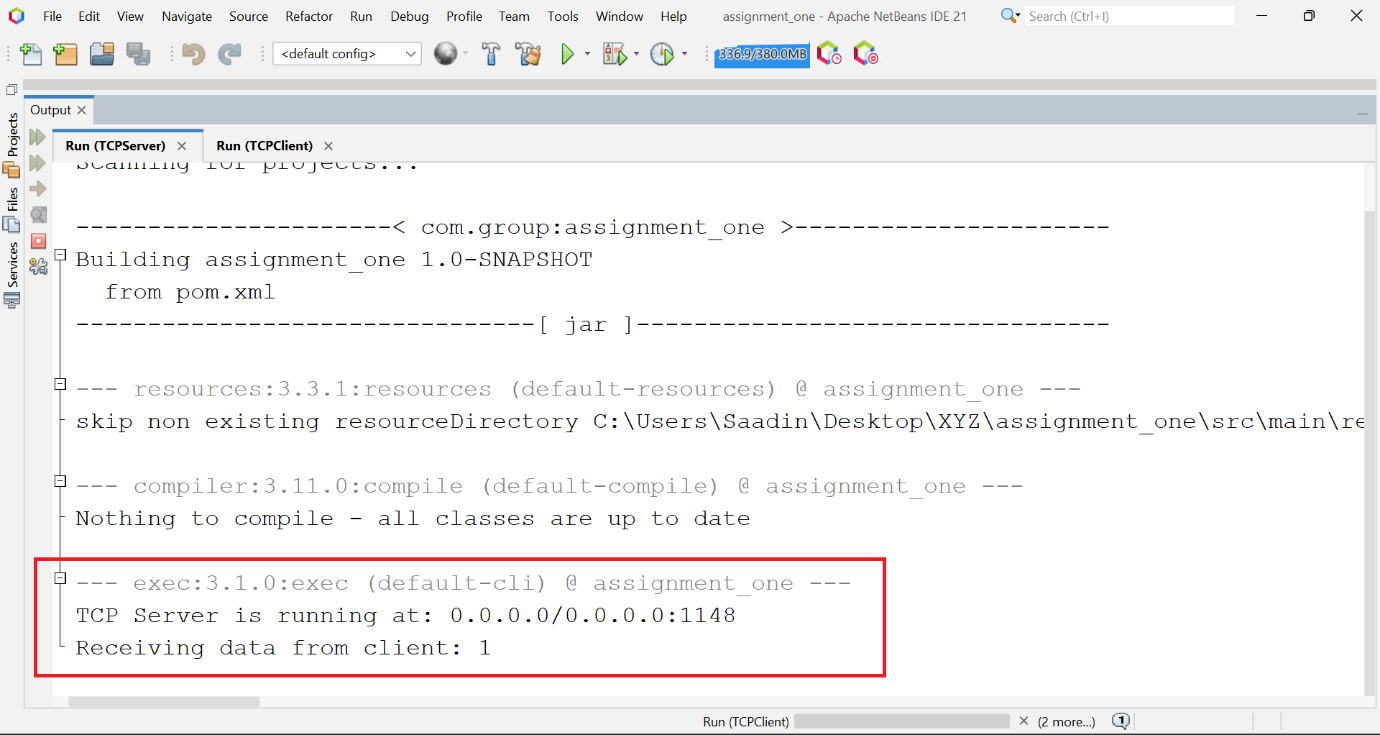


## Member Registeration

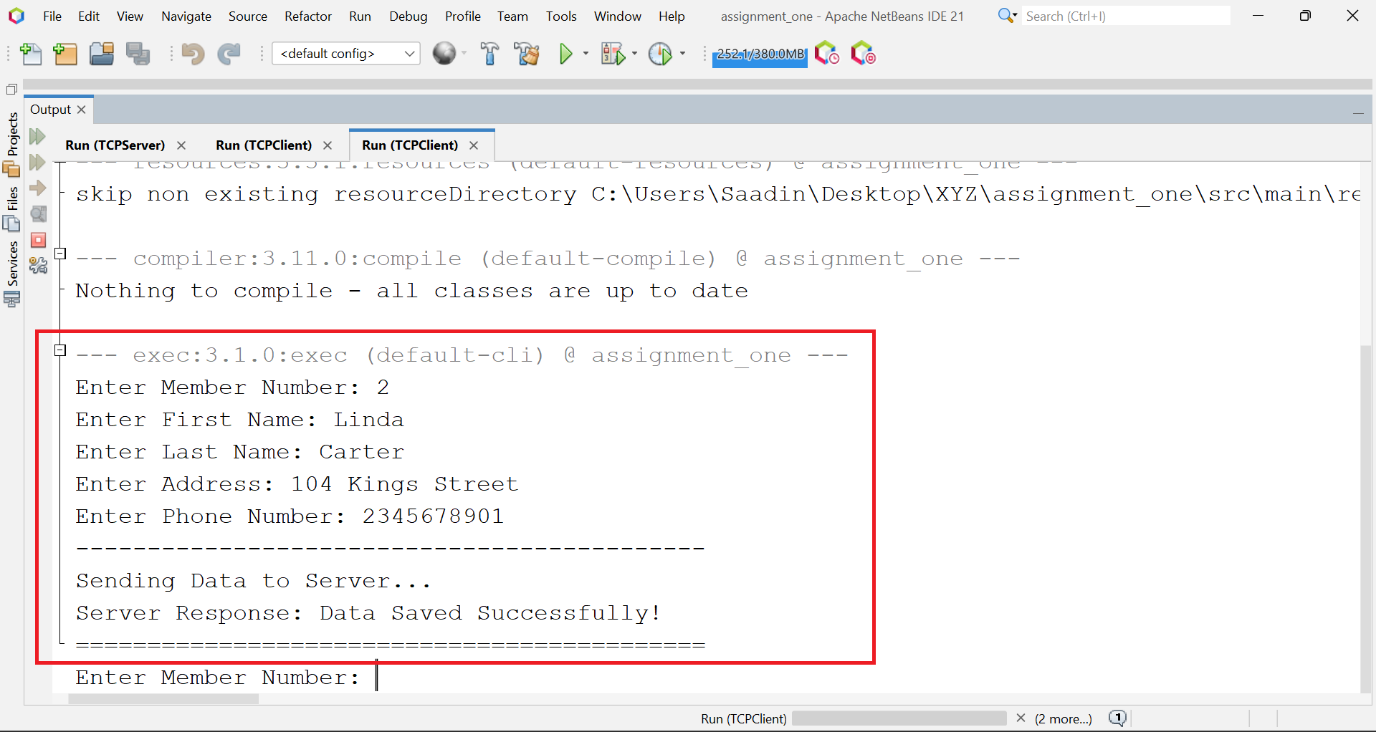
As our server and client applications are running, now we can start the registration process for the members. First of all using client application, we need to take the required member information from the client appliation, like this:



