SDJ 3, Exercises 1

# Exercise 1.1

Implement a server and client of your own choice in Java RMI and test that the client can call the server. If you prefer, you can follow the tutorial at <https://docs.oracle.com/javase/tutorial/rmi/server.html>.

# Exercise 1.2

Consider an instant messaging system. Every user should be able to see a list of online users, and initiate a chat with that user. During the chat the user should be able to write messages to their counterpart, as well as see the status of the counterpart (user is typing, user is inactive, user has disconnected). The chat ends when either user disconnects.

1. Propose at least two architectures for implementing the system above.   
   Consider: server/client vs. peer-to-peer; tiers; communicating entities (objects, components, or services); communication paradigms(request/reply, RPC, RMI, publish/subscribe, message queues, tuple spaces, distributed storage management)
2. How do your architectures deal with application state?
3. How do rate your architectures on the challenges in [Coulouris] 1.5?

# Exercise 1.3

Implement your design from exercise 1.2 in RMI.