Practical No.: 01

Aim: Process Communication:

- **a.** Write a program to give a solution to the producer–consumer problem using shared memory.
- **b.** Write a program to give a solution to the producer–consumer problem using message passing.

Source Code:

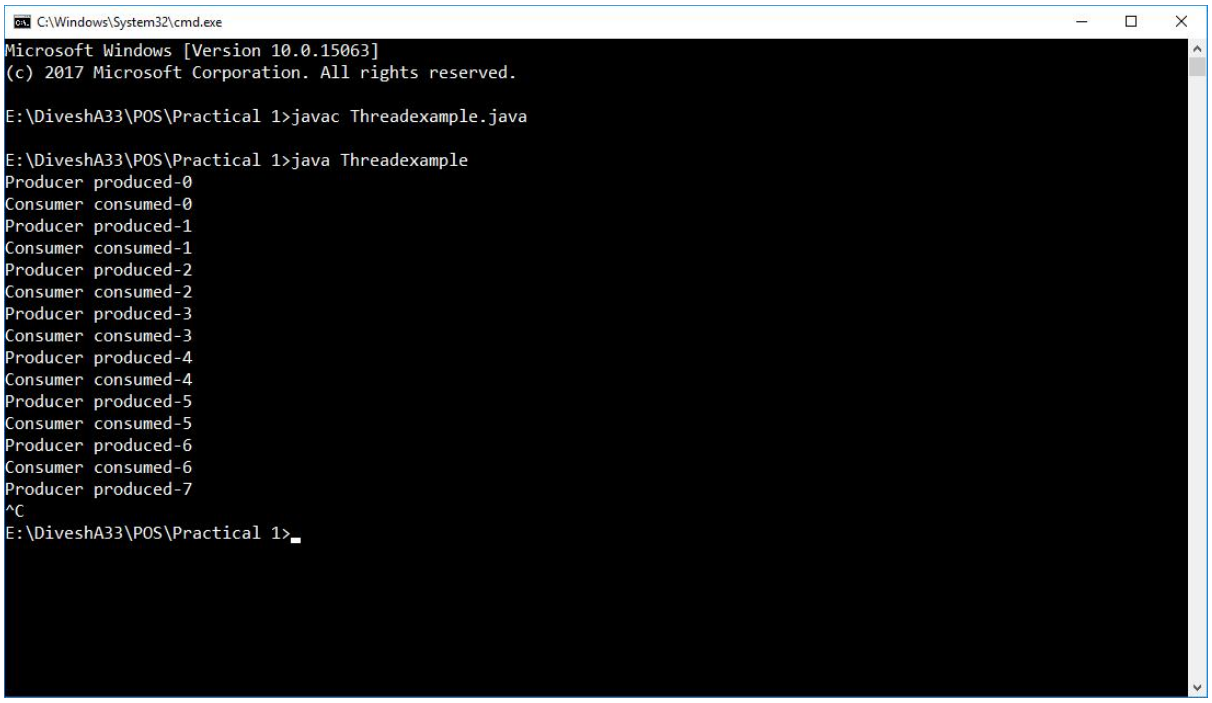
a. Write a program to give a solution to the producer–consumer problem using shared memory.

```
import java.util.LinkedList;
public class Threadexample
{
        public static void main(String args[]) throws InterruptedException
        {
                final PC pc = new PC();

                Thread t1 = new Thread(new Runnable()
                {
                        @Override
                        public void run()
                        {
                                try
                                {
                                        pc.produce();
                                }
                                catch (InterruptedException e)
                                {
                                        e.printStackTrace();
                                }
                        }
                });

                Thread t2 = new Thread(new Runnable()
                {
                        @Override
                        public void run()
                        {
                                try
                                {
                                        pc.consume();
                                }
                                catch (InterruptedException e)
                                {
                                        e.printStackTrace();
                                }
```

```
                    }
            });

            t1.start();
            t2.start();

            t1.join();
            t2.join();

    }

    public static class PC {

            LinkedList<Integer> list = new LinkedList<>();
            int capacity = 2;


            public void produce() throws InterruptedException
            {
                    int value = 0;
                    while (true) {
                            synchronized (this)
                    {
                            while (list.size() == capacity)
                                    wait();

                            System.out.println("Producer produced-"+ value);

                            list.add(value++);

                            notify();

                            Thread.sleep(1000);

                    }
            }
    }

    public void consume() throws InterruptedException
    {
            while (true) {
                    synchronized (this)
                    {
                            while (list.size() == 0)
                                    wait();

                            int val = list.removeFirst();

                            System.out.println("Consumer consumed-"+ val);
```

```
                                        notify();

                                        Thread.sleep(1000);
                                        }
                              }

                    }
          }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

E:\DiveshA33\POS\Practical 1>javac Threadexample.java

E:\DiveshA33\POS\Practical 1>java Threadexample
Producer produced-0
Consumer consumed-0
Producer produced-1
Consumer consumed-1
Producer produced-2
Consumer consumed-2
Producer produced-3
Consumer consumed-3
Producer produced-4
Consumer consumed-4
Producer produced-5
Consumer consumed-5
Producer produced-6
Consumer consumed-6
Producer produced-7
^C
E:\DiveshA33\POS\Practical 1>
```

