

KHAN MOHD. OWAIS RAZA  
20BCD7138

Q.1]

Using runnable interface –

```

1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab12q1;
4  import java.util.concurrent.Semaphore;
5  import javax.lang.model.SourceVersion;
6  class Bounded_Buffer {
7      int x;
8      static Semaphore semC = new Semaphore(0);
9      static Semaphore semP = new Semaphore(1);
10     void get() {
11         try {
12             semC.acquire();
13         }
14         catch (InterruptedException e) {
15             System.out.println("INTERRUPTED EXCEPTION IS CAUGHT");
16         }
17         System.out.println("CONSUMER CONSUMED: " + x);
18         semP.release();
19     }
20     void put(int x) {
21         try {
22             semP.acquire();
23         }
24         catch (InterruptedException e) {
25             System.out.println("INTERRUPTED EXCEPTION IS CAUGHT");
26         }
27         this.x = x;
28         System.out.println("PRODUCER PRODUCED: " + x);
29         semC.release();
30     }
31
32     public SourceVersion getSupportedSourceVersion() {
33         return SourceVersion.latest();
34     }
35 }
36 class Producer implements Runnable{
37     Bounded_Buffer q;
38     Producer(Bounded_Buffer q) {
39         this.q = q;
40         new Thread(this, "PRODUCER").start();
41     }

```

```

43     public void run() {
44         for (int i = 0; i < 11; i++){
45             q.put(i);
46         }
47     }
48     public class Producer_Consumer_Problem implements Runnable {
49         Bounded_Buffer q;
50         Producer_Consumer_Problem(Bounded_Buffer q) {
51             this.q = q;
52             new Thread(this, "CONSUMER").start();
53         }
54         public void run() {
55             for (int i = 0; i < 11; i++){
56                 q.get();
57             }
58         }
59     }
60     class BoundedBuffer{
61         public static void main(String args[]){
62             Bounded_Buffer q = new Bounded_Buffer();
63             new Producer_Consumer_Problem(q);
64             new Producer(q);
65         }
66     }

