# Dead Lock:

**Dead Lock is a situation , where more than one thread is depending on each other in circular dependency.**

**In java applications, once we are getting deadlock then program will be stucked in middle, so that, it will not have any recovery mechanisms, it will have only prevention mechanisms and once deadlock is occurred we are unable to do anything for that.At max we may just avoid the deadlock situations.**

**Coding Example:**

class Register\_Course extends Thread

{

Object course\_Name;

Object faculty\_Name;

Register\_Course(Object course\_Name, Object faculty\_Name)

{

this.course\_Name=course\_Name;

this.faculty\_Name=faculty\_Name;

}

public void run()

{

synchronized(course\_Name)

{

System.out.println("Register\_Course Thread holds course\_Name resource and waiting for faculty\_Name resource.....");

synchronized(faculty\_Name)

{

System.out.println("Register\_Course is success, because,

Register\_Course thread holds both course\_Name and faculty\_Name resources");

}

}

}

}

class Cancel\_Course extends Thread

{

Object course\_Name;

Object faculty\_Name;

Cancel\_Course(Object course\_Name, Object faculty\_Name)

{

this.course\_Name=course\_Name;

this.faculty\_Name=faculty\_Name;

}

public void run()

{

synchronized(faculty\_Name)

{

System.out.println("Cancel\_Course Thread holds faculty\_Name resource and waiting for course\_Name resource.....");

synchronized(course\_Name)

{

System.out.println("Cancel\_Course is success, because,

Cancel\_Course thread holds both faculty\_Name and course\_Name resources");

}

}

}

}

class Test

{

public static void main(String[] args)

{

Object course\_Name=new Object();

Object faculty\_Name=new Object();

Register\_Course registercourse=new Register\_Course(course\_Name, faculty\_Name);

Cancel\_Course cancelcourse=new Cancel\_Course(course\_Name, faculty\_Name);

registercourse.start();

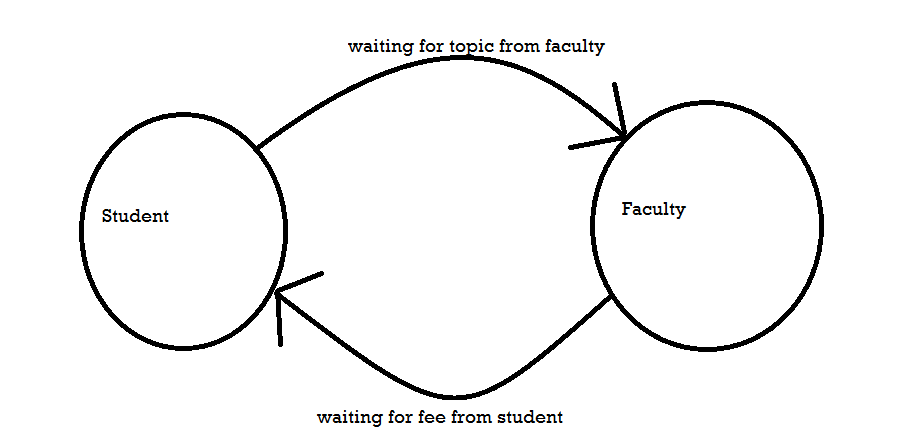
cancelcourse.start();

}

}

**Note:Both are depending on each other for their resources. So here, deadlock situation will occur. When deadlock situation is identified, automatically program is available in stucked mode.**

**Assignment:**



**Implement the above deadlock situation in the code.**