COMP2444/COMP2449 Coursework 2

Using Sockets to Create a Client-Server Application

Deadline: 10am Tuesday 2nd December

This piece of work is worth 10% of the final module grade. You are free to discuss this work but it should be considered as an individual submission and sharing of partial or completed code is prohibited.

Guidance

There are many online examples of this type of application. You are free to base your work on those examples but should ensure that your code meets the prescribed coding standards.

Your report should cite any external sources of information, including books, web resources and code examples.

A single page of coding guidelines are linked from the VLE Coursework section. Any code you submit is expected to conform to this and you will be marked against it.

Application requirements

The application developed will enable a client to access and interact with a server that maintains a folder of image files. A set of image files are provided on the VLE for you to use as test data. You are free to add to this but must respect copyright of images you find online.

1. **Server application**

a. You should design a suitable protocol that the client will use to connect to and interact with the server.
b. The server should handle multiple clients through using the Executor interface and a thread-pool architecture with a maximum of 10 simultaneous connections.
c. Functionality is required to upload and download image files and to provide a list of files stored on the server.
d. Client requests should be permanently logged in a text file in the form:

```
        date:time:ipAddress:request
```

2. **Client application**

a. To list the files stored on the server.
b. To upload/download files to/from the server.
c. A minimal GUI enabling the required functionality.

3. **Overall**

a. Good OO design.
b. Appropriate use of threading for tasks that could otherwise make your applications unresponsive.
c. Coding standards as published on the VLE are adhered to.

Submission

A short report describing the OO structure of your applications. Use diagrams as appropriate.

- a single pdf document beginning with appropriate title and your name

For the overall system:
- the protocol you have specified for communication between client and server
- a diagram of how communication is implemented between client and server (class and method)
- highlight where you have met the requirements of the system
- references

For client and server separately:
- a brief rationale for your OO design
- a brief rationale for use of threading with a diagram
- the name and purpose of each of your classes
- the name and purpose of each method
- your code as an appendix (appropriately formatted)

Demonstration

A demonstration in the lab that your applications can:
- connect client to server when running on different machines;
- meet the functional requirements 1.d, 2.a, 2.b, 2.c using example image files.

Mark scheme (40 total)

Demonstration (10)
- running across the network 2
- client list files and upload/download to/from server 2/2/2
- server logging of client connections and requests 2

Report (22)
- protocol 3
- communication description/diagram 3
- client/server OO design 2/2
- client/server class descriptions 2/2
- client/server method descriptions 2/2
- client/server threading 2/2

Code (8)
- coding standards 2
- OO encapsulation 2
- networking architecture 2
- gui pattern 2

Learning objectives

Java OO programming
Java network application design
Working with Sockets and I/O