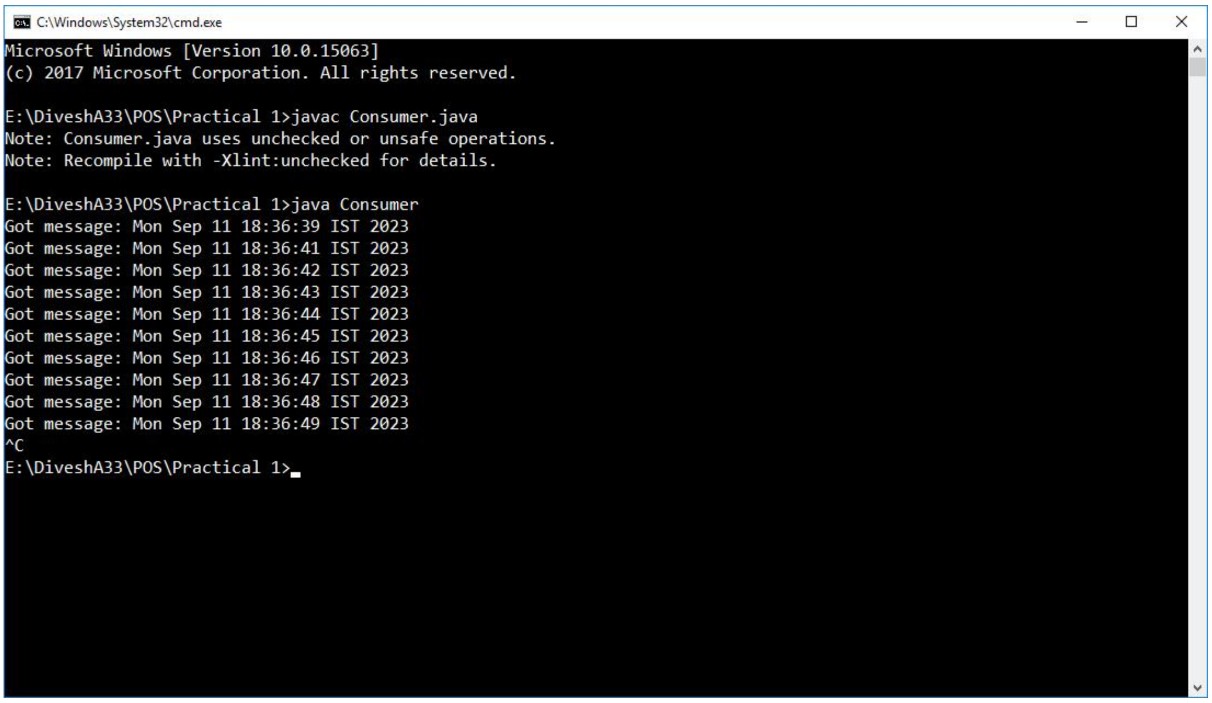
b. Write a program to give a solution to the producer–consumer problem using message passing.

Source Code:

```
import java.util.Vector;

class Producer extends Thread {

        static final int MAX = 7;
        private Vector messages = new Vector();

        @Override
        public void run()
        {
                try
                {
                        while (true){

                                putMessage();
                                sleep(1000);
                        }
                }
                catch (InterruptedException e){
                }

        }
        private synchronized void putMessage() throws InterruptedException {

                while (messages.size() == MAX)
                                wait();

                messages.addElement(new java.util.Date().toString());
                notify();
        }
        public synchronized String getMessage() throws InterruptedException{
                notify();
                while (messages.size() == 0)
                                wait();
                String message = (String)messages.firstElement();

                messages.removeElement(message);
                return message;
```

```
		}
	}

	class Consumer extends Thread{
		Producer producer;

		Consumer(Producer p)
		{
			producer = p;
		}

		@Override
		public void run()
		{
			try
			{
				while (true) {
					String message = producer.getMessage();
					System.out.println("Got message: "+message);
					sleep(2000);
				}
			}
			catch(InterruptedException e){
			}
		}

		public static void main(String args[])
		{
			Producer producer = new Producer();
			producer.start();
			new Consumer(producer).start();
		}


	}
```

Output:



```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

E:\DiveshA33\POS\Practical 1>javac Consumer.java
Note: Consumer.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

E:\DiveshA33\POS\Practical 1>java Consumer
Got message: Mon Sep 11 18:36:39 IST 2023
Got message: Mon Sep 11 18:36:41 IST 2023
Got message: Mon Sep 11 18:36:42 IST 2023
Got message: Mon Sep 11 18:36:43 IST 2023
Got message: Mon Sep 11 18:36:44 IST 2023
Got message: Mon Sep 11 18:36:45 IST 2023
Got message: Mon Sep 11 18:36:46 IST 2023
Got message: Mon Sep 11 18:36:47 IST 2023
Got message: Mon Sep 11 18:36:48 IST 2023
Got message: Mon Sep 11 18:36:49 IST 2023
^C
E:\DiveshA33\POS\Practical 1>
```