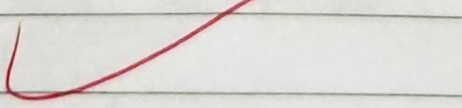


13/02/2024

2. Java program to demonstrate deadlock.

Ans.

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




```

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

```

```

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.demo(b); // get a lock on 'a' in this thread
        System.out.println("Back in main thread");
    }
    public void run() {
        b.bar(a); // get lock on b in the other thread
        System.out.println("Back in other thread");
    }
}

```



```

public static void main (String args[]) {
    new Deadlock();
}

```

Output :

~~RacingThread entered B. bar~~
~~MainThread entered A. demo~~
~~Inside A. last()~~
~~Back in other thread~~
~~Inside~~

MainThread entered A. demo
 RacingThread entered B. bar
 MainThread trying to call B. last()
 RacingThread trying to call A. last()
 Inside B. last()
 Back in main thread
 Inside A. last()
 Back in other thread

Run
 13/2/2024