

Outputs of Multi Threading Assignment

1. ./Assignment/src/com/evenOdd/EvenOdd.java : Print even, odd using synchronized, wait, notify

```
1 package com.evenOdd;
2
3 public class EvenOdd extends Thread {
4     static int i = 1;
5     Object lock;
6
7     EvenOdd(Object lock) {
8         this.lock = lock;
9     }
10
11     @Override
12     public void run() {
13         while (i <= 100) {
14             if (i % 2 == 0 && Thread.currentThread().getName().equals("EvenNumber")) {
15                 synchronized (lock) {
16                     System.out.println(Thread.currentThread().getName() + ": "
17                                     + i);
18                     i++;
19                     try {
20                         lock.wait();
```

Problems @ Javadoc Console Progress

```
<terminated> EvenOdd [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (20-Jan-2019, 11:44:24)
EvenNumber: 84
OddNumber : 85
EvenNumber: 86
OddNumber : 87
EvenNumber: 88
OddNumber : 89
EvenNumber: 90
OddNumber : 91
EvenNumber: 92
OddNumber : 93
EvenNumber: 94
OddNumber : 95
EvenNumber: 96
OddNumber : 97
EvenNumber: 98
OddNumber : 99
EvenNumber: 100
OddNumber : 101
```

2. ./Assignment/src/com/producerConsumer/ProducerConsumer.java: Use BlockQueue to solve Producer Consumer problem

```
1 package com.producerConsumer;
2
3 import java.util.Random;
4 import java.util.concurrent.BlockingQueue;
5 import java.util.concurrent.LinkedBlockingQueue;
6
7 public class ProducerConsumer {
8     static int max = 5;
9     static BlockingQueue<Integer> q = new LinkedBlockingQueue<Integer>(max);
10
11     public static void main(String[] args) {
12
13         Producer producer = new Producer();
14         Consumer consumer = new Consumer();
15         producer.start();
16         consumer.start();
17     }
18
19     static class Producer extends Thread {
20         Random random = new Random();
```

Problems @ Javadoc Console Progress

<terminated> ProducerConsumer [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (20-Jan

Produced: 4
Consumed: 4
Produced: 1
Consumed: 1
Consumed: 2
Produced: 2
Produced: 1
Consumed: 1
Produced: 1
Consumed: 1
Produced: 1
Consumed: 1