Task 3: Synchronization and Inter-thread Communication
Implement a producer-consumer problem using wait() and notify() methods to handle the correct processing sequence between threads.
ANS:

```java
package com.Day23;
import java.util.LinkedList;
import java.util.Queue;
public class Task3 {
  public static void main(String[] args) {
     Buffer buffer = new Buffer(5); // Buffer with capacity of 5
     Thread producerThread = new Thread(new Producer(buffer));
     Thread consumerThread = new Thread(new Consumer(buffer));
     producerThread.start();
     consumerThread.start();
  }
}
class Buffer {
  private final Queue<Integer> queue;
  private final int capacity;
  public Buffer(int capacity) {
     this.queue = new LinkedList<>();
     this.capacity = capacity;
  }
  public synchronized void produce(int item) throws InterruptedException {
     while (queue.size() == capacity) {
        wait(); // Wait until space is available
     }
     queue.add(item);
     System.out.println("Produced: " + item);
     notifyAll(); // Notify consumers that new item is available
  }
  public synchronized int consume() throws InterruptedException {
     while (queue.isEmpty()) {
        wait(); // Wait until items are available
     }
     int item = queue.poll();
     System.out.println("Consumed: " + item);
     notifyAll(); // Notify producers that space is available
     return item;
  }
}
class Producer implements Runnable {
  private final Buffer buffer;
  public Producer(Buffer buffer) {
     this.buffer = buffer;
  }
  @Override
  public void run() {
     int item = 0;
     try {
        while (true) {
           buffer.produce(item++);
```

```java
                Thread.sleep(100); // Simulate time taken to produce item
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
class Consumer implements Runnable {
    private final Buffer buffer;
    public Consumer(Buffer buffer) {
        this.buffer = buffer;
    }
    @Override
    public void run() {
        try {
            while (true) {
                buffer.consume();
                Thread.sleep(150); // Simulate time taken to consume item
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```