Lab-10(b)
DEADLOCK

1)
```
class A
{
 synchronized void foo(B b)
 {
   String name = Thread.currentThread().
                            .getName();
   ~~System.out.println("A Interrupted");~~
 } 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();
 }
}
```

```
class B
{
  Synchronized void bar(A a)
  {
    String name = Thread. currentThread()
                         .getName();
    System.out.println(name + "enter
                -ed 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");
  }

}
```

PAGE
DATE

```
class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock()
    {
        Thread.currentThread().setName
            ("Main Thread");
        Thread t = new Thread(this,
                    "RacingThread");
        t.start();
        a.foo(b);
        ~~thread~~
        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();
    }

}
```

⟹ OUTPUT:-

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

[illegible]
13.02.24