```

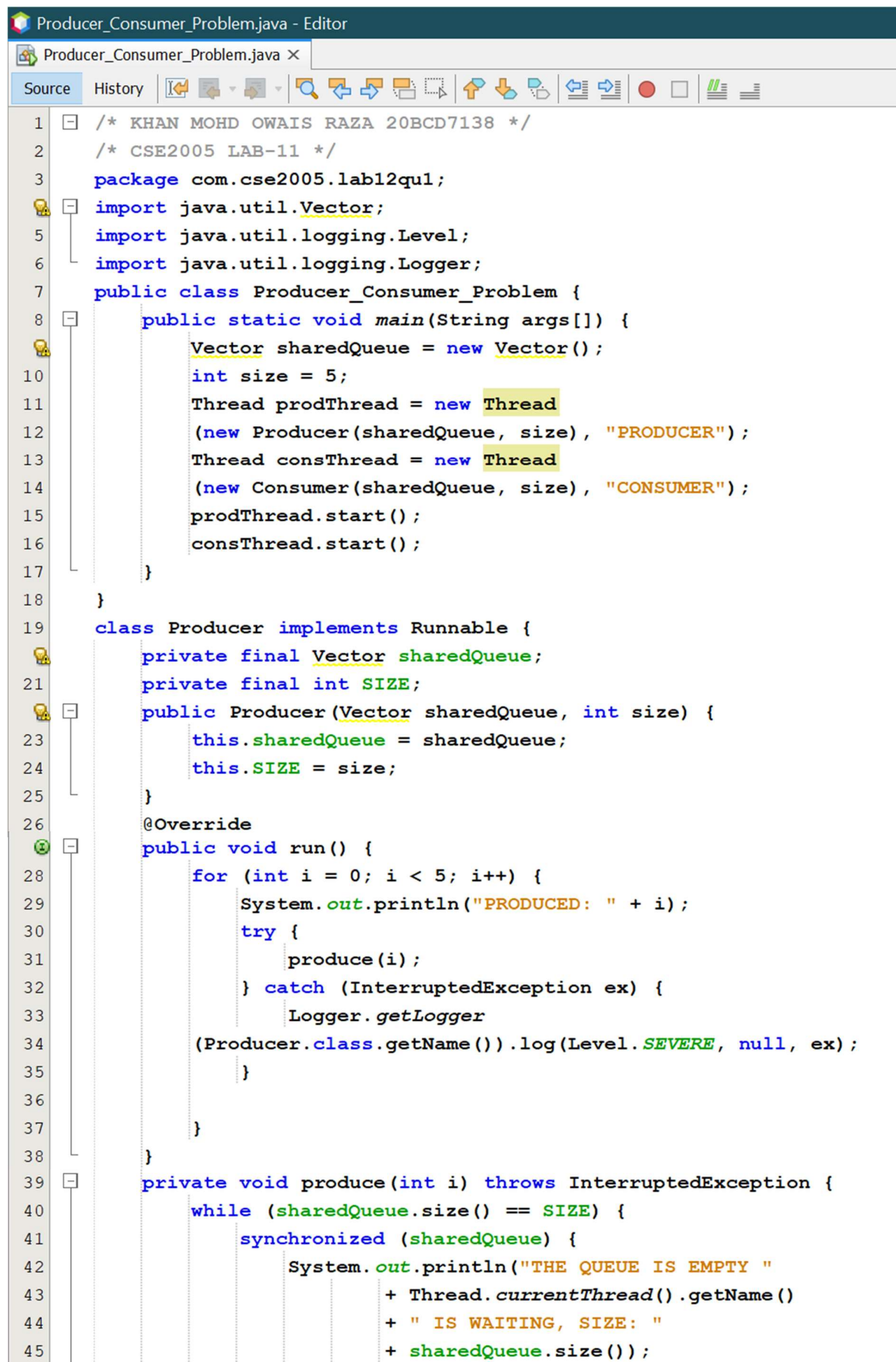
Output console –

```

Output - Run (Lab12Q1) - Editor
Output - Run (Lab12Q1) x
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab12Q1 ---
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
CONSUMER CONSUMED: 7
PRODUCER PRODUCED: 8
CONSUMER CONSUMED: 8
PRODUCER PRODUCED: 9
CONSUMER CONSUMED: 9
PRODUCER PRODUCED: 10
CONSUMER CONSUMED: 10
-----
BUILD SUCCESS
-----
Total time: 1.226 s
Finished at: 2021-12-22T23:08:19+05:30
-----

