

PART II

Question 1:

In the absence of synchronization, if one thread executes `(in.peek() != null)`, it returns true and then the time slice expired and it switch to the other thread. However the other executes `in.remove()`, if there I only one element in the queue before the remove, when switching back, `(isInstance(in.peek().getClass()))` will cause a `NullPointerException`.

Question 2:

There is only one producer and the producer run faster than the middleman (regarded as the consumer of the `GeneralPurposeQueue`). Therefore, there is never a situation where the `MAX_QUEUE_SIZE` constraint is violated.

Question 3:

```
synchronized (out) {  
    if (out.size() >= 10) {  
        continue;  
    } else {  
        out.offer(outObj);  
        if (out.contains(null)) {  
            System.out.println("why did this happen?");  
        }  
        outObj = null;  
    }  
}
```

I add synchronization to the operation of the out queue to avoid other thread accessing or modifying the data of out between one thread's operation when switching the threads.