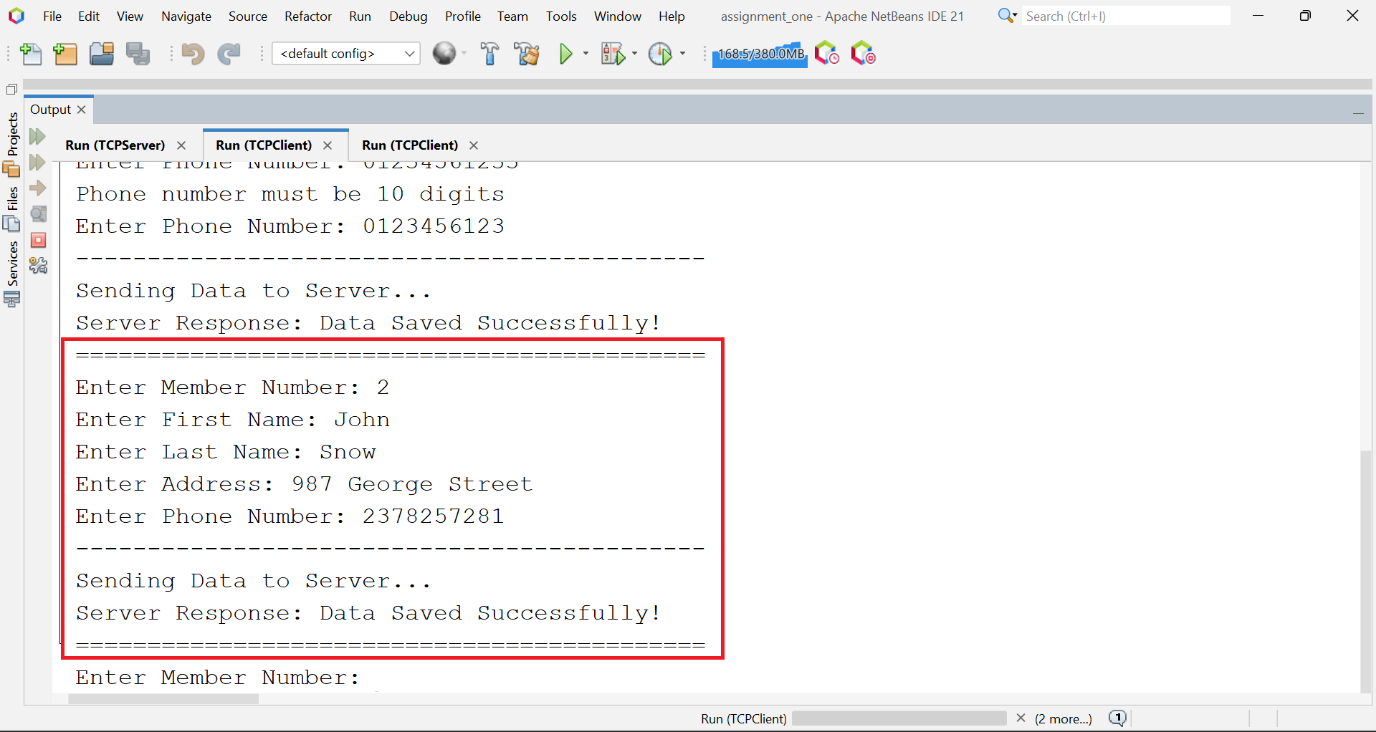
As we sent data from the client application to the server application for registration, an acknowledgement prompt was shown to the server console log – before the acknowledgement message being sent to the client.



