Week - 10 : Open Ended Exercise II

Date 19,12,24
Page

Q] Demonstrate "Inter process Communication & deadlock

```
→ //deadlock
class A {
    synchronized void Foo(B b) {
        String name = Thread.currentThread().getName();
        System.out.println(name + " entered A.Foo");
        try {
            Thread.sleep(1000);
        } catch (Exception e) {
            System.out.println("A Interrupted");
        }
       System.out.println(name+ "trying to call B.last()");
        b.last();
    }
    synchronised void last() {
        System.out.println("Inside A.last.");
    }
}

class B {
   synchronised void bar(A a) {
      String name = Thread.currentThread.getName();
      system.out.printh(name + " entered B.bar");
      try {
          Thread.sleep(1000);
      } catch (Exception e) {
          System.out.println("B interupted");
      }
    System.out.println("name + " trying to call A.last()");
    a.last();
  }
synchronized void last () {
  system. out. println ("Inside B. last");
  }
}

class Deadlock implements Runnable {
  A a = new A ();
  B b = new B ();
  Thread t;

  Deadlock () {
   Thread. currentThread. setName ("Main Thread");
     t = new Thread (this, "Racing Thread");
  }

  void deadlockStart () {
    t. start ();
    a. foo (b);
    System. out. println ("Back in main thread");
  }

  public void run () {
    b. bar (a);
    System. out. println ("Back in other thread");
  }
 public static void main (String xx[]) {
   Deadlock d1 = new Deadlock ();
    d1.deadlockStart ();
 }
}
```

Date 19/12/24
Page

```
// Interprocess Communication

Class Q {
    int n; boolean valueSet = false;
    synchronized int get() {
        while (!valueset) {
            try { wait(); }
            catch (InterruptedException e) {
                System.out.println(e.getMessage());
            }
        System.out.println("Got : +n);
        valueSet = false;
        notify();
        return n;
    }

    Synchronized void put (int n) {
        while (valueSet) {
            try { wait(); }
            catch (InterruptedException e) {
                system.out.println("Int. excep. caught");
            }
        this.n = n;
        valueset = true;
        System.out.println("Put : " +n);
        notify();
    }
}

class Producer implements Runnable {
    Q q; Thread t;
    producer (Q q) {
        this.q = q;
        t = new Thread (this, "Producer");
    }
public void run() {
        int i=0;
        while (true) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
        Q q; Thread t;
        Consumer (Q q) {
            this.q = q;
            t = new Thread (this, "consumer");
        }
        public void run() {
            while (true) {
                q.get();
            }
        }
}
class PCFixed {
        public static void main (String xx[]) {
            Q q = new Q();
            Producer p = new producer (q);
            Consumer c = new consumer(q);
            p.t.start();
            c.t.start();
            system.out.println("Press Ctrl c to stop");
        }
}
```