

13/2/24

=>

Program to Demonstrate 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()");

}

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, "Racing Thread");
        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 args[]) {
    new Deadlock();
}
```

[Scribbled out text]

Output -

Main Thread entered A.foo
Racing Thread entered B.bar
Main thread trying to call B.last()
Inside A.last
Back in main thread
Racing thread trying to call A.last()
Inside A.last
Back in other thread.

[Handwritten signature]
13.02.24

Shreyas.k
BM22CS271

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]

[Faint handwritten notes]