**Multithreading – Part-09 – Synchronization Block**

* **Introduction:**

If very few lines of the code require synchronization then it’s not recommended entire method as synchronized, we have to enclose those few lines of the code by using synchronized block.

The main advantage of synchronized block over synchronized method is, it reduces waiting time of threads and improves performance of the system/application.

Note: Recall about the Hydrabad to Vijayawada travel with narrow bridge in between.

* **Declaring Synchronization Block:**

We can declare synchronized block as follows

* + To get lock of current object:

synchronized(this){

//code

}

If a thread got lock of current object then only it is allowed to execute this area.

* + To get lock of particular object “ b”:

synchronized(b){

//code

}

If a thread got lock of particular object b, then only it is allowed to execute this area.

* + To get class level lock:

synchronized(Display.class){

// code

}

If a thread got class level of “Display” class, then only it is allowed to execute this area.

* **Example:**

class Display{

public void wish(String name){

// 1lakh lines of code

synchronized(this){

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

System.out.print(“Good morning:”);

}

try{

Thread.sleep(2000);

} catch(InterruptedException ie){

}

System.out.println(name);

}

}

}

class MyThread extends Thread{

Display d;

String name;

MyThread(Display d, String name){

this.d = d;

this.name = name;

}

public void run(){

d.wish(name);

}

}

class SynchronizedDemo{

public static void main(String[] args){

Display d = new Display();

MyThread t1 = new MyThread(d, “Dhoni”);

MyThread t2 = new MyThread(d, “Yuvraj”);

t1.start();

t2.start();

}

}

* **Conclusion:**

Lock concept applicable for object types and class types, but not for primitives hence we can’t pass primitive type as argument to synchronized block otherwise we will get compile time error saying

int x = 10;

synchronized(x){

}

Unexpected Type:

found: int

required: reference

* **Frequently Asked Questions:**

1. What is synchronized keyword, where we can apply?
2. Explain advantage of synchronized keyword.
3. Explain disadvantage of synchronized keyword.
4. What is Race condition?

If multiple threads are operating simultaneously on same Java object then there may be a chance of data inconsistency problem, this is called “Race Condition”.

We can overcome this problem by using “synchronized” keyword.

1. What is object lock and when it is required?
2. What is class level lock and when it is required?
3. What is the difference between class level and object level lock?
4. While a thread executing synchronized method on the given object, is the remaining threads are allowed to execute any other synchronized method simultaneously on the same object?

NO

1. What is synchronized block?
2. How to declare synchronized block to get lock of current object?
3. How to declare synchronized block to get class level lock?
4. What is the advantage of synchronized block over synchronized method?
5. Is a thread can acquire multiple locks simultaneously?

Yes, of course from different objects.

Example:

class X{

public synchronized void m1(){

// Here thread has lock “x” object

Y y = new Y();

synchronized(y){

//here the thread has lock of x and y

Z z = new Z();

synchronized(z){

// Here thread has locks of x, y and z.

}

}

}

}

1. What is synchronized statement (terminology created by interview people)?

The statements present in synchronized method and synchronized block are called synchronized statements.