University of Westminster
Faculty of Science and Technology

Module code and title: 5COSC004W-Client Service Architecture Tutorial Manual	
Coursework Title	**Coding Exercise Two (CEX2) Ref Def**
Coursework Type	**Building a Real Client and Real Server from a Dummy Project**
Referral/Deferral	**20/07/19**

Contents

Rules and Regulations

1. You have 120 minutes to complete this test
2. YOU MUST BE PHYSICALLY present to the test, your signature will be checked.
3. YOU CANNOT UNDER ANY CIRCUMSTANCE sign for another person.
4. This is a closed book test, however you are allowed to consult the Java API documentation under https://docs.oracle.com/javase/8/docs/api/
5. Questions are randomized, so do not bother trying to copy or help any of your colleagues.
6. Do NOT navigate to any other site once you have started the test.
7. You are NOT allowed to have any notes, lecture slides, tutorial exercises, practice test or any other materials with you during the test.
8. You are not allowed to use any communication device during the test
9. You are not allowed to communicate with anybody other than the tutor during the test.
10. Once the test has commenced, if you leave the room, this action will terminate your attempt.
11. You will be supplied with blank paper for rough work which you must hand back to the tutor at the end of the test
12. Once you complete the test and submit, leave the room quietly as others are still working.
13. There is a total of 100 marks in 6 questions.
14. **Failing in respecting naming conventions will result in penalties.**

TASKS to prepare the code for the questions (Task 1 to 12)

1) Start Netbeans in your system.
2) Download the Dummy Project from Blackboard and import it into your Netbeans.
3) As you can observe in Figure 1 , the project you have imported contains a DummyClient which contains the logic of the test and a DummyServer which specifies the interface (not the logic, which you must develop) of the Server you have to create and deploy.

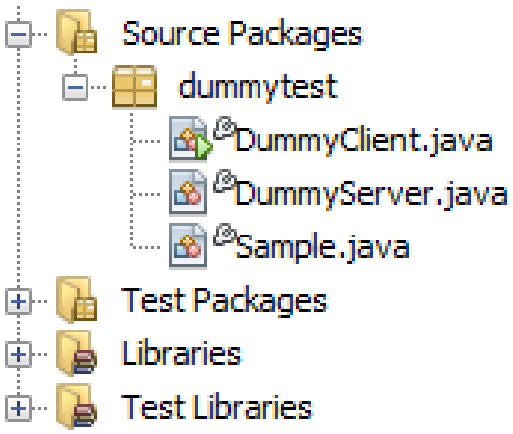

Figure 1, Dummy Project Structure

4) You can observe the test logic by analysing the method executeTest of the class DummyClient.
5) You can observe the Web Methods of the Web Service you must create by analysing the file DummyServer which defines the interface (Figure 2) of the Real Web Service you must create.

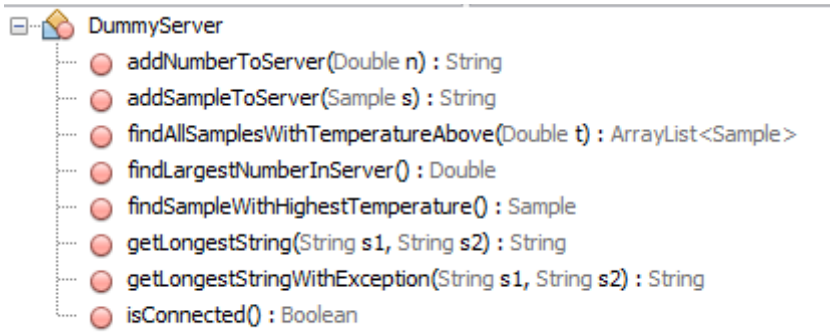

Figure 2, Dummy Server Methods

Modifier and Type	Method and Description
java.lang.String	addNumberToServer(java.lang.Double n) Adds the Double number passed as a parameter to the ArrayList of numbers on the Server. The method can return any String value.
java.lang.String	addSampleToServer(Sample s) Adds the Sample passed as a parameter to the ArrayList of samples on the Server. The method can return any String value.
java.util.ArrayList<Sample>	findAllSamplesWithTemperatureAbove(java.lang.Double t) Returns all the samples with the temperature higher than parameter t passed
java.lang.Double	findLargestNumberInServer() Returns the largest number present in the Server
Sample	findSampleWithHighestTemperature() Returns the sample with the highest temperature
java.lang.String	getLongestString(java.lang.String s1, java.lang.String s2) Returns the longest of the two String passed as a parameter
java.lang.String	getLongestStringWithException(java.lang.String s1, java.lang.String s2) Returns the length of the String passed as a parameter
java.lang.Boolean	isConnected() Returns the Boolean value true if the server is correctly running.

