# Functional Programming (ECS713U/ECS713P)


## Shrey Mahar (ID No.- 220307635)

## School of Electronic Engineering and Computer Science

## Queen Mary University of London

s.mahar@se22.qmul.ac.uk


We used Control.Concurrent library to spawn threads, System.Ransom to generate random numbers, Control.Concurrent.MVar, MVar which is used for communication between concurrent threads, and first of all we defined user data type which include the username and message data type which contains sender, receiver and the body(text of message). Then we've created some random users with their username and the random messages between a-z. Following output we got after this:-

User 3 Sent this message: e

User 1 Sent this message: x

User 9 Sent this message: y …….

Then a function to stimulate a user thread is created, simulateUser. In this we used modifyMVar and readMVar. modifyMVar is used to modify the content of the MVar and replace the original content of MVar, later we used readMVar which reads the data to MVar and raiseCount function is called in the simulateUser to add the count and The simulateUser function is used simulate a user sending random messages, where the number of messages sent is limited to 100.

**Issues:-** While working on the project faced some issues on how to increment the count and later solved it by calling it in the simulateUser function.

**Main-Thread:-**The 'main' thread makes a list of 10 'User' with random usernames and sends 100 messages in total. The totalMessage MVar is created to keep track of the number of messages sent by each user. Another design decision made is to use randomR from the System.Random module to generate random message values. This is the final output we are getting:-

User 1 received 11 messages.

User 2 received 11 messages.

……..

User 9 received 11 messages.

User 10 received 10 messages.