**Name: Kush Raina**

**ID: 1001567809**

**Summary:**

This program is implementation of Student Advising Process as we see it in our advising process, using RMI concept in Java. This program is part of the course CSE 5306 Fall 2017. I have developed this program in Java 8.

Requirements: Java installed on the system and Eclipse.

**Instructions:**

1. Import java project attached in the zip file in Eclipse.
2. Once you import the project in eclipse you will be able to see six packages in the project folder.

* AdvisingProcess
* MessageQueueServer
* notificationProcessGUI
* RMIImplementataion
* RMIInterface
* StudentProcess

1. Expand the **MessageQueueServer** package and you will be able to see **studentMSQ.java** which is the message Queue server we have to start before starting our project. This java class will start the RMI Registry. Run the java file as java application. A pop up window with the title “Message Queue” will open.
2. After the MessageQueueServer has started expand the **StudentProcess** package run **studentGUI.java** file as java application. When you run it a GUI window will open asking you to enter the name of student and subject which the student is taking. After entering the student name and subject click submit to enter . User can see the entries going into the message queue from student to advisor. User can enter how many entry he wants and if he wishes to exit the student process he can do it by pressing exit button.
3. After entering the students detail, open **AdvisingProcess** package where we will find **advisingProcessGUI.java**. Run this java file and we will be prompted with advisingProcess GUI where the advisor will get values from message queue server and will decide based on random probability that whether student will be admitted or not. You will be able to see in Advising window that advisor has taken message from message queue server. After processing the message the advisor will send its decision to message queue for notification process. User can anytime exit from notification process by clicking on EXIT button on advising process GUI. If there are no messages from message queue to advising process it will wait for 3 seconds and message queue window will pop-up where you can verify if there are any messages to Advising process also in advising process it will show **nomessages** if no messages are present for advisor.

User can anytime see what messages are going into the message queue.

1. After advising process open the **Notification Process** package where we can see **NotificationProcessGUI.**java . Run the java file and window will be opened where user can get values from message queue server to notification process about the advisor decision. The notification process will continuously be checking for the messages from the message queue for Notification process.

Simultaneously, we will be getting the messages updated on the messagequeue when notification process the message for the student process. When there are no messages for the notification process it will display no values on the screen and will wait for 7 seconds before it checks it again.

7. After the notification process is completed then again run the student process ,on the bottom you will able to see the messages from notification to the student

***Note***: In this project I have used a text file named messagequeue.txt file to store all the messages .

Also, to see the messages from notification in the student process you have to run the student process fresh everytime.

I have referenced my code from below given URL’s:

1. <https://www.youtube.com/watch?v=3fq4AdaiGFA>
2. <https://www.youtube.com/watch?v=qpOWzdTpCW4>
3. <https://www.mkyong.com/java/how-to-read-file-from-java-bufferedreader-example/>
4. <https://stackoverflow.com/questions/20307754/save-data-from-linkedlist-in-java>