Table 1, Web Methods signature details

6) You have to create a Web Application named CEX2WebServer.
7) Inside the Web Application develop in point (6), you have to create Java package called server and in that package a Web Service named CEX2WebService.
8) Copy and refactor the class Sample from package dummyTest of the CEX2DummyProject into the package server of the CEX2WebServer.
9) Add the empty Web Method signatures to the Web Service following the specification of the interface of the Dummy Server.
10) Create a new Java Project called CEX2Client where you will develop the Client.
11) Add the WebService Client Stubs to the project.
12) Now you are ready to answer the questions, if you get ClassNotFind errors when executing the client, copy and paste the entire server package from the GeneratesSource folder to the SourcePackages Folder.

Coding Exercise Questions (Questions 1 to 6)

1) **Question 1 – Test Connection – 20 points**
   Write the logic on the Server of the testConnection Web Method. It should return the Boolean value true, add the usual logging lines to the Web Method.
   Deploy the Server
   Test that the Web Method works correctly with the "Test Web Service" tool.
   Drag an drop the client stubs into the client code

   Write the logic on the client test to test that the connection is running. (You can adapt the code already present on the DummyProject)

2) **Question 2 – Simple Method - 20 points**
   Write the logic on the Server of the getLongestString Web Method. It should return the longest of the two string passed as a parameters, add the usual logging lines to the Web Method.
   Deploy the Server
   Test that the Web Method works correctly with the "Test Web Service" tool.
   Drag an drop the client stubs into the client code

   Write the logic on the client test to test that the method is giving the correct result. Check that the longest between "ShortString" and "LongerString" is "ShortString".
   (You can adapt the code already present on the DummyProject)

3) **Question 3 – Multiple Methods - 10 points**
   Add an ArrayList of Doubles to the server and write the logic on the Server of the addNumberToServer Web Method, add the usual logging lines to the Web Method.
   Deploy the Server
   Test that the Web Method works correctly with the "Test Web Service" tool.
   Drag an drop the client stubs into the client code

   Write the logic on the client test to test that the method is giving the correct result. Add the following numbers to the Server: 0.0, 1.0, 2.1, 3.5, -10.0.

   (You can adapt the code already present on the DummyProject)
   Write the logic on the Server of the findLArgestNumberInServer Web Method, add the usual logging lines to the Web Method.
   Deploy the Server
   Test that the Web Method works correctly with the "Test Web Service" tool.
   Drag an drop the client stubs into the client code

   Write the logic on the client test to test that the method is giving the correct result (3.5).

4) **Question 4 – Exceptions - 10 points**
   Write the logic on the Server of the getLongestStringWithException Web Method. It should return the longest of the two string passed as parameters and it should throw an exception if one, or both strings passed as a parameter are null. Add the usual logging lines to the Web Method.
   Deploy the Server
   Test that the Web Method works correctly with the "Test Web Service" tool.
   Drag an drop the client stubs into the client code
   Write the logic on the client test to test that the method is giving the correct result. Check that exceptions are thrown only in the right occasion
   (You can adapt the code already present on the DummyProject)

5) **Question 5 – User-Defined Types - 20 points**
   Add an ArrayList of Samples to the server and write the logic on the Server of the addSampleToServer Web Method, add the usual logging lines to the Web Method.
   Deploy the Server. Drag an drop the client stubs into the client code.
   Write the logic on the client test. Define four instances of Samples, a user defined type that contains three fields: city (String), temperature (Double), humidity(Double). Add them to the server.

   - city = "London"
   - temperature = 0.2
   - humidity = 85.2

   - city = "Rome"
   - temperature = 12.2
   - humidity = 49.2

   - city = "Dublin"
   - temperature =0.1
   - humidity = 99.9

   - city = "Dubai"
   - temperature = 30.1
   - humidity = 15.0

Write the logic on the Server of the findSampleWithHighestTemperature Web Method that returns the Sample with the highest temperature. Add the usual logging lines to the Web Method. Deploy the Server. Drag an drop the client stubs into the client code

Write the logic on the client test to test that the method is giving the correct result (The sample with city = "Dubai").

6) Question 6 – Mistery Question - 20 points

Write the logic on the Server of the findAllSamplesWithTemperatureAbove Web Method that returns the ArrayList of all the Sample with the temperature above the threshold passed as a parameter. Add the usual logging lines to the Web Method.

Deploy the Server. Drag an drop the client stubs into the client code

Write the logic on the client test to test that the method is giving the correct result (With threshold 1.0, the returned array list should contain two elements).