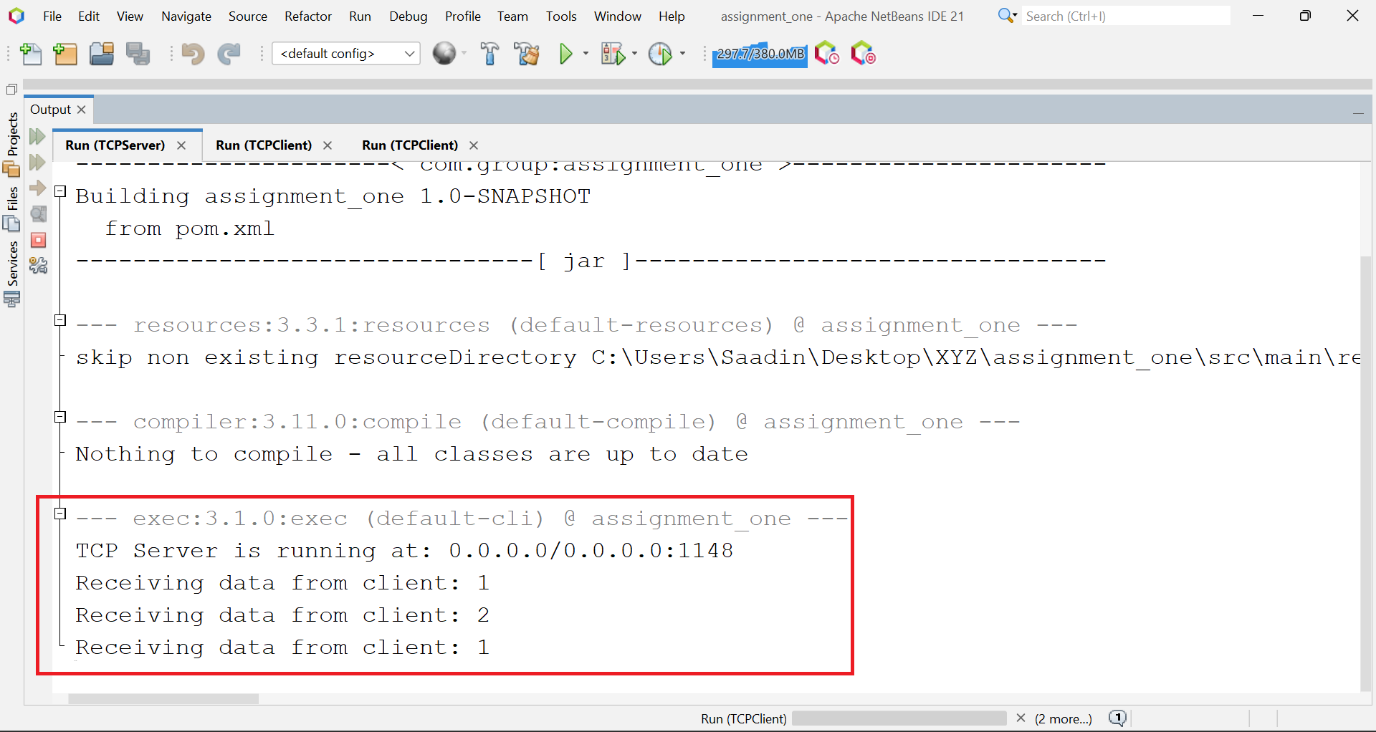
Now let’s see demonstrate that our server application can handle data coming from several client applications at a time. To do that, we will open another TCP client application and will start sending data from that application as well.