```

Another technique using runnable interface –



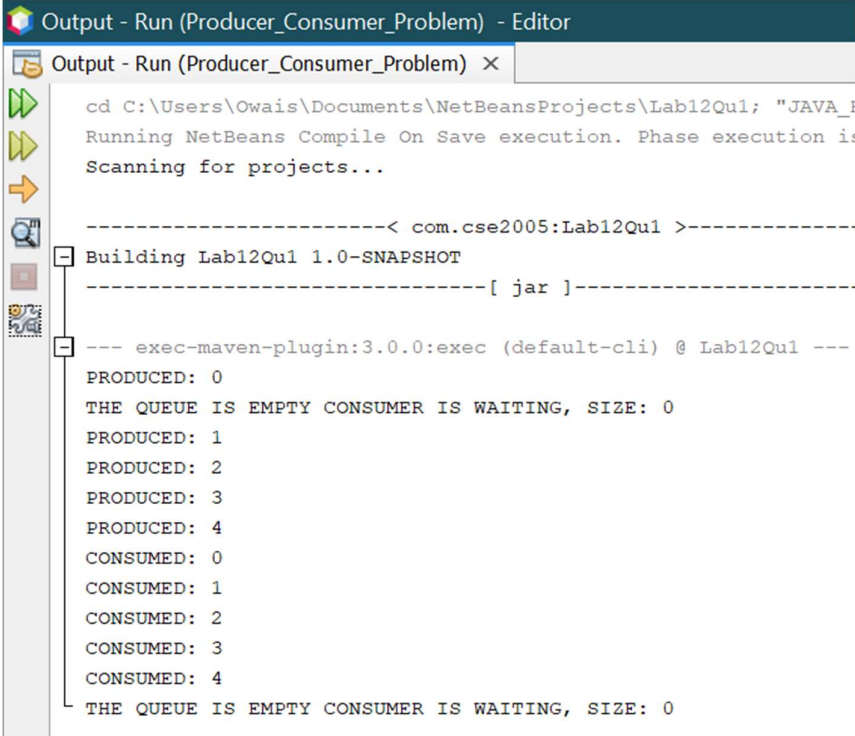
```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab12qul;
4  import java.util.Vector;
5  import java.util.logging.Level;
6  import java.util.logging.Logger;
7  public class Producer_Consumer_Problem {
8      public static void main(String args[]) {
9          Vector sharedQueue = new Vector();
10         int size = 5;
11         Thread prodThread = new Thread
12             (new Producer(sharedQueue, size), "PRODUCER");
13         Thread consThread = new Thread
14             (new Consumer(sharedQueue, size), "CONSUMER");
15         prodThread.start();
16         consThread.start();
17     }
18 }
19 class Producer implements Runnable {
20     private final Vector sharedQueue;
21     private final int SIZE;
22     public Producer(Vector sharedQueue, int size) {
23         this.sharedQueue = sharedQueue;
24         this.SIZE = size;
25     }
26     @Override
27     public void run() {
28         for (int i = 0; i < 5; i++) {
29             System.out.println("PRODUCED: " + i);
30             try {
31                 produce(i);
32             } catch (InterruptedException ex) {
33                 Logger.getLogger
34                     (Producer.class.getName()).log(Level.SEVERE, null, ex);
35             }
36         }
37     }
38 }
39 private void produce(int i) throws InterruptedException {
40     while (sharedQueue.size() == SIZE) {
41         synchronized (sharedQueue) {
42             System.out.println("THE QUEUE IS EMPTY "
43                 + Thread.currentThread().getName()
44                 + " IS WAITING, SIZE: "
45                 + sharedQueue.size());
```

```

46         sharedQueue.wait();
47     }
48 }
49 synchronized (sharedQueue) {
50     sharedQueue.add(i);
51     sharedQueue.notifyAll();
52 }
53 }
54 }
55
56 class Consumer implements Runnable {
57
58     private final Vector sharedQueue;
59     private final int SIZE;
60
61     public Consumer(Vector sharedQueue, int size) {
62         this.sharedQueue = sharedQueue;
63         this.SIZE = size;
64     }
65
66     @Override
67     public void run() {
68         while (true) {
69             try {
70                 System.out.println("CONSUMED: " + consume());
71                 Thread.sleep(10);
72             }
73             catch (InterruptedException ex) {
74                 Logger.getLogger
75                     (Consumer.class.getName()).log(Level.SEVERE, null, ex);
76             }
77         }
78     }
79     private int consume() throws InterruptedException {
80         while (sharedQueue.isEmpty()) {
81             synchronized (sharedQueue) {
82                 System.out.println("THE QUEUE IS EMPTY "
83                     + Thread.currentThread().getName()
84                     + " IS WAITING, SIZE: " + sharedQueue.size());
85
86                 sharedQueue.wait();
87             }
88         }
89         synchronized (sharedQueue) {
90             sharedQueue.notifyAll();
91             return (Integer) sharedQueue.remove(0);
92         }
93     }
94 }

