CSC3004 Cloud and Distributed Computing
Tutorial / Lab 2

1. Based on the Java RMI Tutorial, implement a Calculator Service using Java RMI. The following Remote Interface called `calculator` has been defined:

```
public interface calculator extends java.rmi.Remote {

    // add takes two long values, adds them together and returns the resulting
    // long value
    public long add(long a, long b)
         throws java.rmi.RemoteException;

    // sub takes two long values: a and b. It subtracts b from a and returns
    the resulting long value
    public long sub(long a, long b)
          throws java.rmi.RemoteException;

    // mul takes two long values and multiplies them together. The resulting
    long value is returned
    public long mul(long a, long b)
          throws java.rmi.RemoteException;

    // div takes two long values a and b. a is divided by b and the resulting
    long value is returned
    public long div(long a, long b)
          throws java.rmi.RemoteException;

    // pow takes a long value a and an int value b. Where a^b is performed
    public long pow(long a, int b)
          throws java.rmi.RemoteException;

}
```

2. Implement the servant class called `calculatorimpl` to provide the implementation for all the remote methods defined in the `calculator` remote interface. The following shows an example implementation of the method `add`:

```
// Implementation of the add method
// The two long parameters are added and the result is returned
public long add(long a, long b)
      throws java.rmi.RemoteException {

      System.out.println("performing addition: " + a + " + " + b);

      return a + b;
}
```

3. Create the `calculatorserver` class to host the `calculatorimpl` servant class and bind the resulting remote object instance to the `RMIRegistry`. This will create a remote object reference for the `calculator` remote object.

4. Create a `calculatorclient` class to perform remote invocation on the server, by first lookup for the remote object from the `RMIRegistry`, and then invoke the method `add(), sub(), mul(), div()` and `pow().`

5. To run Java RMI:

    a. Start `RMIRegistry` in the same directory as your Java classes.
    b. Run the `calculatorserver`
    c. Run the `calculatorclient`
    d. If you have two devices, e.g., laptop and Raspberry Pi, try running the server on Raspberry Pi and running the client on your PC/Laptop. Note that both devices must be on the same network and you will need to change “localhost” in the `calculatorclient` to the IP address of the Raspberry Pi.