Assignment 19

**NAME- VENKATESHWARLU EGURLA (SHASHI)**

**Write a program to demonstrate synchronized block.**

**class** Call{

**public** **void** Callname(String name){

**synchronized**(**this**){

**for**(**int** i=0;i<=10;i++){

System.***out***.println("Hello : " +name);

**try**{

Thread.*sleep*(400);

}**catch**(Exception e){

System.***out***.println(e);

}

}

}

}

}

**class** MyThread1 **extends** Thread{

Call C;

MyThread1(Call C){

**this**.C=C;

}

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

C.Callname("raja");

}

}

**class** MyThread2 **extends** Thread{

Call C;

MyThread2(Call C){

**this**.C=C;

}

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

C.Callname("abhi");

}

}

**public** **class** TestSynchronization1 {

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

Call obj = **new** Call();

MyThread1 c1=**new** MyThread1(obj);

MyThread2 c2=**new** MyThread2(obj);

c1.start();

c2.start();

}

}

**Output**

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : raja

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

Hello : abhi

|  |  |
| --- | --- |
| **Method synchronized** | **Block synchronized** |
| If you declare any method as synchronized, it is known as synchronized method. | Synchronized block can be used to perform synchronization on any specific resource of the method. |
| Synchronized method is used to lock an object for any shared resource. | Suppose you have 50 lines of code in your method, but you want to synchronize only 5 lines, you can use synchronized block. |
| When a thread invokes a synchronized method, it automatically acquires the lock for that object and releases it when the thread completes its task. | If you put all the codes of the method in the synchronized block, it will work same as the synchronized method. |