

**Assignment 2**  
**Distributed Systems, Monsoon 2017**  
**Deadline: 13th Sept. 2017 at 9 PM.**

**Q1)** Your task is to develop a Distributed Banking System, in a client-server setup using *Java's RMI protocol*. You can think of the client as an ATM machine that is trying to call certain Remote Methods provided by the central Banking server. **(50m + 10m bonus)**

The server manages all the client details using the fields given below. You can pre-populate the fields on the server side.

- Name
- Account No. : Unique 10 digit number
- Account Type : Premium/Basic
- Contact Info : The port/IP where the client can receive a notification of their transaction (for SMS Feature)
- Balance
- Transaction History: Contains 'transaction\_id', 'amount', 'date' and 'time' of the transaction.

Optimized Data Structures based on the task to be performed are expected. Eg: Hash table for data enquiry etc. Don't worry about memory requirements. The focus is on quicker response time from the server. **(5 marks)**

The Methods provided by the server include:

- deposit(acc\_no, amt): Returns a confirmation of the *increase* in balance with the transaction\_id and revised account balance. **(10 marks)**
- withdraw(acc\_no, amt): Returns a confirmation of the *decrease* in balance with the transaction\_id and revised account balance. **(10 marks)**
- balance(acc\_no): Returns the account balance **(5 marks)**
- transaction\_details(acc\_no, start\_date, end\_date):
  - Return the Transaction History for transactions based only on the date range. **(8 marks)**
  - Return the Transaction History for all of the transactions. **(7 marks)**
  - Display the result in a neat tabular format. **(5 marks)**

**Bonus (10 marks) - Theory Question:**

SMS Feature: When the user deposits or withdraws money, a message containing the 'Transaction History' of that transaction should be sent to the the mobile phone of the user. The mobile phone of the user is another client that is connected via the port/ip given in the 'Contact

Info' field. How will you implement this? (i.e., features/APIs from Java's RMI protocol required)  
Explain in 5-10 lines.

## Q2)

Implement a client-server model in Java based on RMI. **(50 marks)**

Server should have these functionalities:

- Primality Test
- Palindrome Test
- N<sup>th</sup> Fibonacci Number
- String case converter (upper to lower and lower to upper)

The client and the server must first set up a secret key for communication using Diffie Hellman Key Exchange protocol. Any further communication would be through encrypted messages.

**Encryption Scheme**(Message m):

```
//Key K  
Return  $m \oplus K$ 
```

**Decryption Scheme**(Cipher c):

```
//Key K  
Return  $c \oplus K$ 
```

If the key length is less than the message length, the key is repeatedly appended to itself till the message size is met and then xor-ed.

Ex:    Message - 01101010, Key - 011  
      Cipher -  $(01101010) \oplus (01101101)$

Server should decrypt the received encrypted binary message, convert into appropriate data type, perform requested operation, then encrypt the answer and send back to the client. Upon receiving the encrypted answer, client should decrypt and print it to the console.

### Marks Distribution:

Implementation	-	10
Functionalities	-	5 each
Encryption & Decryption	-	20

### Upload Format:

- 3 source files : named as Client\_<Roll\_number>.java, Server\_<Roll\_number>.java, Interface\_<Roll\_number>.java
- Directory Structure:
  - <Roll\_number Folder>
    - Q1

- all the .java source files
  - Q2
    - all the .java source files
  - Readme.pdf
  - Bonus.pdf
- The README should contain implementation details (design of your system), compilation procedure, features implemented.

**Note:**

- Submit the answer for the Bonus question in a PDF
- There will be VIVA asked for each question.
- Please follow the above upload format strictly. Assignments that do not follow it will not be evaluated.
- Any form of plagiarism shall result in *negative marks*.