

```

package myChat;

/**
 *
 * @author Patrick Neubauer
 * @version 1.0 alpha
 *
 * Course: Distributed Systems
 * Project P1: Internet Chat System
 *
 * Team members:
 * Tenzin Dakpa, Patrick Neubauer
 *
 */
import java.net.*;
import java.io.*;

/* class that EXTENDS the standard Thread class of the Java Library,
extending it with server, socket and data stream functionality
*/
public class TCPServerThread extends Thread {

    // defining private properties
    private TCPServer server = null;
    private Socket socket = null;

    private DataInputStream streamIn = null;
    private DataOutputStream streamOut = null;

    private int ID = -1;

    /* constructor to create a TCPServerThread object
    *
    */
    public TCPServerThread(TCPServer givenServer, Socket givenSocket) {
        super(); // calling the constructor of the superclass Thread
        server = givenServer;
        socket = givenSocket;
        ID = socket.getPort();
    }

    /*
    * Return ID of the client thread.
    */
    public int getID() {
        return ID;
    }

    /* run server thread
    */
    public void run() {
        System.out.println("Server Thread with ID " + ID + " says: I am running.");
        while(true) {
            try {
                server.handle(ID, streamIn.readUTF());
            } catch (IOException ioe) {
                System.out.println("Server Thread: Error while reading form ID " + ID + ". \n Error: " + ioe.getMessage());
                server.remove(ID); // remove client from server
                stop();
            }
        }
    }
}

```

```
/*
 * Open buffered datastreams on socket
 */
public void open() throws IOException {
    streamIn = new DataInputStream(new BufferedInputStream(socket.getInputStream()));
    streamOut = new DataOutputStream(new BufferedOutputStream(socket.getOutputStream()));
}

/* close socket and datastreams
 */
public void close() throws IOException {
    if (socket != null) socket.close();
    if (streamIn != null) streamIn.close();
    if (streamOut != null) streamOut.close();
}

/* Sending message in writing to DataOutputStream
 */
public void send(String message) {
    try {
        streamOut.writeUTF(message); // UTF is a string encoding
        streamOut.flush(); // flush output stream
    } catch (IOException ioe) {
        System.out.println("Server Thread: Error while sending to ID " + ID + ".\n Error: " + ioe.getMessage());
        server.remove(ID); // remove client from server
        stop();
    }
}
}
```