Terna Engineering College
Computer Engineering Department
Program: Sem VIII

Course: Distributed Computing Lab (CSL802)

Faculty: Rohini Patil

Experiment No. 6

A.1 Aim: To Implement Token Ring Mutual Exclusion Algorithm.

PART B
(PART B: TO BE COMPLETED BY STUDENTS)

Roll No. 50	**Name:** AMEY MAHENDRA THAKUR
Class: BE COMPS B 50	**Batch:** B3
Date of Experiment: 18-02-2022	**Date of Submission:** 18-02-2022
Grade:	

B.1 Software Code written by a student:

- **TokenRing.java**

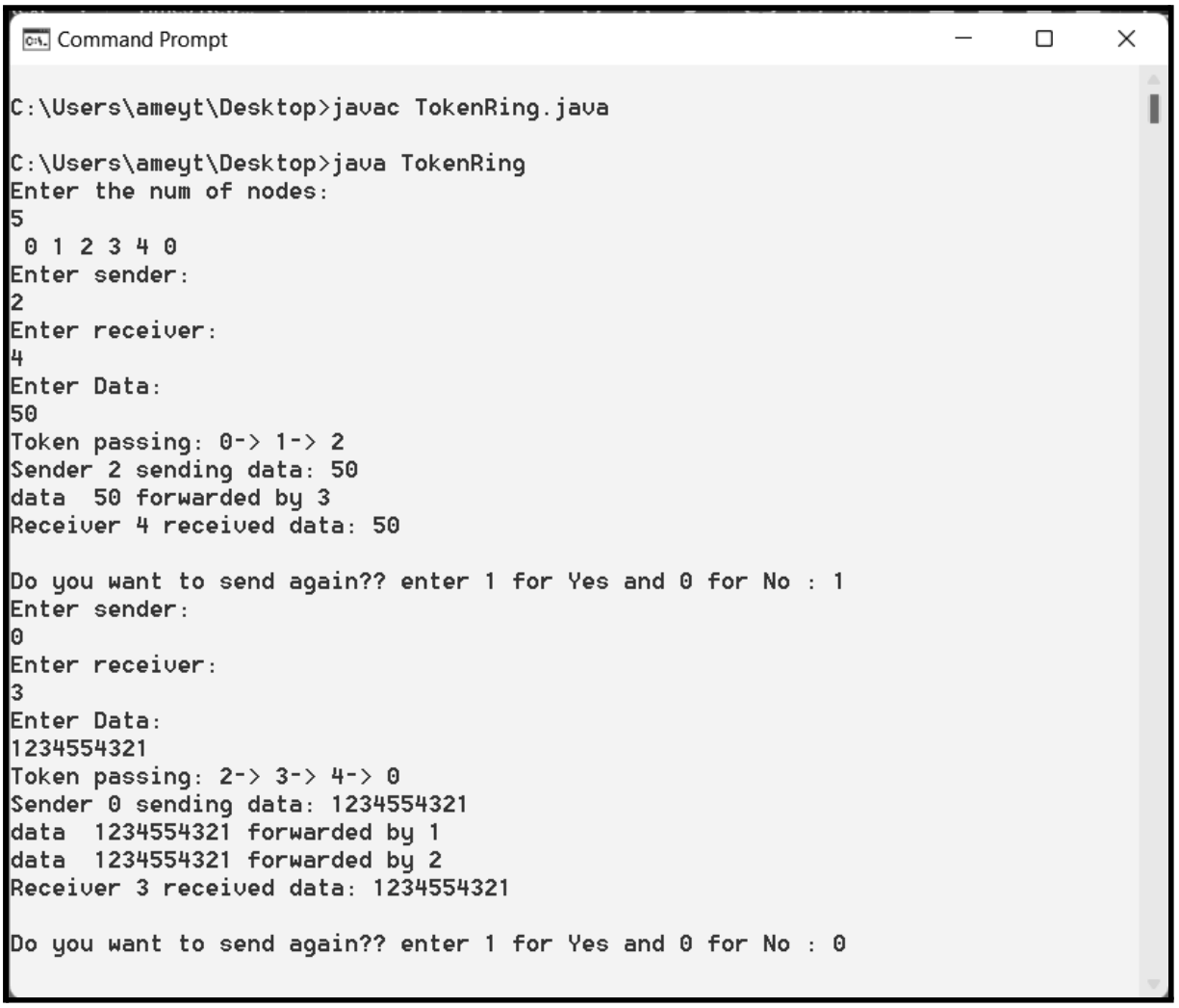
```
import java.io.*;
import java.util.*;

class TokenRing {

   public static void main(String args[]) throws Throwable {
      Scanner scan = new Scanner(System.in);
      System.out.println("Enter the num of nodes:");
      int n = scan.nextInt();
      int m = n - 1;
      // Decides the number of nodes forming the ring
      int token = 0;
      int ch = 0, flag = 0;
      for (int i = 0; i < n; i++) {
         System.out.print(" " + i);
      }
```

```
      System.out.println(" " + 0);
      do{
         System.out.println("Enter sender:");
         int s = scan.nextInt();
         System.out.println("Enter receiver:");
         int r = scan.nextInt();
         System.out.println("Enter Data:");
         int a;
         a = scan.nextInt();
         System.out.print("Token passing:");
         for (int i = token, j = token; (i % n) != s; i++, j = (j + 1) % n) {
            System.out.print(" " + j + "->");
         }
         System.out.println(" " + s);
         System.out.println("Sender " + s + " sending data: " + a);
         for (int i = s + 1; i != r; i = (i + 1) % n) {
            System.out.println("data  " + a + " forwarded by " + i);
         }
         System.out.println("Receiver " + r + " received data: " + a +"\n");
         token = s;
         do{
            try {
               if( flag == 1)
         System.out.print("Invalid Input!!...");
                  System.out.print("Do you want to send again?? enter 1 for Yes and 0 for
No : ");
               ch = scan.nextInt();
               if( ch != 1 && ch != 0 )
         flag = 1;
               else
         flag = 0;
            } catch (InputMismatchException e){
               System.out.println("Invalid Input");
            }
         }while( ch != 1 && ch != 0 );
      }while( ch == 1 );
   }
}
```

B.2 Input and Output:



B.3 Observations and learning:

Token Ring algorithm achieves mutual exclusion in a distributed system by creating a bus network of processes. A logical ring is constructed with these processes and each process is assigned a position in the ring. Each process knows who is next in line after itself.

The algorithm works as follows:

- When the ring is initialised, process 0 is given a token.
- The token circulates around the ring.
- When a process acquires the token from its neighbour, it checks to see if it is attempting to enter a critical region. If so, the process enters the region, does all the work it needs to, and leaves the region.
- After it has exited, it passes the token to the next process in the ring. It is not allowed to enter the critical region again using the same token. If a process is handed the token by its neighbour and is not interested in entering a critical region, it just passes the token along to the next process.

B.4 Conclusion:

We have successfully implemented the Token Ring Mutual Exclusion Algorithm using Java.

B.5 Question of Curiosity:

Q1: For each critical section (CS) execution, Ricart-Agrawala algorithm requires _____ messages per CS execution and the Synchronization delay in the algorithm is _____.

A. 3(N – 1), T
B. 2(N – 1), T
C. (N – 1), 2T
D. (N – 1), T

ANS: B. 2(N – 1), T

Q2: "In Suzuki-Kasami's Broadcast Algorithm, if a site does not hold the token when it makes a request, the algorithm requires 5N-1 messages to obtain the token."

A. True
B. False

ANS: B. False

Q3: In the token passing approach of distributed systems, processes are organized in a ring structure

A. logically
B. physically
C. both logically and physically
D. none of the mentioned

ANS: A. logically

Q4: According to the ring algorithm, links between processes are

A. bidirectional
B. unidirectional
C. both bidirectional and unidirectional
D. none of the mentioned

ANS: B. unidirectional