

APPLICATION USING TCP SOCKETS: CHATS

PROGRAM:

Server.java

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

public class Server {

    public static void main(String[] args) {

        try {

            ServerSocket serverSocket = new ServerSocket(5000);

            System.out.println("🟢 Server started. Waiting for client...");

            Socket socket = serverSocket.accept();

            System.out.println("✅ Client connected.");

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

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

            String clientMsg;

            while ((clientMsg = in.readLine()) != null) {

                System.out.println("👤 Client: " + clientMsg);

                // Automatically respond

                if (clientMsg.equalsIgnoreCase("hi")) {

                    out.println("Hello from server!");

                } else if (clientMsg.equalsIgnoreCase("bye")) {

                    out.println("Goodbye!");

                    break;

                } else {

                    out.println("You said: " + clientMsg);

                }

            }

        }

    }

}
```

```

    }
    socket.close();
    serverSocket.close();
    System.out.println("🔴 Server stopped.");
} catch (IOException e) {
    System.out.println("Error: " + e.getMessage());
}
}
}

```

Client.java

```

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

public class Client {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 5000);
            System.out.println("🟢 Connected to server.");

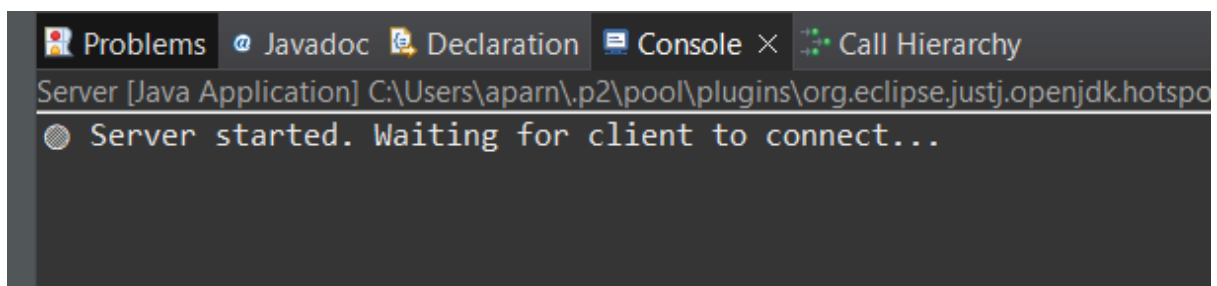
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
            BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));

            String input;
            while (true) {
                System.out.print("💬 You: ");

```

```
input = userInput.readLine();
out.println(input);
if (input.equalsIgnoreCase("bye")) {
    System.out.println("👋 Disconnected.");
    break;
}
String response = in.readLine();
System.out.println("👤 Server: " + response);
}
socket.close();
} catch (IOException e) {
    System.out.println("Error: " + e.getMessage());
}
}
}
```

OUTPUT:



The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for 'Problems', 'Javadoc', 'Declaration', 'Console', and 'Call Hierarchy'. The 'Console' tab is active, displaying the output of a Java application. The text in the console is: 'Server [Java Application] C:\Users\aparn\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspoc' followed by a new line and a progress indicator (a circle with a dot) and the text 'Server started. Waiting for client to connect...'. The console background is dark gray.

<terminated> Client [Java Application] C:\Users\aparn\.p2\pool\plugins\org.eclipse.justj.

● Connected to server.

💬 You: hi

👤 Server: Hello from server!

💬 You: how are you?

👤 Server: You said: how are you?

💬 You: bye

👤 Disconnected.