**Name: Manohar Rajaram**

**Student ID: 1001544414**

# Note

* Language used: **Python**
* Python version: **3.6.7**
* IDE used: **PyCharm**

# Project Description

Online Advising Simulation Using Message Queuing

# Execution Steps

Step 1

* Run the file ‘mqs.py’. It’s the Message Queue Server
* If running from command window, change directory to the folder containing the server file. Execute the following command from command window/terminal or via PyCharm:

**python mqs.py**

* If running from PyCharm, run the mqs.py file.
* A server window pops up.

Step 2

* Similarly, run the files ‘student.py’, ‘advisor.py’ and ‘notifier.py’
* If running from command window, change directory to the folder containing the above files. Execute the following command on different command window/terminal or via PyCharm:

**python student.py**

**python advisor.py**

**python notifier.py**

* If running from PyCharm, open the respective file and run the same
* Three windows pop up, one per each student, advisor and notifier process
* The Message Queue server displays the connected list of process
* We can follow the prompts on the client from here on or follow the below steps
* You may need to move the windows to adjust visibility

Step 3

* On the student window, enter name and course
* Observe the advisor and notifier windows for approval and rejection of student- courses.
* The Message Queue server window, shows the live queue content.
* The notifier and advisor display ‘No Messages Found’ on their windows if no input is given on the student window.

Step 4

* To verify the volatility(non) of the Message queue server, disconnect the notifier (or the advisor) using the ‘Quit’ button and input few student-courses from the student window.
* We can see that the queue on the server, doesn’t get cleared because the notifier window is closed.
* Terminate the message queue server by clicking ‘Disconnect’ and end other process as well.
* Restart the message queue server. We can see that the message queue server is already loaded with the previously non-notified input before the server was disconnected.
* Upon server disconnection, the queue content, if any, will be stored in a text file called ‘buffer.txt’ and it stays there till the message queue server is restarted again.
* Upon restarting the advisor and notifier process, the queue in the server gets cleared and passed to the advisor and notifier.

# Limitations

* Since we have to use sleep() to while polling, it takes few seconds before the notifier and advisor window closes once the user clicks on Quit.
* I am using “:” as the delimiter between student name and course during display, so both of them should not contain “:” in them

# References

* [Chat App- Start Up](https://medium.com/swlh/lets-write-a-chat-app-in-python-f6783a9ac170)
* [Linked List Queue](https://www.sanfoundry.com/python-program-implement-queue-data-structure-using-linked-list/)
* [File Access](https://www.geeksforgeeks.org/reading-writing-text-files-python/)