Candidate Number: 1705299

Explanation for 'RMI vs Java Web Services' coursework

*Explanation for Step 2

- To return a result of calculator operation, made a new invocated method 'RemotelyInvocatedMethodNew' which returns a 'String type' result.
- Added a new method called 'CalculatorNew' for calculator operator. It is called by the RemotelyInvocatedMethodNew'.

*Explanation for Step 3

- Changed the main method of the RMI client program to call a new invocated method of the server program.
- With a scanner object, until the user terminates the while loop, it keeps call the RMI server method.

The disadvantages of the RMI client implementation here, the client and the server are tightly coupled. So, a client program directly accesses functions of the server program. If functions of the server are changed, the client side program using the functions must be immediately changed also. If not, it results compile errors on the server side. **As a result, before updates of the functions commit to the server program, it should be notified to the client program, to prevent the compile errors on the client program.**

*Explanation for Step 4

On the other hand, the one of the advantages of Web Service implementation is the client program has a loose coupling with the server program. As importing Web Services by a wsdl, the client independently has the resources published from the server side to the its own server, such as classes, functions. If changes occurred of server program, it doesn't result compile errors instantly on the client server. So, although the client implementation and the updated functions of the server side don't match, it doesn't cause inevitable compile errors to the client program. **Thus, the beneficial outcome of using Web Services is that the client program can have enough time to prepare before updating its program.** But to use the functionalities of the server program, the update on the server program must be happened.