

Dt : 22/11/2022

Ex:(Program to demonstrate Producer-Consumer problem)

Producer.java

```
package test;
public class Producer implements Runnable{
    public StringBuffer sb=null;
    public Producer() {
        sb = new StringBuffer();//Buffer created while
        Constructor execution
    }
    @Override
    public void run() {
        try {
            //Locking the Buffer
            synchronized(sb) {
                for(int i=1;i<=10;i++)
                {
                    sb.append(i+":"); //Adding data to Buffer
                    System.out.println("Producer Appending
data....");
                    Thread.sleep(2000);
                } //end of loop
                sb.notify();
                //release the lock on buffer and Send message to next
                waiting thread
            } //end of lock
        } catch (Exception e) {e.printStackTrace();}
    }
}
```

Consumer.java

```
package test;
public class Consumer implements Runnable{
    public Producer prod=null;
    public Consumer(Producer prod) {
        this.prod=prod;
    }
    @Override
    public void run() {
        try {
            //Locking the Buffer
            synchronized(prod.sb) {
```

```

        System.out.println("Consumer blocked...and
Waiting for msg");
        prod.sb.wait();
        System.out.println("*****Display Using
Consumer*****");
        System.out.println(prod.sb.toString());
    } //end of lock
} catch (Exception e) {e.printStackTrace();}
}
}

```

**DemoThread5.java(MainClass)**

```

package maccess;
import test.*;
public class DemoThread5 {
    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(); //Consumer
        t1.start(); //Producer
    }
}

```

**o/p:**

**Consumer blocked....and Waiting for msg**

**Producer Appending data....**

**Producer Appending data....**

**Producer Appending data....**

**Producer Appending data....**

**Producer Appending data....**

**Producer Appending data....**

**Producer Appending data....**

*Producer Appending data....*

*Producer Appending data....*

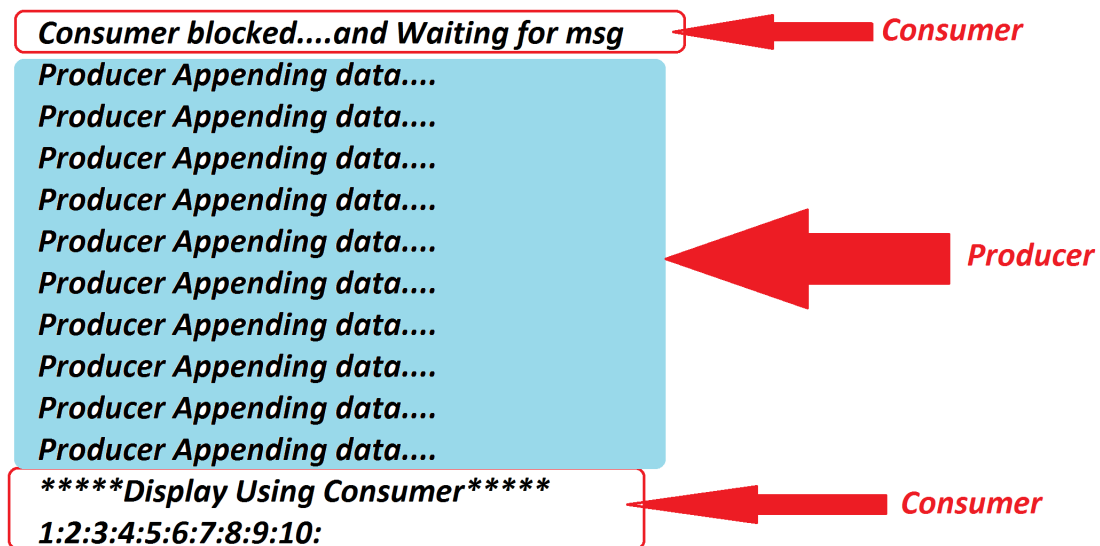
*Producer Appending data....*

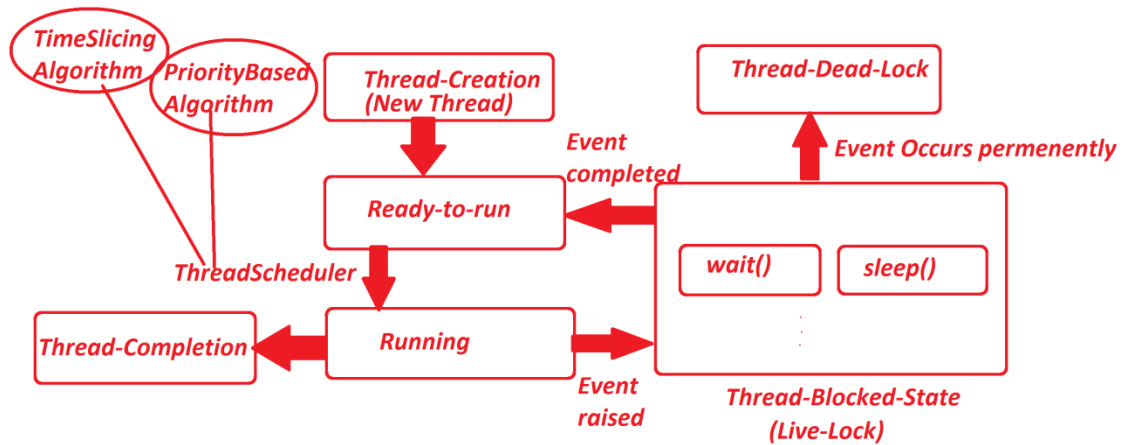
**\*\*\*\*\*Display Using Consumer\*\*\*\*\***

**1:2:3:4:5:6:7:8:9:10:**

-----

*Execution flow of above program:*





**Note:**

=>Consumer will wait using wait() method until Producer sends the msg in the form of notify().

**faq:**

**define Thread Life-Cycle?**

=>Thread Life-Cycle demonstrates different states of thread from thread-creation to thread-completion or thread-creation to thread-termination.

=>The following are the stages of thread:

**1.Thread Creation**

**2.Ready-to-run**

**3.Running**

**(a)Thread-Completion**

**(b)Thread-Blocked-state**

=>Thread-Dead-Lock

Diagram:

