Green University of Bangladesh
Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Semester: (Spring, Year:2025), B.Sc. in CSE (Day)

Lab Report NO # 02
Course Title: Computer Networking Lab
Course Code: CSE-312 Section:223-D2

Lab Experiment Name: Implementation of a CLI-Based Group Chat Application Using Socket Programming and Threading.

Student Details

	Name	ID
1.	Hridoy Mia	223902010

Submission Date : 12/05/2025
Course Teacher's Name : Md. Sabbir Hosen Mamun

Lab Report Status
Marks: …… **Signature:.....**
Comments:..... **Date:.....**

Title: Implementation of a CLI-Based Group Chat Application Using Socket Programming and Threading.

Objectives:

The objectives of the application are:

- Create a CLI-based group chat app using Java.
- Use socket programming for client-server communication.
- Enable multiple clients using multithreading.
- Broadcast messages to all connected clients.
- Run smoothly in Apache NetBeans without errors.

Procedure:

- **Step 1:** Open Apache NetBeans and create a new Java project named `LabReport02-SocketThreading`.
- **Step 2:** In the `src` folder, create two Java classes: `ChatServer.java` and `ChatClient.java`.
- **Step 3:** Implement the `ChatServer` to accept multiple client connections using `ServerSocket` and handle each client with a separate thread.
- **Step 4:** Implement the `ChatClient` to connect to the server using `Socket`, send messages via one thread, and receive messages via another thread.
- **Step 5:** Start the `ChatServer` class to begin listening on a port (e.g., 12345).
- **Step 6:** Run multiple instances of the `ChatClient` class, enter user names, and exchange messages.
- **Step 7:** Verify that all connected clients receive every message sent in real time.
- **Step 8:** Type `exit` to close any client, ensuring the server handles disconnections gracefully.

Implementation:

1. ChatServer.java

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

public class ChatServer {
    private static final int PORT = 12345;
    private static Set<PrintWriter> clientWriters = new HashSet<>();

    public static void main(String[] args) throws IOException {
        System.out.println("Group Chat Server started...");
        ServerSocket serverSocket = new ServerSocket(PORT);

        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;
        private BufferedReader in;

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

        public void run() {
            try {
                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);
                    broadcastMessage(message);
```

```
            }
        } catch (IOException e) {
            System.out.println("Error handling client: " + e);
        } finally {
            try {
                socket.close();
            } catch (IOException e) {
                // ignore
            }
            synchronized (clientWriters) {
                clientWriters.remove(out);
            }
        }
    }

    private void broadcastMessage(String message) {
        synchronized (clientWriters) {
            for (PrintWriter writer : clientWriters) {
                writer.println(message);
            }
        }
    }
  }
}
```

2. ChatClient.java

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

public class ChatClient {
  private static final String SERVER_IP = "127.0.0.1"; // or use localhost
  private static final int PORT = 12345;

  public static void main(String[] args) throws IOException {
    Socket socket = new Socket(SERVER_IP, PORT);
    System.out.println("Connected to group chat!");

    new Thread(new MessageReceiver(socket)).start();
    new Thread(new MessageSender(socket)).start();
  }

  private static class MessageReceiver implements Runnable {
    private BufferedReader in;
```

```
        public MessageReceiver(Socket socket) throws IOException {
            in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        }

        public void run() {
            try {
                String message;
                while ((message = in.readLine()) != null) {
                    System.out.println("\n[New Message] " + message);
                }
            } catch (IOException e) {
                System.out.println("Connection closed.");
            }
        }
    }

    private static class MessageSender implements Runnable {
        private PrintWriter out;
        private BufferedReader consoleInput;

        public MessageSender(Socket socket) throws IOException {
            out = new PrintWriter(socket.getOutputStream(), true);
            consoleInput = new BufferedReader(new InputStreamReader(System.in));
        }

        public void run() {
            try {
                System.out.print("Enter your name: ");
                String name = consoleInput.readLine();
                String input;
                while (true) {
                    input = consoleInput.readLine();
                    if (input.equalsIgnoreCase("exit")) break;
                    out.println(name + ": " + input);
                }
                System.exit(0);
            } catch (IOException e) {
                System.out.println("Error sending message.");
            }
        }
    }
}
```

Output:

```
Output ×
Run (ChatServer) ×   Run (ChatClient) ×
cd C:\Users\ASUS\Documents\NetBeansProjects\Client; "JAVA_HOME=C:\\Program Files\\Java\\jdk-22" cmd /c "\"C:\\Pro
Scanning for projects...

------------------------< com.mycompany:Client >------------------------
Building Client 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Client ---
skip non existing resourceDirectory C:\Users\ASUS\Documents\NetBeansProjects\Client\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Client ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 22] to target\classes

--- exec:3.1.0:exec (default-cli) @ Client ---
Connected to group chat!
Enter your name:
[New Message] Enter your name:
Hridoy
Assalamo Alikom
Rakib: Hi Hridoy!
Rony : Hello Everyone!
```

```
Output ×
Run (ChatServer) ×   Run (ChatClient) ×
cd C:\Users\ASUS\Documents\NetBeansProjects\Server; "JAVA_HOME=C:\\Program Files\\Java\\jdk-22" cmd /c "\"C:\\Pro
Scanning for projects...

------------------------< com.mycompany:Server >------------------------
Building Server 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Server ---
skip non existing resourceDirectory C:\Users\ASUS\Documents\NetBeansProjects\Server\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Server ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Server ---
Starting Chat Server...
Server started on port 12345



Hridoy: Assalamo Alikom has joined.
Hridoy: Assalamo Alikom: Hridoy: Rakib: Hi Hridoy!
Hridoy: Assalamo Alikom: Hridoy: Rony : Hello Everyone!
```