**Multithreading – Part-12 – Deadlock and Starvation**

* **DeadLock:**

If two threads are waiting for each other forever, such type of infinite waiting is called deadlock.

synchronized keyword is the only reason for deadlock situation, hence while using synchronized keyword we have to take special care.

There are no resolution techniques for deadlock but several prevention techniques are available.

* **Example:**

class A{

public synchronized void d1(B b){

System.out.println(“Thread 1 starts execution of d1() method”);

try{

Thread.sleep(5000);

} catch(InterruptedException ie){

}

System.out.println(“Thread 1 trying to call B’s last()”);

b.last();

}

public synchronized void last(){

System.out.println(“Inside A, this is last() method”);

}

}

class B{

public synchronized void d2(A a){

System.out.println(“Thread 2 starts execution of d2() method”);

try{

Thread.sleep(5000);

} catch(InterruptedException ie){}

System.out.println(“Thread 2 trying to call A’s last method”);

a.last();

}

public synchronized void last(){

System.out.println(“Inside B, this is last() method”);

}

}

class DeadLock extends Thread{

A a = new A();

B b = new B();

public void m1(){

this.start();

a.d1(b); // This line executed by main thread

}

public void run(){

b.d2(a); // This line executed by child thread

}

public static void main(String[] args){

DeadLock d = new DeadLock();

d.m1();

}

}

Output:

Thread1 starts execution of d1() method

Thread2 starts execution of d2() method

Thread2 trying to call A’s last()

Thread1 trying to call B’s last()

Then the program will stuck.

In the above program if we remove one synchronized keyword, then the program won’t entered into deadlock, hence synchronized keyword is the only reason for deadlock situation. Due to this while using synchronized keyword we have to take special care.

* **DeadLock VS Starvation:**

Long waiting of a thread where waiting never ends is called deadlock.

Whereas long waiting of a thread where waiting ends at certain point is called starvation.

For example, low priority thread has to wait until completing all high priority threads it may be long waiting but ends at certain point, which is nothing but starvation.

Let’s say there are 1 crore threads, among them only one thread is low priority and other threads are high priority, so this low priority thread has to wait until the high priority threads complete their work.