6-02-24

LAB-10

Demonstrate interprocess communication & dead lock

```
class Q {
 int n;
 boolean valueSet= false;
 synchronised int set(){
 while (!valueSet)
  try {
 system.out. println("\n Consumer waiting \n");
  wait();
 } catch (InterruptedException e) {
 System.out .println("Interrupted Exception caught");
 }
 System.out println (" Got: "+n);
   value set= false;
 system.out println("\n Intimate Producer\n");
 notify ();
 return n;
 }
 Synchronized void put (int n) {
 while(value set)

try {
      System.out println("\n Producer waiting \n");
  wait();
 } catch (Interrupted Exception e) {
  System.out println(" Interrupted Exception caught");
  }
   this.n = n;
  valueSet = true;
  System.out println("Put "+n);
System.out.println("\nIntimate Consumer\n");
notify();
}
}
class Produce implements Runnable{

Q q;
Produce (Q q) {
this.q = q;

new Thread(this, "Produce").start();
}
public void run() {
int i = 0;

while (i<15) {
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<15) {
int r = q.get();
System.out.println("consumed" +r);
i++;
}
}
}
```

```
}
class Main {

public static void main( String args[]) {
Q q = new Q();

new producer (q);
new Consumer (q);

System.out.println(" press Control-C to stop");
 }
 }
```