Also, I sent data for one more member registeration from the TCP Client 01.

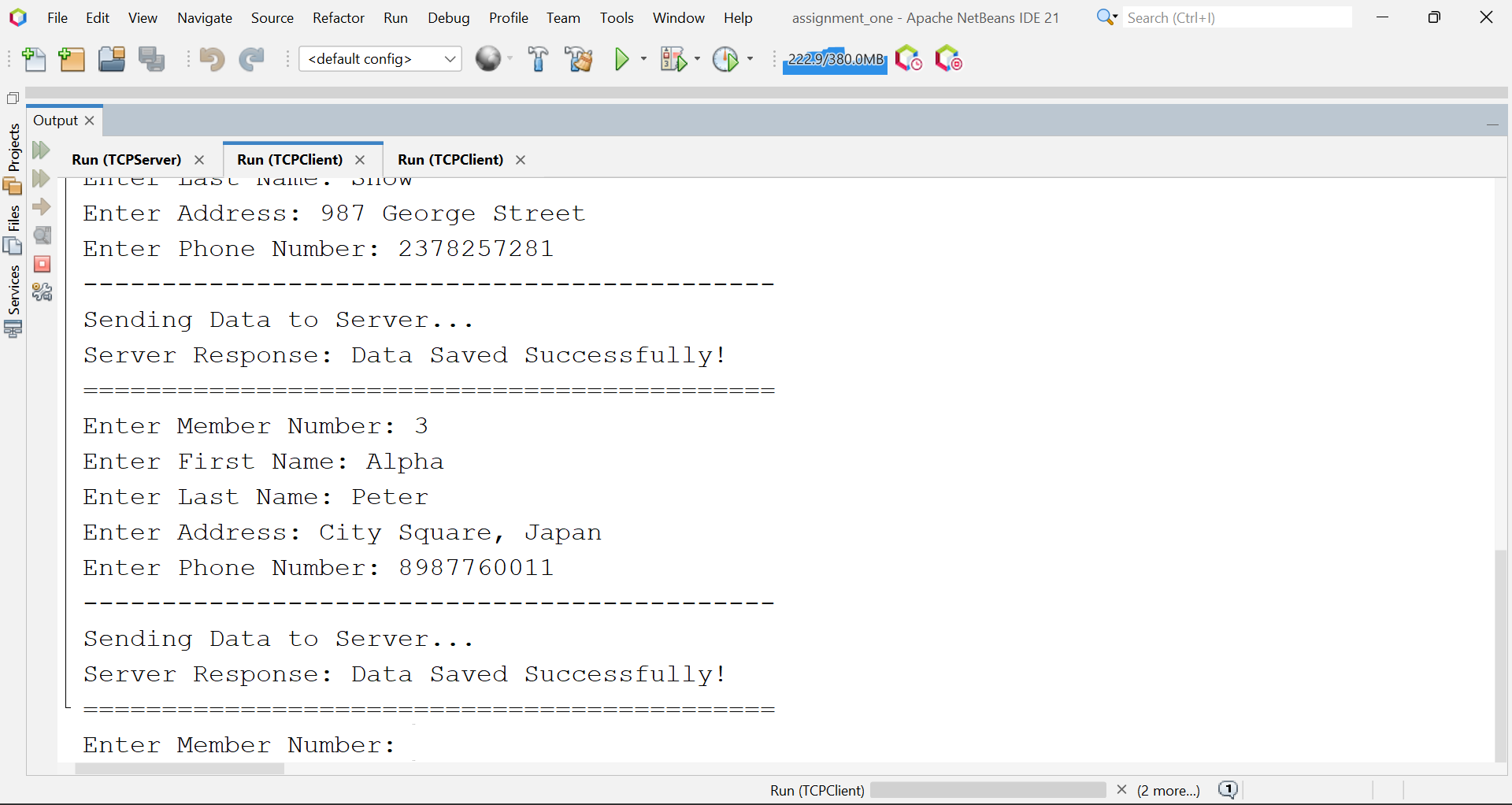


Below are the acknowledgment logs being shown on the server application’s console.



