SSC Coursework Breakdown

**Client application** - An application running in a user's machine.

**Web Service** - A web service is an Application Program Interface (API) that runs on the server (Glassfish), which provide data to the client over http through a standardized messaging system.

**Markup Language** - A markup language is a computer language that uses tags to define elements within a document

**XML (eXtensible Markup Language)** - Designed to store and transport data.

**JAXB (Java Architecture for XML Binding)** - A software framework that allows Java developers to map Java classes to XML representations. JAXB allows storing and retrieving data in memory in any XML format.

**WSDL (Web Services Description Language)** - An XML-based file that basically tells the client application what the web service does. The WSDL file is used to describe in a nutshell what the web service does and gives the client all the information required to connect to the web service and use all the functionality provided by the web service.

A - Core Web Service (weighting - 35%)­

Implement Shares Brokering service

**Theory**: The angular thing helps format the web service operations on the web browser.

**Theory**: The data is stored in a mysterious XML file (potentially what’s generated after the binding).

**Theory**: The web service does the calculations and the client calls upon the ‘services’ that perform the calculations that are then presents to via the client app.

Theory: Unmarshelling data is what allows the web service to access the data in the XML file.

*Where is data stored (Web Service or Client)? (Maybe Lab 7 binding info?)*

Maybe need to convert main class to POJO structure (Copy AAD Work)

**Part A (Done)**

Current shares on trade should be held in an XML file based on an XSD schema containing:

* company name (**DONE**)
* company symbol on the stock exchange (**DONE**)
* number of available shares (**DONE**)
* a complex ‘share\_price’ element containing currency and value (**DONE**)
* date of the last-update of the share price. (**DONE**)

You can opt to use conventional structures to store the data (arrays, Lists, etc.) instead of XML objects. However, this will affect the standard of your work and also progression into Section-B.

**Part B**

*Check Lab 7 part 2*

The web service should allow users to:

* list shares details
* update the number of shares on offer when a purchase is made.

You might opt to utilise JAXB to generate helper Java classes, which allow reading and populating the XML documents.

You may choose to implement the core Web Service using as a RESTful service, but for full marks you need to evidence that you can build & communicate a Shares data structure similar to what can be provided by an XML Schema as in 1.a) above.

*Reading and writing XML (*[*Link*](https://stackoverflow.com/questions/7373567/how-to-read-and-write-xml-files)*)*

*Marshalling data (*[*Link*](https://stackoverflow.com/questions/21751624/how-to-write-data-into-xml-file-using-jaxb)*)*

*More Marshalling (*[*Link*](https://www.intertech.com/Blog/jaxb-tutorial-how-to-marshal-and-unmarshal-xml/)*)*

*Even more Marshalling (*[*Link*](https://stackoverflow.com/questions/13788617/jaxb-marshalling-java-to-output-xml-file)*)*

*JAXB and Root Elements (*[*Link*](http://blog.bdoughan.com/2012/07/jaxb-and-root-elements.html)*)*

*Writing data to file (*[*Link*](https://stackoverflow.com/questions/29473055/how-to-write-to-the-next-blank-line-in-a-document)*)*

**Need to get marshalling to write to next blank line and stop overwriting data.**

**Part C**

Implement a search functionality allowing customers to list shares’ offerings using various criteria such as company details, highest price, etc. More sophisticated search functionality will merit higher marks.

* The choice of client is your decision. Java GUI Apps or JSP are more appreciated but should not be attempted at the expense of providing core functionality