**PRACTICAL – 7(6)**

**Aim: Write a program to solve producer-consumer problem using**

**thread synchronization.**

**SOURCE CODE:**

public class Practical\_7\_6 {

    public static *void* main(String[] *args*) {

        Producer p = new Producer();

        Consumer c = new Consumer(p);

        Thread t1 = new Thread(p);

        Thread t2 = new Thread(c);

        t2.start();

        t1.start();

    }

}

class Producer extends Thread {

    StringBuffer buffer;

    Producer() {

        buffer = new StringBuffer(4);

    }

    public *void* run() {

        synchronized (buffer) {

            try {

                for (*int* i = 0; i < 4; i++) {

                    System.out.println("Process " + i + " is generated");

                    buffer.append(i);

                }

            } catch (Exception e) {

                System.out.println(e);

            }

            System.out.println("\nBuffer is Full");

            buffer.notify();

        }

    }

}

class Consumer extends Thread {

    Producer p;

    Consumer(Producer *temp*) {

        p = *temp*;

    }

    public *void* run() {

        synchronized (p.buffer) {

            try {

                p.buffer.wait();

            } catch (Exception e) {

                System.out.println(e);

            }

            System.out.println("\nFor Consumer side:");

            for (*int* i = 0; i < 4; i++) {

                System.out.println(p.buffer.charAt(i) + " Process is Consumed");

            }

            System.out.println("\nBuffer is empty.");

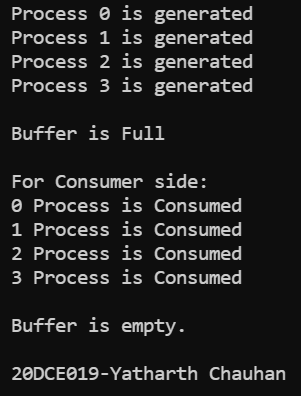
        }

        System.out.println("\n20DCE019-Yatharth Chauhan");

    }

}

**OUTPUT:**

****

**CONCLUSION:** In this practical we can solve the producer and consumer problem by using the java multithreading concept.