LAB 10.

Interprocess comunication.

```
class Q
{
        int n;
        boolean value set = false;
        Synchronized int get()
        {
         while (!value set)
         try
         {
          System.out.println ("\n Consumer waiting\n");
          wait();
          }
          catch (InterruptedException e)
          {
           System.out.println ("InterruptedException
          caught");
          }.
          System.out.println ("Got:" + n);
         valueSet = false;
          System.out.println ("\n Intime Producer\n");
          notify();
          return n;
        }.
Synchronized void put(int n)
{
        while (valueSet)
        try
        {
        System.out.println ("\n Producer waiting \n");
        wait();
        }
catch (InterruptedException e)
{
 System.out.println ("InteruptedException caught");
}
 this.n = n;
 valueSet = true;
 System.out.println ("Put :" +n);
 System.out.println ("\n Intime consumer\n");
 notify();
 }
}
class Producer implements Runnable
{
      Q q;
      Producer (Q q)
      {
         this.q = q
         new Thread (this , "Producer"). start();
      }
      public void run()
       {
         int i = 0;
         while( i < 5)
         {
             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 PCFixed
    {
          public static void main( String args[])
          {
              Q q = new Q();
              new Producer (q);
              new Consumer (q);
              System.out.println ("Press Control-C to
               stop.");
            }
    }
```

output:

```
Put:0
Intimate consumer
Producer waiting
Got:0
Intimate Producer
Put:1
Intimate consumer
Producer waiting
Got:1
Intimate Producer
Put:2
Intimate Consumer
Producer waiting
Got : 2
Intimate Producer
Put : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Producer
Put :4
Intimate consumer
Producer waiting
Got :4
Intimate Producer.
```

output :

```
Put :0
Intimate consumer
Producer waiting
Got :0
Intimate Producer
Put :1
Intimate consumer
Producer waiting
Got : 1
Intimate Producer
Put : 2
Intimate Consumer
Producer waiting
Got : 2
Intimate Producer
Put : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Producer
Put :4
Intimate consumer
Producer waiting
Got :4
Intimate Producer.
```