```

## Output console –



The image shows a screenshot of the NetBeans IDE's Output Console window. The window has a title bar that reads "Output - Run (Producer\_Consumer\_Problem) - Editor". Below the title bar is a tab labeled "Output - Run (Producer\_Consumer\_Problem) X". On the left side of the console, there is a vertical toolbar with icons for running, debugging, and other actions. The main area of the console displays the following text:

```
cd C:\Users\Owais\Documents\NetBeansProjects\Lab12Qu1; "JAVA_F
Running NetBeans Compile On Save execution. Phase execution is
Scanning for projects...

-----< com.cse2005:Lab12Qu1 >-----
Building Lab12Qu1 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab12Qu1 ---
PRODUCED: 0
THE QUEUE IS EMPTY CONSUMER IS WAITING, SIZE: 0
PRODUCED: 1
PRODUCED: 2
PRODUCED: 3
PRODUCED: 4
CONSUMED: 0
CONSUMED: 1
CONSUMED: 2
CONSUMED: 3
CONSUMED: 4
THE QUEUE IS EMPTY CONSUMER IS WAITING, SIZE: 0
```



Q.2]

```
CSE2005_LAB12_QUESTION2.java - Editor
CSE2005_LAB12_QUESTION2.java x
Source History

1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab12que2;
4  class CONVERSATION{
5      boolean flag = false;
6      public synchronized void Herbert(String msg){
7          if (flag){
8              try{
9                  wait();
10             }
11             catch (InterruptedException e){
12                 e.printStackTrace();
13             }
14         }
15         System.out.println("RAZA:"+msg);
16         flag = true;
17         notify();
18     }
19     public synchronized void Schildt(String msg){
20         if (!flag){
21             try {
22                 wait();
23             }
24             catch (InterruptedException e){
25                 e.printStackTrace();
26             }
27         }
28         System.out.println("KELLY:"+msg);
29         flag = false;
30         notify();
31     }
32 }
33 class USER1 implements Runnable
34 {
35     CONVERSATION m;
36     String a[]={ "HI, GOOD AFTERNOON",
37                 "I WAS THINKING TO MMET YOU ?" ,
38                 "LET'S SPEND TIME AT MY HOME" ,
39                 "GOOD NIGHT, WILL MEET YOU TOMORROW"};
40     public USER1(CONVERSATION m1){
41         this.m=m1;
42         new Thread(this,"USER1").start();
43     }
}
```

```

46     }
47     }
48 }
49 }
50 class USER2 implements Runnable{
51     CONVERSATION m;
52     String b[]={ "GOOD AFTERNOON",
53                 "YES!!! LET'S SPEND SOME TIME" ,
54                 "OK, SURE!!" ,
55                 "GOOD NIGHT, TAKE CARE"};
56     public USER2(CONVERSATION m1){
57         this.m=m1;
58         new Thread(this,"USER2").start();
59     }
60     public void run(){
61         for (int i = 0; i < b.length; i++){
62             m.Schildt(b[i]);
63         }
64     }
65 }
66 public class CSE2005_LAB12_QUESTION2 {
67     public static void main(String[] args){
68         CONVERSATION x=new CONVERSATION();
69         new USER1(x);
70         new USER2(x);
71     }
72 }

```

Output - Run (CSE2005\_LAB12\_QUESTION2) - Editor

Output - Run (CSE2005\_LAB12\_QUESTION2) X

```

cd C:\Users\Owais\Documents\NetBeansProjects\Lab12Que2; "JAVA_HOME=C:\\Pr
Running NetBeans Compile On Save execution. Phase execution is skipped an
Scanning for projects...

-----< com.cse2005:Lab12Que2 >-----
Building Lab12Que2 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab12Que2 ---
RAZA:HI, GOOD AFTERNOON
KELLY:GOOD AFTERNOON
RAZA:I WAS THINKING TO MMET YOU ?
KELLY:YES!!! LET'S SPEND SOME TIME
RAZA:LET'S SPEND TIME AT MY HOME
KELLY:OK, SURE!!
RAZA:GOOD NIGHT, WILL MEET YOU TOMORROW
KELLY:GOOD NIGHT, TAKE CARE

BUILD SUCCESS

Total time: 1.249 s
Finished at: 2021-12-23T00:35:54+05:30

