

# Distributed and Parallel Computing Lab

## CS461 Lab5

Name: Dipean Dasgupta

ID:202151188

### Task: Creation and Implementation of a Distributed Chat Application.

#### Server Side:

To receive incoming connections, the server code opens a socket. A new thread is started by the client when it connects in order to process its messages. The server broadcasts messages to every client that is connected and keeps track of client information on a map.

#### Code:

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

public class Server {
    private static final int PORT = 5000;
    private static Set<PrintWriter> clients = new HashSet<>();

    public static void main(String[] args) throws Exception {
        System.out.println("The chat server is running...");
        ServerSocket listener = new ServerSocket(PORT);
        try {
            while (true) {
                new Handler(listener.accept()).start();
            }
        } finally {
            listener.close();
        }
    }

    private static class Handler extends Thread {
        private Socket socket;
        private PrintWriter out;
        private BufferedReader in;

        public Handler(Socket socket) {
            this.socket = socket;
        }

        public void run() {
            try {
```



## Client Side:

The client component connects to the server and transmits the user ID. After then, it shows the messages it receives from the server while listening for them. Messages sent by the client to the server are also broadcast to any other clients that are connected.

### Code:

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

public class Client {
    private static final String SERVER_IP = "127.0.0.1";
    private static final int SERVER_PORT = 5000;

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

        System.out.print("Enter your name: ");
        String name = scanner.nextLine();
        out.println(name + " has joined the chat.");

        new Thread(new Runnable() {
            @Override
            public void run() {
                while (true) {
                    try {
                        String message = in.readLine();
                        System.out.println(message);
                    } catch (IOException e) {
                        e.printStackTrace();
                        break;
                    }
                }
            }
        }).start();
        while (true) {
            String message = scanner.nextLine();
            out.println(name + ": " + message);
        }
    }
}
```

In the client side,

`Socket` takes server id and server port as input for establishing connection with the server.

Each message is displayed with `sender: message` format.

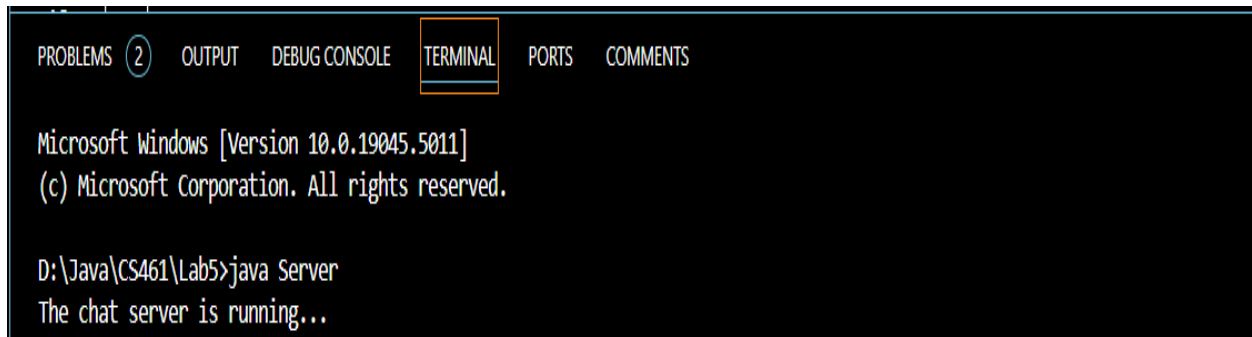
`getInputStream` read messages from the client and write responses back to the client using `getOutputStream`.

## Implementation of the chat Application:

In the first terminal the following command is run to initiate the files:

```
javac Server.java Client.java
```

