**Exercises problem 3.**

Q1. Explain what is meant by the stream abstraction. What is the relationship between

streams and the observer pattern? What are streams useful for modeling and when

might you use them in Rich Web development?

A stream is an abstraction of a sequence data, which will be distributed in time, usually passed by a click event or data from a loop.

A observable is the subject that is being subscribed to, Meaning that you can subscribe to the stream of data to get an update on changes through an onclick event or loop etc.

Streams are used to read and continuously write data. You should use streams when reading in a big file, so that you don’t use a lot of server memory sending all the users the big file, so we use streams to read in each file and don’t consume a lot of memory.

Q2. Assume that you are building an interface to an API in your Rich Web App. Describe in

detail how you could use the RxJS library to handle asynchronous network responses to

API requests. In your opinion, what are the benefits to using a streams library for

networking over, say, promises? And what do you think are the downsides?

The RxJS library can be used to handle asynchronous network responses to API requests, it can use functions like flatMap(), mergeMap() etc

Q3. Consider three asynchronous tasks, A,B & C. What are the consequences of these

functions sharing global state? What is a good practice to alleviate any problems

associated with this?