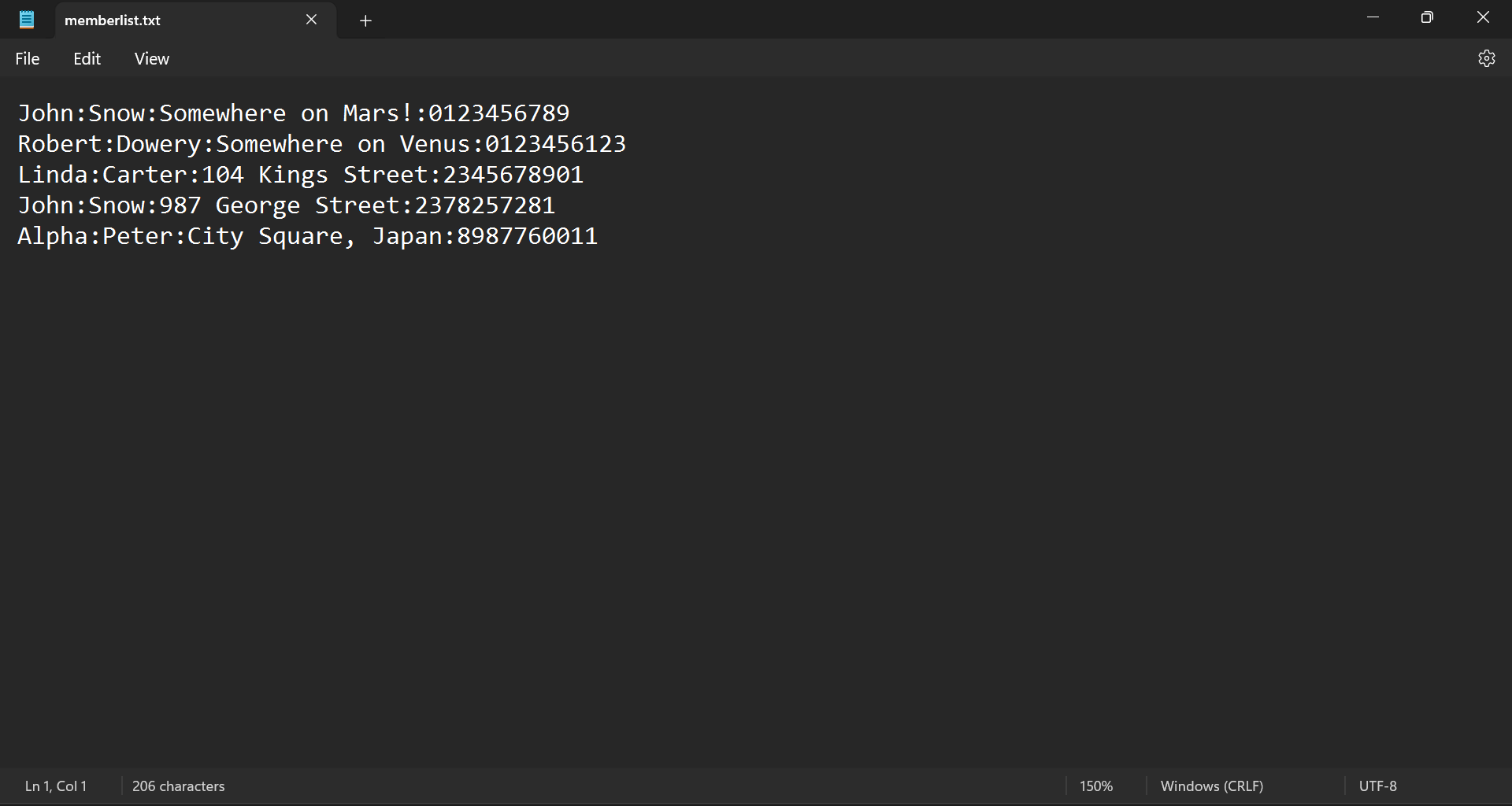
This thing proves that our implemented server application is to handle the requests coming from multiple clients simultaneously.