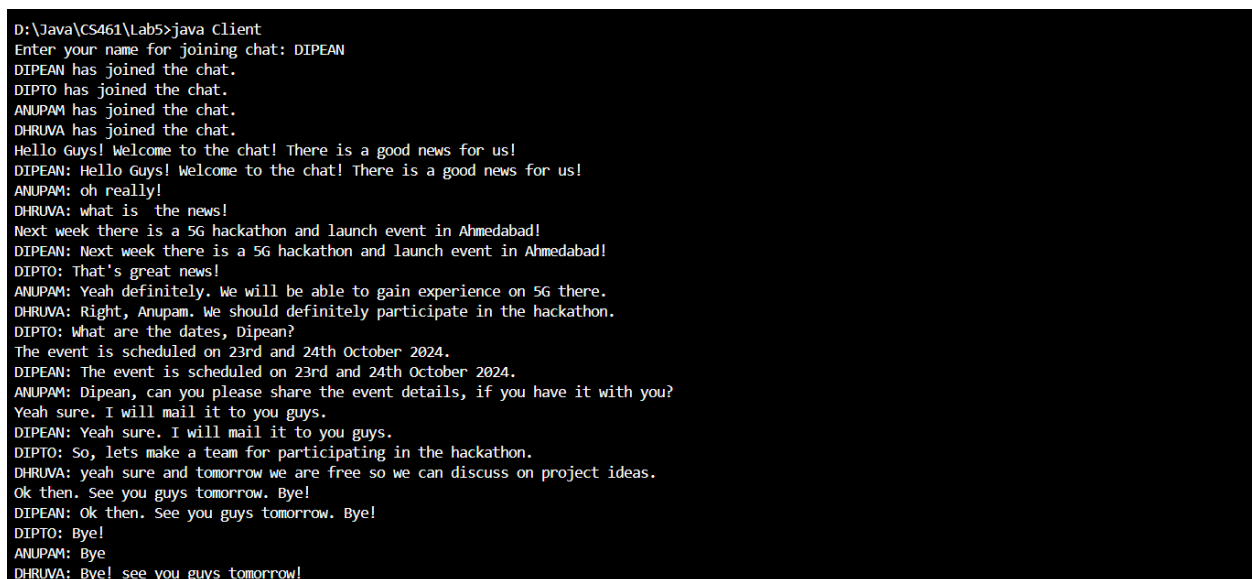
`java Server` command is executed to enable running of the chat server.



The screenshot shows a Windows terminal window with the title bar "Microsoft Windows [Version 10.0.19045.5011] (c) Microsoft Corporation. All rights reserved." The terminal is running the command `D:\Java\CS461\Lab5>java Server`, and the output is "The chat server is running...". The terminal window has tabs for "PROBLEMS (2)", "OUTPUT", "DEBUG CONSOLE", "TERMINAL" (which is selected), "PORTS", and "COMMENTS".

Now 4 clients are initiated so that a chat can be displayed.

### Client 1:



The screenshot shows a Windows terminal window with the title bar "D:\Java\CS461\Lab5>java Client". The terminal displays the following output: "Enter your name for joining chat: DIPEAN", "DIPEAN has joined the chat.", "DIPTO has joined the chat.", "ANUPAM has joined the chat.", "DHRUVA has joined the chat.", "Hello Guys! Welcome to the chat! There is a good news for us!", "DIPEAN: Hello Guys! Welcome to the chat! There is a good news for us!", "ANUPAM: oh really!", "DHRUVA: what is the news!", "Next week there is a 5G hackathon and launch event in Ahmedabad!", "DIPEAN: Next week there is a 5G hackathon and launch event in Ahmedabad!", "DIPTO: That's great news!", "ANUPAM: Yeah definitely. We will be able to gain experience on 5G there.", "DHRUVA: Right, Anupam. We should definitely participate in the hackathon.", "DIPTO: What are the dates, Dipeon?", "The event is scheduled on 23rd and 24th October 2024.", "DIPEAN: The event is scheduled on 23rd and 24th October 2024.", "ANUPAM: Dipeon, can you please share the event details, if you have it with you?", "Yeah sure. I will mail it to you guys.", "DIPEAN: Yeah sure. I will mail it to you guys.", "DIPTO: So, lets make a team for participating in the hackathon.", "DHRUVA: yeah sure and tomorrow we are free so we can discuss on project ideas.", "Ok then. See you guys tomorrow. Bye!", "DIPEAN: Ok then. See you guys tomorrow. Bye!", "DIPTO: Bye!", "ANUPAM: Bye", "DHRUVA: Bye! see you guys tomorrow!".

As the client is initiated, it first asks the client to write his name. As soon as another client joins it shows in the clients chat that new client has joined. Then both the clients exchange messages between each other.

After conversation client has left and connection with server ends for that particular client.

