**Deadlock**

**Deadlock** occurs when two or more threads are blocked indefinitely, each waiting for a resource held by the other. This can cause the program to freeze.

* **Causes**:
  + Circular waiting: Thread A waits for Thread B, while Thread B waits for Thread A.
  + Holding locks in different orders by different threads.

**Avoiding Deadlocks**:

* **Lock Ordering**: Establish a fixed order in which locks are acquired.
* **Timeouts**: Use tryLock with timeouts to acquire locks.

If we change the consistent order of aqring lock we can over come this deadlock situation

**package** Locking;

**class** Pen {

**public** **synchronized** **void** writeWithPenAndPaper(Paper paper) {

System.***out***.println(Thread.*currentThread*().getName() + " is using pen " + **this** + " and trying to use paper " + paper);

paper.finishWriting();

}

**public** **synchronized** **void** finishWriting() {

System.***out***.println(Thread.*currentThread*().getName() + " finished using pen " + **this**);

}

}

**class** Paper {

**public** **synchronized** **void** writeWithPaperAndPen(Pen pen) {

System.***out***.println(Thread.*currentThread*().getName() + " is using paper " + **this** + " and trying to use pen " + pen);

pen.finishWriting();

}

**public** **synchronized** **void** finishWriting() {

System.***out***.println(Thread.*currentThread*().getName() + " finished using paper " + **this**);

}

}

**class** Task1 **implements** Runnable {

**private** Pen pen;

**private** Paper paper;

**public** Task1(Pen pen, Paper paper) {

**this**.pen = pen;

**this**.paper = paper;

}

@Override

**public** **void** run() {

pen.writeWithPenAndPaper(paper); // thread1 locks pen and tries to lock paper

}

}

**class** Task2 **implements** Runnable {

**private** Pen pen;

**private** Paper paper;

**public** Task2(Pen pen, Paper paper) {

**this**.pen = pen;

**this**.paper = paper;

}

@Override

**public** **void** run() {

**synchronized** (pen){ // if paper has the pen’s lock den it can aquire paper lock

paper.writeWithPaperAndPen(pen); // thread2 locks paper and tries to lock pen

}

}

}

**public** **class** DeadlockExample {

**public** **static** **void** main(String[] args) {

Pen pen = **new** Pen();

Paper paper = **new** Paper();

Thread thread1 = **new** Thread(**new** Task1(pen, paper), "Thread-1");

Thread thread2 = **new** Thread(**new** Task2(pen, paper), "Thread-2");

thread1.start();

thread2.start();

}

}