```

Another technique (using scanner function) –



```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-11 */
3  package com.cse2005.lab12q2;
4  import java.util.Scanner;
5  class CONVERSATION{
6      Scanner sc=new Scanner(System.in);
7      String message;
8      final Object rec = new Object();
9      final Object send = new Object();
10     boolean msg_send = false;
11     boolean msg_rcv = false;
12     void rcvMessage(){
13         synchronized (send){
14             while(!msg_send) {
15                 try{
16                     send.wait();
17                 }catch(InterruptedException e){
18                     System.out.println("TNTERRUPTED EXCEPTION CAUGHT");
19                 }
20             }
21             msg_send = false;
22         }
23         synchronized (rec) {
24             String name = Thread.currentThread().getName();
25             System.out.println(name);
26             String otherName =
27                 name.contentEquals("USER-1") ? "USER-2" : "USER-1";
28             System.out.println(name + "(" + otherName + "):" + message);
29             System.out.println("----- RECIEVED -----");
30             msg_rcv = true;
31             rec.notify();
32         }
33     }
34     void sendMessage(){
35         synchronized (send) {
36             String name = Thread.currentThread().getName();
37             System.out.println(name);
38             String otherName =
39                 name.contentEquals("USER-1") ? "USER-2" : "USER-1";
40             System.out.print(name + "(" + otherName + "):");
41             message = sc.nextLine();
42             System.out.println("SEND");
43             msg_send = true;
44             send.notify();
45         }
46     }
47 }
```

