This homework can be done in groups of 2 students. Submit one solution for the entire group. Submission requirements will be given at a later time.

DUE: DEC 8TH

Create an Authentication Service Using RMI

Server Description:

1. Create RMI engine that accepts connections and remote method calls.
2. You may use hard code a port number that the RMI server and client will be communicating on.

Client Description:

1. Client program will prompt user to enter user name and password.
2. Client program will then use RMI (item 1) to identify if this user name/password combination is correct.
    a. If user name/password combination is NOT correct, display invalid credentials on the client screen and exit.
    b. If user name/password combination is correct, display welcome User or welcome Administrator based on user account type.
3. If the regular user logged in give option:
    (1) Display user's schedule (RMI item 5)
    (2) Add an event to user's schedule (RMI item 6)
    (3) Edit an event from user's schedule (RMI item 7)
    (4) Delete an event from user's schedule (RMI item 8)
    (5) Change password to user's account (RMI item 3)
    (user may only change his/her password)
    (6) Exit Program
4. If the administrator logged in give option
    (1) Create/Delete User Account (RMI item 2)
    (2) Reset Password for User Account (RMI item 3)
    (admin may change password for all accounts)
    (3) Display List of User Account (RMI item 4)
    (4) Display schedule associated with specific user (RMI item 5)
    (5) Add an event to selected user's schedule (RMI item 6)
    (6) Edit an event from selected user's schedule (RMI item 7)
    (7) Delete an event from selected user's schedule (RMI item 8)
    (8) Exit Program

Each event is composed of a date and the event title, the event will be displayed according to *ascending* order where earlier events are displayed first, and later events are displayed last.

Methods to be Executed Remotely:

1. Handle authentication requests to verify if user name and password is correct.
2. Handle account creation request to create and delete new accounts. (No need to worry about conflict account creation) *All accounts created will be regular accounts
3. Handle request to change password.
4. Handle request to list all user accounts.
5. Handle request to display schedule.
6. Handle request to add an event to the schedule.
7. Handle request to edit an event from the schedule.
8. Handle request to delete an event from the schedule.

Implementation Tips:

Base your implementation on the RMI Pi compute example.

Default admin account should be **admin**/**admin** for user name and password.
Server can keep track of all accounts and associated schedule in text file or in memory, it will be the student's choice.
Menu options should be repeatedly presented to the user until exit program is selected.

Sample Testing Procedure:
1. Login with admin/admin => success login, present administrator menu
2. Create account smith/1234 => account created successfully
3. Exit admin account
4. Login with smith/123 => fail login
5. Login with smith/1234 => success login, present regular user menu
6. Change password to 12345
7. Create an event and store on the server.
8. Exit smith account
9. Login with smith/1234 => fail login
10. Login with smith/12345 => success
11. Display smith's schedule
12. Exit smith account