Let’s add one more member to the system.

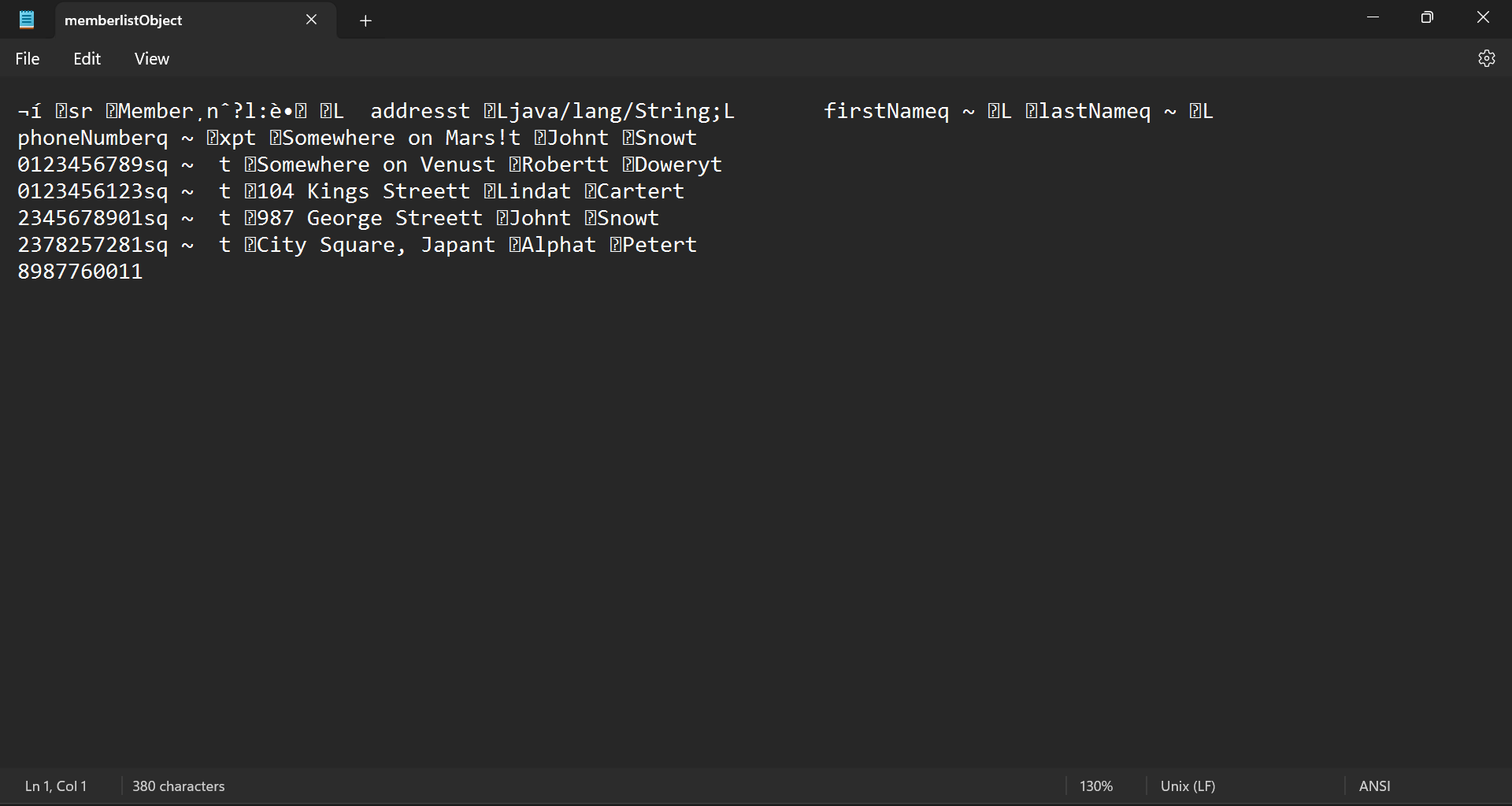


## File Contents

Below are the file contents from the memberlist.txt file, after performing the above activities.