## Client 2:

```
D:\Java\CS461\Lab5>java Client
Enter your name for joining chat: DIPTO
DIPTO has joined the chat.
ANUPAM has joined the chat.
DHRUVA has joined the chat.
DIPEAN: Hello Guys! Welcome to the chat! There is a good news for us!
ANUPAM: oh really!
DHRUVA: what is the news!
DIPEAN: Next week there is a 5G hackathon and launch event in Ahmedabad!
That's great news!
DIPTO: That's great news!
ANUPAM: Yeah definitely. We will be able to gain experience on 5G there.
DHRUVA: Right, Anupam. We should definitely participate in the hackathon.
What are the dates, Dipeon?
DIPTO: What are the dates, Dipeon?
DIPEAN: The event is scheduled on 23rd and 24th October 2024.
ANUPAM: Dipeon, can you please share the event details, if you have it with you?
DIPEAN: Yeah sure. I will mail it to you guys.
So, lets make a team for participating in the hackathon.
DIPTO: So, lets make a team for participating in the hackathon.
DHRUVA: yeah sure and tomorrow we are free so we can discuss on project ideas.
DIPEAN: Ok then. See you guys tomorrow. Bye!
Bye!
DIPTO: Bye!
ANUPAM: Bye
DHRUVA: Bye! see you guys tomorrow!
□
```

## Client 3:

```
D:\Java\CS461\Lab5>java Client
Enter your name for joining chat: ANUPAM
ANUPAM has joined the chat.
DHRUVA has joined the chat.
DIPEAN: Hello Guys! Welcome to the chat! There is a good news for us!
oh really!
ANUPAM: oh really!
DHRUVA: what is the news!
DIPEAN: Next week there is a 5G hackathon and launch event in Ahmedabad!
DIPTO: That's great news!
Yeah definitely. We will be able to gain experience on 5G there.
ANUPAM: Yeah definitely. We will be able to gain experience on 5G there.
DHRUVA: Right, Anupam. We should definitely participate in the hackathon.
DIPTO: What are the dates, Dipeon?
DIPEAN: The event is scheduled on 23rd and 24th October 2024.
Dipeon, can you please share the event details, if you have it with you?
ANUPAM: Dipeon, can you please share the event details, if you have it with you?
DIPEAN: Yeah sure. I will mail it to you guys.
DIPTO: So, lets make a team for participating in the hackathon.
DHRUVA: yeah sure and tomorrow we are free so we can discuss on project ideas.
DIPEAN: Ok then. See you guys tomorrow. Bye!
DIPTO: Bye!
Bye
ANUPAM: Bye
DHRUVA: Bye! see you guys tomorrow!
□
```

## Client 4:

```
D:\Java\CS461\Lab5>java Client
Enter your name for joining chat: DHRUVA
DHRUVA has joined the chat.
DIPEAN: Hello Guys! Welcome to the chat! There is a good news for us!
ANUPAM: oh really!
what is the news!
DHRUVA: what is the news!
DIPEAN: Next week there is a 5G hackathon and launch event in Ahmedabad!
DIPTO: That's great news!
ANUPAM: Yeah definitely. We will be able to gain experience on 5G there.
Right, Anupam. We should definitely participate in the hackathon.
DHRUVA: Right, Anupam. We should definitely participate in the hackathon.
DIPTO: What are the dates, Dipeon?
DIPEAN: The event is scheduled on 23rd and 24th October 2024.
ANUPAM: Dipeon, can you please share the event details, if you have it with you?
DIPEAN: Yeah sure. I will mail it to you guys.
DIPTO: So, lets make a team for participating in the hackathon.
yeah sure and tomorrow we are free so we can discuss on project ideas.
DHRUVA: yeah sure and tomorrow we are free so we can discuss on project ideas.
DIPEAN: Ok then. See you guys tomorrow. Bye!
DIPTO: Bye!
ANUPAM: Bye
Bye! see you guys tomorrow!
DHRUVA: Bye! see you guys tomorrow!
```

So here we can see 4 clients can chat with each other though the server. The only issue is that for more clients all messages will be broadcast to all clients. So particular 2 people chat is not possible in this case. Also, the person who came at last will not be able to see previous messages.

## Terminating Connection:

```
D:\Java\CS461\Lab5>java Server
The chat server is running...
java.net.SocketException: Connection reset
java.net.SocketException: Connection reset
java.net.SocketException: Connection reset
java.net.SocketException: Connection reset
```

So, when all the 4 clients left and disconnected gradually then the **socket exception** is seen in the server side ensuring disconnection from client.