

Module code and title: 5COSC010C.2-Client Service Architecture

Summative Assessment	In Class Test 2 (60% of final grade)
Week 12	07/04/2021

Contents

Assessment Rules and Regulations	2
Important information:.....	2
Prepare for the Assessment.....	2
Question 1 – Create the Server (15 points)	3
Question 2 – Create the Client (15 points)	3
Question 3 – Simple Method (15 points)	4
Question 4 – Exceptions (15 points)	4
Question 5 – Related Methods (15 points)	4
Question 6 – Exceptions in Related Methods (5 points)	5
Question 7 – Mystery Question (20 points)	5

Assessment Rules and Regulations

1. Summative Assessment 2 (ICT2) is worth 60% of your final mark.
2. You need to obtain 30 marks to pass it
3. Once started, you will have 120 minutes to complete the test.
4. The test will consist in a coding exercise to create a Web Service and a connected client.
5. The test will be done on Blackboard.
6. The test requires to have a working development environment (NetBeans 8.2 EE or NetBeans 8.0.2 EE)
7. All grades are provisional until moderation.

Important information:

*This online test is held under the same restrictions and codes of practice as all examinations undertaken in the University of Westminster. **You must not collude with others or plagiarise work from any source other than the module's lecture material and by submission of this test you are acknowledging that the answers you submit are your own work.***

We reserve the right to interview any student in order to discuss any aspect of their test submission. As in all examinations, any suspected cases of academic misconduct will be reported.

Prepare for the Assessment.

- a. Start Netbeans in your system.
- b. Do not have any other project on your Netbeans. If you have them please remove them.
- c. Once started, you have 120 minutes to complete and submit your assessment.
- d. You can export the zip file(s) after doing each question as a backup.
- e. ONLY upload your final zip file(s) answer to blackboard.

Question 1 – Create the Server (15 points)

- a. Create a new Web Application project in Netbeans. We will call this **ICT2WebApplication**.
- b. Create a new Java package in the Web Application where we will put our server code. We will call this **server**.
- c. Now, we can create a proper Web Service in the Project of Step (Q1.a). We will call this **ICT2WebService**.
- d. If you want, you can remove the WebMethod hello. We will not use it in the Assessment.
- e. Add a WebMethod called **testConnection** that takes no parameters and returns the **Double value 1.0**.
- f. Test the Web Service with the Test Web Service tool of NetBeans
- g. Export the Server Project as a Zip File.

Question 2 – Create the Client (15 points)

- a. Create a new Java Application project in Netbeans. We will call this **ICT2Client**.
- b. Create the client stub which connects to the Web Service you have created for Question 1.
- c. Write the Client code to create an instance of the ICT2Client class, define and invoke the **execute()** method.
- d. Drag and drop the client stub for the isConnected Web Method into the client code.
- e. Inside the execute() method, invoke the **testConnection** client stub to check if the Server is connected and returns the **correct value: 1.0**.
- f. Export the Client Project as a Zip File.

Question 3 – Simple Method (15 points)

- a. On the Server, implement a Web Method called **longestString(String s1, String s2) which returns the longest between s1 and s2.**
- b. Test the Web Service with the Test Web Service tool of NetBeans
- c. On the Client, refresh the Web Service reference.
- d. Drag and Drop the client stub for **longestString** into the client code.
- e. Only if the test of Q2 was successful, test the **longestString** Web Method on client.
- f. Export the Client and Server Projects as a Zip (you can overwrite the old once).

Question 4 – Exceptions (15 points)

- a. On the Server, implement a Web Method called **safeLongestString(String s1, String s2) which returns the longest between s1 and s2** and throws an **Exception** when one or both the passed parameters are null.
- b. Test the Web Service with the Test Web Service tool of NetBeans
- c. On the Client, refresh the Web Service reference.
- d. Drag and Drop the client stub for the **safeLongestString** Web Method into the client code.
- e. Test the **safeLongestString** Web Method on the client.
- f. Export the Client and Server Projects as a Zip Files (you can overwrite old once).

Question 5 – Related Methods (15 points)

- a. On the Server, implement a Web Method called **addDoubleOnServer(Double)** which adds the passed Double to the Server. You are free to use any type to store the values but ArrayList is a good choice.
- b. Test the Web Service with the Test Web Service tool of NetBeans.
- c. On the Server, implement a Web Method called **getLargestDouble()** which returns the largest of the Double values previously added to the Server.
- d. Test the Web Service with the Test Web Service tool of NetBeans.
- e. On the Client, refresh the Web Service reference.
- f. Drag and Drop the client stub for the **addDoubleOnServer** and **getLargestDouble** Web Methods and Test both methods.
- g. Export the Client and Server Projects as a Zip Files (you can overwrite the old once).

Question 6 – Exceptions in Related Methods (5 points)

- a. Add exceptions to the Web Methods of Question 5 to safeguard against crash in the system (e.g. finding the largest value without samples added and adding a null value on the server).
- b. Test both exceptional behaviour from the Client.
- c. Export the Client and Server Projects as a Zip Files (you can overwrite the old once).

Question 7 – Mystery Question (20 points)

- a. On the Server, implement a Web Method called **getAllDoubleSamplesAbove(Double)** which returns all Double values (added to the server in Q5) greater or equal than the Double Value passed as a parameter. Although it is not mandatory, it is suggested to return an `ArrayList<Double>` type.
- b. Test the Web Service with the Test Web Service tool of NetBeans.
- c. On the Client, refresh the Web Service reference.
- d. Drag and Drop the client stub for the Web Methods into the client code.
- e. Test the **getAllDoubleSamplesAbove** Web Method with different values.
- f. Export the Client and Server Projects as a Zip Files (you can overwrite the old once).