LAB-10

Date 27/11/24
Page

```
class A
{ synchronized void foo(B b)
  {
    String name = Thread.currentThread().getName();
    SOP(name + " entered A.foo");

    try
    {
      Thread.sleep(1000);
    }
    catch(Exception e)
    { SOP("A Interrupted");
    }

    SOP(name + "trying to call B.last()");
    b.last();
  }

synchronized void last()
{
  SOP("Inside A.last()");
}
}

class B{
  synchronized void bar(A a)
  { String name = Thread.currentThread().getName();
    SOP(name + "entered B.bar");

    try { Thread.sleep(1000); }
      catch(Exception e)
      { SOP("B Interrupted");
    }
```

```
SOP (name + "trying to call A.last()");
a.last();
}

synchronized void last()
{
    SOP ("Inside B.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.foo(b);
        SOP("Back in main thread");
    }

    public void run()
    { b.bar(a)
        SOP("Back in other thread");
    }

    PSVM
    {
        SOP("VSN : Name : ");
    } new Deadlock();
}
```