Synchronization in Threads

Synchronization in Java is a mechanism that controls access to **shared resources** by multiple threads, ensuring that only one thread accesses the resource at a time

Key Benefits of Synchronization:

Benefit	Description
Data Consistency	Ensures shared data remains accurate and consistent.
Prevents Race Conditions	Stops multiple threads from updating shared data simultaneously.
Thread Safety	Protects critical sections so program works correctly with multiple threads.

Main Types of Synchronization

Type	Description	Example
1. Synchronized Method	The entire method is synchronized, meaning only one thread can execute that method on a given object at a time.	synchronized void display() { ... }
2. Synchronized Block	Only a specific block of code is synchronized (instead of the whole method). It gives better performance.	synchronized(this) { ... }

Without Synchronization

```
class Callme
{
    static void call(String msg){
        System.out.print("[" + msg);
        try{
            Thread.sleep(1000);
        }
        catch(InterruptedException e){
            System.out.println("Interrupted");
        }
        System.out.println("]");
    }
}
class Caller extends Thread
{
    String msg;
    public Caller(String s){
        msg = s;
    }
    public void run(){
        Callme.call(msg);//call is from Thread class
    }
}
class threaddemo
{
    public static void main(String[] args){
        Caller ob1 = new Caller("Learn");
        Caller ob2 = new Caller("Java");
        Caller ob3 = new Caller("Programming");
        ob1.start();    //to call run method
        ob2.start();
        ob3.start();
    }
}
```

With Synchronization method

```
class Callme
{
    synchronized static void call(String msg){
        System.out.print("[" + msg);
        try{
            Thread.sleep(1000);
        }
        catch(InterruptedException e){
            System.out.println("Interrupted");
        }
        System.out.println("]");
    }
}
class Caller extends Thread
{
    String msg;
    public Caller(String s){
        msg = s;
    }
    public void run(){
        Callme.call(msg);//call is from Thread class
    }
}
class threaddemo
{
    public static void main(String[] args){
        Caller ob1 = new Caller("Learn");
        Caller ob2 = new Caller("Java");
        Caller ob3 = new Caller("Programming");
        ob1.start();    //to call run method
        ob2.start();
        ob3.start();
    }
}
```

Output

```
[Programming[Learn[Java]
]
]
```

```
[Learn]
[Programming]
[Java]
```