Interthread communication by using Queue:

Queues Concept is the most enhanced Mechanism for interthread communication and to share data between threads.

Queue internally has Condition and that Condition has Lock.Hence whenever we are using Queue we are not required to worry about Synchronization.

If we want to use Queues first we should import queue module.

import queue

We can create Queue object as follows

```
q = queue.Queue()
```

Important Methods of Queue:

1. put(): Put an item into the queue.
2. get(): Remove and return an item from the queue.

Producer Thread uses put() method to insert data in the queue. Internally this method has logic to acquire the lock before inserting data into queue. After inserting data lock will be released automatically.

put() method also checks whether the queue is full or not and if queue is full then the Producer thread will entered in to waiting state by calling wait() method internally.

Consumer Thread uses get() method to remove and get data from the queue. Internally this method has logic to acquire the lock before removing data from the queue.Once removal completed then the lock will be released automatically.

If the queue is empty then consumer thread will entered into waiting state by calling wait() method internally.Once queue updated with data then the thread will be notified automatically.

Note:
The queue module takes care of locking for us which is a great advantage.

Eg:

```
1) from threading import *
2) import time
3) import random
4) import queue
5) def produce(q):
```


DURGASOFT, # 202, 2nd Floor, HUDA Maitrivanam, Ameerpet, Hyderabad - 500038,
☎ 040 – 64 51 27 86, 80 96 96 96 96, 92 46 21 21 43 | www.durgasoft.com