Lab 10 : deadlock

Lab 10 : dea	HOLOCK
3	
	Prom 8
THE STREET	Pram-9 — Dead lock !-
-	class A
1	Synchronizal
1	Synchronized void too (B, b) 1 String name = Thread (Urron)
1	system. out : mill corrent Thread ()
1	string name = Thread. current Thread () get Now ():  try ! contared A. foo");
Value of the same	Thread - Steen Const.
	System (Exception e)
	System out - println ("A Interrupted");
	interrupted );
	b. Last (); print In (name + "trying to call B. last)"
-	b. last ();
-	The state of the s
	Void 1034 () 1
- Mars	System. out - prin+tn-(" Inside A. last");
	4
	Charles - Charles of the Asian Asian Asian
	Class B 1
	Synchorized void bar (A a) {
	String name = Thread. current Thread () get Name);  system.out. println (name + "entered B. bar");
	system out printin change a dieter
	try 1
	Thread surp (1000);
	System out println ("B Inkrupted");
	system. out - prantin (name + " trying to call A
	system. out. prostin
	lasty
	a. (as+1);
	7
	void (an+(1)?

system . out . println ( Inside a last) class beadlock implements ponnible ! A a = new A(); B 6 = nen B(); Dead lock of Thread - current Thread () - Set Name (" main thread" Thread + = new thread (this "Racking thread") +. Start (11 a.foolb); system out print in ("Back in win thread"). poblic wid ron () 6-bar(a); system out printle ("Back in other throat) public static void Main (string args 17) { Me Dead lock (1: Mainthreador entered A. 400 Bacherthread entered B. bar heinthread trying to call B. kest U Inside A last Back in Main Howard Rading thread trying to out A-last () Inside A. 457 Back to other thread.

```
Code:
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();
}
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 = new A();
Bb = new B();
Deadlock() {
Thread.currentThread().setName("M
ainThread");
Thread t = new Thread(this,
"RacingThread");
t.start();
a.foo(b); // get lock on a in this
thread.
System.out.println("Back in main
thread");
```

```
public void run() {
b.bar(a); // get lock on b in other
thread.
System.out.println("Back in other
thread");
}
public static void main(String args[]) {
new Deadlock();
}
Output:
 MainThread entered A.foo
 RacingThread entered B.bar
 RacingThread trying to call A.last()
 MainThread trying to call B.last()
 Inside A.last
 Back in main thread
 Back in other thread
```