LAB-10   2.

Deadlock.

```
class A
{
    Synchronized void foo(B b)
    {
      String name = Thread.currentThread().getNam
    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  bas (A a)
  {
  String name = Thread.currentThread().
      getName();
```

```
System.out.pri
try
{
 Thread.slee
}
catch (Exce
{
  System .ou
}.
System.out.p
   A.last()"
 a.last();
}.
void last()
{
   System.o
 }
}.
class Deadl
{
   A a = ne
   B b = ne
   Deadloc
   {
   Thread.
   Thread
   t.sta
   a.foo
   Syster
   }.
  public
  {
     b.
```

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()");
          b.last();
        }.
        void last()
        {
             System.out.println ("Inside A.last()
        }
}.
 class B
 {
         synchronized void bas (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, "RacingThread");
   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();
    }
}
```

<u>output</u> :

```
  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 )
 Insid  A.last
   Back  in  other  thread
```

13/2/24