Below are the file contents from memberlistObject file, after performing the above activities.

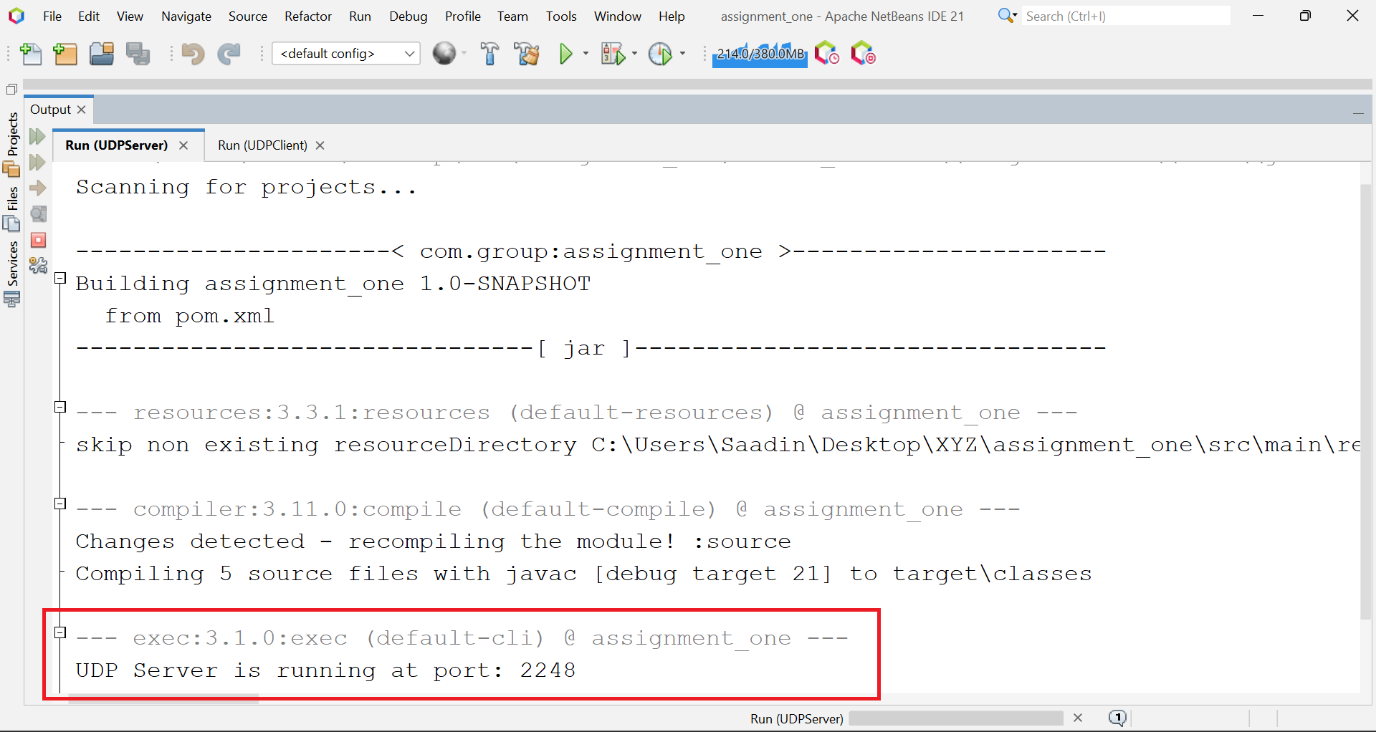


# UDP Application Testing Screenshots

## UDP Server Start

Before, we start the client application for manager, we should have server application running. To start the server application for manager, we need to run following commands:

* javac UDPServer.java
* java UDPServer



## UDP Client Start

As our server application is running, now is the time to start the client application. To start the client application for registration, we need to run following commands:

* javac UDPClient.java
* java UDPClient memberlistObject

