LAB-10

Demonstrate inter process communication & deadlock.

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

  try { Thread.sleep (1000);
  } catch (Exception e)
  { S.O.P ("A Interrupted");
  }
    sop (name + " trying to call B.last()");
    b.last();
  }
  void last()
  {
    S.O.P ("Inside A.last");
  }
}

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

```
Thread.sleep (1000);
}
catch (Exception e)
{ sop ("B Interrupted");
}
sop (name + "trying to call A.last()");
a.last();
}
void last(){
  Sop ("Inside B.last");
}}

class Deadlock implements Runnable {
    A a = new A();
    B b = new B();

Deadlock()
{ Thread.currentThread().setName ("Main Thread");
  Thread t = new Thread (this, "Racing Thread");
    t.start();
    a.foo(b);
    s.o.p ("Back in main thread");
  }

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

public static void main (String args[])
{ s.o.p ("Name: Akusha gp ; USN: 1BM23CS005");
  new Deadlock();
}}
```

O/P:- RacingThread entered B.bar
MainThread Entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Inside B.last

Back in main thread.
Back in other thread.

Seen
03/12/24