

Practical 10

```
1 package package;
2
3 //Shared resource between Producer and Consumer
4 class Shared {
5     int data;
6     boolean hasData = false; // Initially, no data is available
7
8     // Producer puts data
9     synchronized void produce(int value) {
10         while (hasData) {
11             try {
12                 wait();
13             } catch (InterruptedException e) {
14                 e.printStackTrace();
15             }
16         }
17         data = value;
18         System.out.println("Produced: " + data);
19         hasData = true;
20         notify();
21     }
22
23     // Consumer gets data
24     synchronized int consume() {
25         while (!hasData) { // If no data yet, wait
26             try {
27                 wait();
28             } catch (InterruptedException e) {
29                 e.printStackTrace();
30             }
31         }
32         hasData = false;
33         System.out.println("Consumed: " + data);
34         notify(); // Wake up producer
35         return data;
36     }
37 }
38
39 //Producer Thread
40 class Producer extends Thread {
41     Shared shared;
42
43     Producer (Shared s) {
44         shared = s;
45     }
46
47     public void run() {
48         for (int i = 1; i <= 5; i++) {
49             shared.produce(i);
50         }
51     }
52 }
```

```
54
55 //Consumer Thread
56 class Consumer extends Thread {
57     Shared shared;
58
59     Consumer (Shared s) {
60         shared = s;
61     }
62
63     public void run() {
64         for (int i = 1; i <= 5; i++) {
65             shared.consume();
66         }
67     }
68 }
69
70 public class ThreadCommunicationSimple {
71     public static void main(String[] args) {
72         Shared obj = new Shared();
73         Producer p = new Producer(obj);
74         Consumer c = new Consumer(obj);
75
76         p.start();
77         c.start();
78     }
79 }
80
81 Output:
82 Produced: 1
83 Consumed: 1
84 Produced: 2
85 Consumed: 2
86 Produced: 3
87 Consumed: 3
88 Produced: 4
89 Consumed: 4
90 Produced: 5
Consumed: 5
```