```
import threading
import time
import logging
import random
from queue import Queue

logging.basicConfig(level=logging.DEBUG,
                    format='(%(threadName)-9s) %(message)s', )

BUF_SIZE = 20
q = Queue(BUF_SIZE)


class ProducerThread(threading.Thread):
    def __init__(self, target=None, name=None):
        super(ProducerThread, self).__init__()
        self.target = target
        self.name = name

    def run(self):
        while True:
            if not q.full():
                item = random.randint(1, 10)
                q.put(item)
                logging.debug('Putting ' + str(item)
                              + ' : ' + str(q.qsize()) + ' items in queue')
                time.sleep(random.random())
        return


class ConsumerThread(threading.Thread):
    def __init__(self, target=None, name=None):
        super(ConsumerThread, self).__init__()
        self.target = target
        self.name = name
        return

    def run(self):
        while True:
            if not q.empty():
                item = q.get()
                logging.debug('Getting ' + str(item)
                              + ' : ' + str(q.qsize()) + ' items in queue')
                time.sleep(random.random())
        return


if __name__ == '__main__':
    pTh = ProducerThread(name='producer')
    cTh = ConsumerThread(name='consumer')

    pTh.start()
    time.sleep(4)
    cTh.start()
    time.sleep(4)
```