Design Document for Lab03

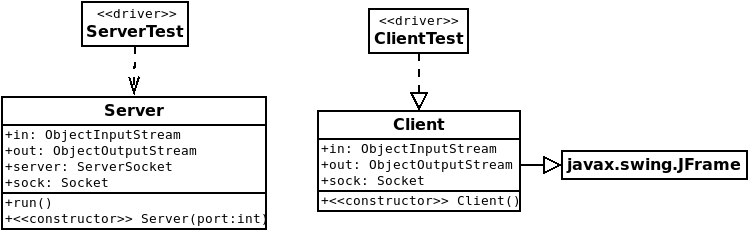
Christopher Medlin

### Networking

The protocol for communicating between client and server will use Object(Output/Input)Streams and look something like this:

|  |  |
| --- | --- |
| Client | Server |
|  | Listening... |
| Request connection | Accept connection |
| Send 2 Matrix objects through ObjectOutputStream | Read 2 Matrix objects through ObjectInputStream |
|  | Do matrix addition, send a boolean indicating success |
| Read boolean, if true, begin waiting for a Matrix object | Send Matrix object containing result |
| Read Matrix object, close connection |  |

### **Class Structure**

The new classes will follow a very similar structure to the example in the web chapter on networking

### **UI Design**



**Note on shared memory:** For the last lab, it was stated that I was duplicating matrices rather than storing them in shared memory. The way my program works is that the references to the matrices are passed to the constructor and stored in the Runnable implementation and manipulated from there. Since the same references to the 2 matrices and the result matrix are stored in that class, I don’t think any duplication is occurring, and each thread should be operating on the same matrices in shared memory, not duplicated ones. I would also have to go through a laborious process of copying the data from the duplicated matrices into a new one, which isn’t something that I did. This is why I have chosen to keep most of my code carried over from the prior lab the same.