**Student Name: Snehal Shendware**

**Total Score: 85**

Students were asked to take the sample SockClient[State] and SockServer[State]

code I gave them in class and complete the 5 functional requirements below.

The submission should have 13 files, 6 clients, 6 servers, 1 readme.

Clients and Servers are numbered as they were to build solution 2 from

solution 1, solution 3 from solution 2, etc.

1. (10) Make the server stateful by accepting only a single integer and calculating a running total.

**=: 10**

2. (10) Change the protocol by allowing the client to send a "reset" to set the running total back to 0

**=: 0 (Reset did not work)**

3. (20) Make the server stateful on a per client basis – meaning a separate running total per distinct client

**=: 20**

4. (10) Solve the integer encoding problem

**=: 10**

5. (15) Persist the state of your server-side calculator to an XML file and be able to restore it when

starting from an XML file. This will add a command-line parameter to your SockServer

**=: 15**

6. (35) Improve server throughput by making it multi-threaded yet threadsafe.

Introduce an artificial delay parameter on the command line so I can test this behavior.

**=: 30**

**(-5 Reason : Other clients are not addressed when one client's requests are being handled in synchronized mode. )**