1

A stream is an abstraction that represents a device on which input and output operations are performed. A stream can basically be represented as a source or destination of characters of indefinite length. stream is an abstraction for some data which

may or may not be present now or may or may not arrive in the

future. Once a stream is created over some future data, the stream’s data can operate on and transformed into new streams. Streams implement what is known as the observer pattern. Useful for modelling asynchronous data sources.

2

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?

RxJS is great at handling asynchronous network responses as it is an asynchronous programming language that deals with data streams and the propganation of change. Observables hand these network responses because they are like callbacks and promises

3

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?

If they are sharing a global state, they will all run together and if they use the same variables, it will cause the program to be unpredictable. You wouldn’t know when a variable change and in what order it does. This will cause many problems if this variable is writing data later. If you make them all be called by various things at different times or have the variables not be global it will help with this issue.