lab -10 open Ended Exercise

• Demonstrate Inter process communication and

```
class Q{
 int n
 Synchronized int get(){
    System.out.println("Got:" +n);
    return n; }
 Synchronized void put(int n){
    this.n=n;
    System.out.println("Put "+n);
 }}

class Producer implements Runnable {
    Q p;
    Producer(Q p){
       this.q=q;
       new Thread(this, "Producer").start();
    }
    public void run(){
       int i=0
       while(i<8){
         q.put(i++);
       }}}
class Consumer implements Runnable{
   Q q;
   Consumer(Q q){
      this.q=q;
      new Thread(this, "consumer").start()
   }
```

```
public void run (){
  int i = 0
  while (i<8) {
  int r = q.get();
    i++;
  } }}

class PC{
  public static void main (String args[]){
  System.out.println ("Santhosh S");
  Q q = new Q();
  new Producer(q);
  new Consumer (q);
  System.out.println(" Press Control-c
  to stop. "); }}
```

output : Santhosh S 2023BIMB2586

Put 0
Got 0
Put 1
Got 1
Put 2
Got 2
put 3
Got 3
put 4
Put 4
put 5
Got 5

13.02.24

13-12-2024

10) 2. Deadlock

```
class A{
   Synchronized void foo( B b) {
     String name= Thread. Current Thread (). get
     System.out.println ("name + " entered A. fo
    try{
     thread. Sleep (1000);
     }
     Catch (Exception e){
      System.out. println (" A interrupted ");
     }
    System.out. println (name + "trying to call B.
     b.last ();
   }
  void last(){
     System. out.println ("Inside A. last");
    }
}
class B{
   Synchronized void bar (A a) {
     String name = Thread. Current Thread. get Name(
     System.out.println (name + " entered B.bar")
    try{
      Thread. Sleep (millis : 1000);
     }
     Catch (Exception e){
        System. out.println (" B is Interrupted");
     }
     System. out.println (name + "trying to call A.b
    a.last();
   }
```

```
void last() {
    System.out.println("Inside A.last");
}
}

public class Deadlock implements Runnable{
    A a = new A();
    B b = new B();
    Deadlock() {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this, "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in mainthread");
    }
    public void run() {
        b.bar(a);
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) {
        new Deadlock();
    }
}
```

output

```
Racing thread Entered B.bar
MainThread enter A.foo
Main Thread to call B.last
inside A.last
Racing thread trying to call A.last()
inside A.last
Back in other thread
Back in main thread.
```

13.02.24