

```
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 TCPClientThread extends Thread {

    // defining private properties
    private TCPClient client = null;
    private Socket socket = null;
    private DataInputStream streamIn = null;

    /* constructor that creates a new TCP client on a certain socket.
    */
    public TCPClientThread(TCPClient givenClient, Socket givenSocket) {
        client = givenClient;
        socket = givenSocket;
        open();
        start();
    }

    /* opening data input stream from the socket
    */
    public void open() {
        try {
            streamIn = new DataInputStream(socket.getInputStream());
        } catch (IOException ioe) {
            System.out.println("Error while opening input stream. Error: \n" + ioe);
            client.stop(); // stopping client caused by error
        }
    }

    /* closing data input stream from the socket
    */
    public void close() {
        try {
            if (streamIn != null) streamIn.close();
        } catch (IOException ioe) {
            System.out.println("Error while closing input stream. Error: \n" + ioe);
            client.stop(); // stopping client caused by error
        }
    }

    /* start running the client thread
    */
    public void run() {
        while (true) {
            try {
                // handle the data input stream from the client and read his message
            } catch (IOException ioe) {
                System.out.println("Error while reading input stream. Error: \n" + ioe);
                client.stop(); // stopping client caused by error
            }
        }
    }
}
```

```
        client.handle(streamin.readorr());  
    } catch (IOException ioe) {  
        System.out.println("Error while listening on input stream. Error: \n" + ioe);  
        client.stop(); // stopping client caused by error  
    }  
}
```