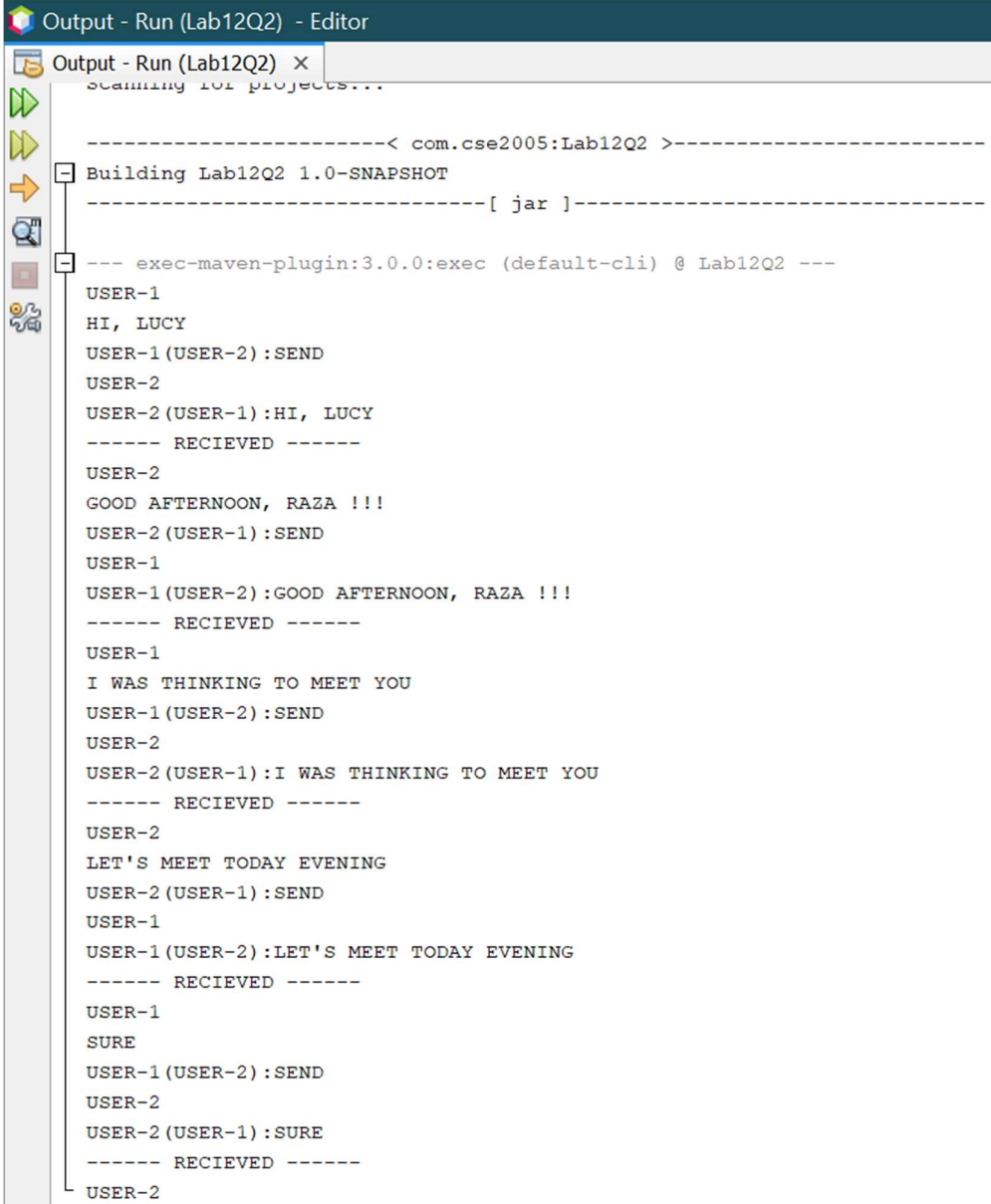


```

46         synchronized (rec) {
47             while (!msg_rcv) {
48                 try {
49                     rec.wait();
50                 }
51                 catch (InterruptedException e) {
52                     System.out.println("INTERRUPTED EXCEPTION CAUGHT");
53                 }
54             }
55             msg_rcv = false;
56         }
57     }
58 }
59 class Person1 implements Runnable{
60     CONVERSATION ex;
61     public Person1 (CONVERSATION ex) {
62         this.ex = ex;
63         Thread u2=new Thread(this, "USER-1");
64         u2.start();
65     }
66     public void run() {
67         while(true) {
68             ex.sendMessage();
69             ex.recvMessage();
70         }
71     }
72 }
73 class Person2 implements Runnable{
74     CONVERSATION ex;
75
76     public Person2 (CONVERSATION ex) {
77         this.ex=ex;
78         Thread u1=new Thread(this, "USER-2");
79         u1.start();
80     }
81     public void run() {
82         while(true) {
83             ex.recvMessage();
84             ex.sendMessage();
85         }
86     }
87 }
88 public class Main{
89     public static void main(String args[])
90     {
91         CONVERSATION ex =new CONVERSATION();
92         new Person1(ex);
93         new Person2(ex);
94     }
95 }

```

Output console –



```
Output - Run (Lab12Q2) - Editor
Output - Run (Lab12Q2) x
Scanning for projects...

-----< com.cse2005:Lab12Q2 >-----
Building Lab12Q2 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab12Q2 ---
USER-1
HI, LUCY
USER-1(USER-2):SEND
USER-2
USER-2(USER-1):HI, LUCY
----- RECIEVED -----
USER-2
GOOD AFTERNOON, RAZA !!!
USER-2(USER-1):SEND
USER-1
USER-1(USER-2):GOOD AFTERNOON, RAZA !!!
----- RECIEVED -----
USER-1
I WAS THINKING TO MEET YOU
USER-1(USER-2):SEND
USER-2
USER-2(USER-1):I WAS THINKING TO MEET YOU
----- RECIEVED -----
USER-2
LET'S MEET TODAY EVENING
USER-2(USER-1):SEND
USER-1
USER-1(USER-2):LET'S MEET TODAY EVENING
----- RECIEVED -----
USER-1
SURE
USER-1(USER-2):SEND
USER-2
USER-2(USER-1):SURE
----- RECIEVED -----
USER-2
```