**Description:**

In this project, we have simulated an online advising system which is similar to what students use for registering classes. Students send a request to the advisor for waving a course. After some time, the advisor approves or disapproves the request subsequently the student is notified of the advisor’s decision. We will require 4 processes to simulate this: a student process, an advisor process, a notification process and a message queuing server (MQS). The student process, advisor process and notification process communicate through the message queuing server (message oriented middleware). Communication from these processes to the message queuing server is through remote method invocation.

In this lab, the following features have been incorporated:

1.Message queuing server is able to store or return a message from or to any requesting process.

2. Implemented the process in a single data structure.

3. Dynamic persistence implemented and retrieved messages are deleted from the queue.

4.The code works independently of the order in which the processes are started.

**Steps to execute**

1. Run the server first followed by the student process.
2. Run the student process and enter your name and the course you want to be waived.
3. Run the advisor process and check whether the course was waived or not.
4. Run the notification process to check the response of the advisor.
5. If you want to enter more courses then you can repeat step 2 to 4

**References:**

<https://www.youtube.com/watch?v=GURClZeR96E&t=662s>

<https://www.javatpoint.com/Scanner-class>

https://www.youtube.com/watch?v=V1IZdKCjTgg

https://github.com/varunnmurthy/personal-project.git

https://docs.oracle.com/javase/tutorial/rmi/implementing.html

<http://www.vogella.com/tutorials/JavaCollections/article.html>