

Aim: To achieve synchronisation among multiple threads.

Theory:

Synchronisation - It is used to control the access of multiple threads to shared resources, preventing data inconsistencies and conflicts that might occur when multiple threads access the same resource simultaneously.

Syntax:

```
public synchronised void synchronised_Method()  
{  
    // Synchronised code block  
}
```

Program 1:

```
class Counter  
{  
    int count;  
    public synchronised void increment()  
    {  
        count++;  
    }  
}
```

}

public class Sync

76

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

```
    Counter c = new Counter();
```

```
    Thread t1 = new Thread(new Runnable()
{
```

```
    public void run()
{
```

```
    {
```

```
        for(int i=0; i<1000; i++)
```

```
        {
```

```
            c.increment();
```

```
        }
```

```
    }
```

```
});
```

```
    Thread t2 = new Thread(new Runnable()
{
```

```
    {
```

```
        public void run()
{
```

```
        {
```

```
            for(int i=0; i<1000; i++)
```

```
            {
```

```
                c.increment();
```

```
            }
```

```
        }
```

```
    });
```

```

t1.start();
t2.start();
t1.join(); // join() waits for completion of
t2.join();   thread.
System.out.println("count "+c.count);
}
}

```

steps to display output:

1. start
2. Open notepad
3. write code
4. compile code in command prompt.
5. Run code in command prompt.
6. Check output.
7. end.

conclusion: hence, we have achieved synchronisation among multiple threads.

9/10/24