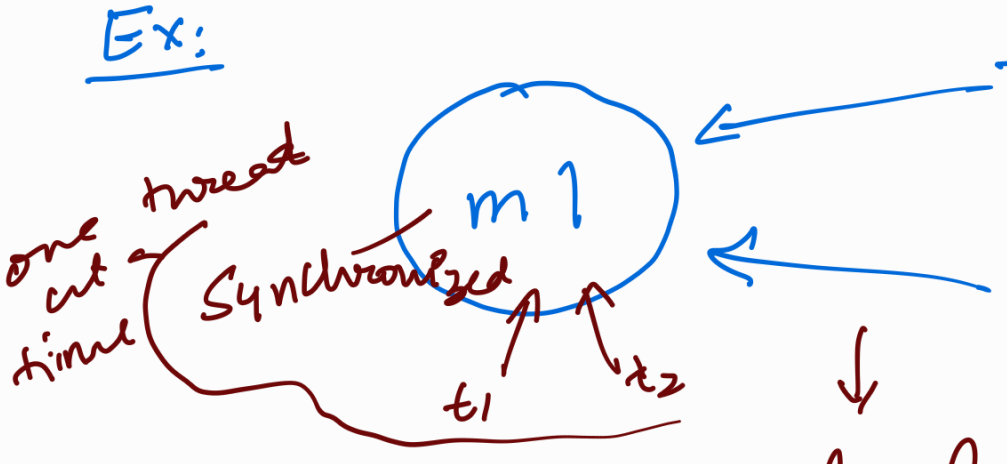
Synchronization of Thread in Java

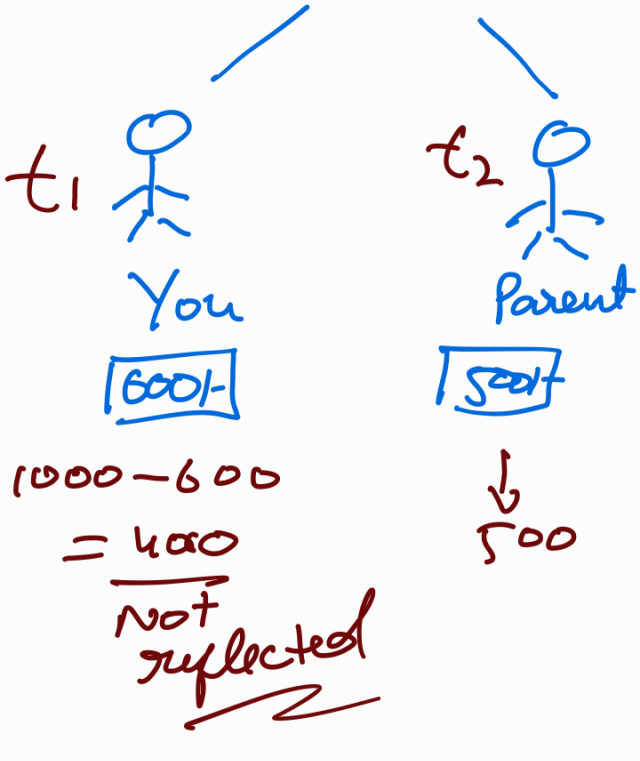
used to solve data inconsistency.

Ex:



while two threads of same object try to access a method, the data is accessed twiced causing data inconsistency.

Ex: consider → Your Account 10001- object



Note

→ Synchronization can be applied only for methods and block (not to class and variables)

→ Synchronized keyword in Java creates a block of code known as **critical section**.

⇓

to enter into critical section, a thread needs to obtain corresponding object's **lock**.

Syntax

```
Synchronized(object)
{
    // statements.
}
```

```
Class A
{
    void addnew (int i)
    {
    Thread = Thread.currentThread()
    for (int n=1 ; n<=5; n++)
      {
       S.O.P (t.getName()+"—"+
                          (i+n));
      }
}
```

```
Class B extends Thread
{
   A a1 = new A()
   public void run()
   {
      a1.addnew(100);
   }
```

```
    Class mainclass
    {
        P.S.V.M (S a[])
        {
            B b = new B();
            Thread t1 = new Thread (b);
            Thread t2 = new Thread (b);
              t1.setName ("T1");
              t2.setName ("T2");
              t1.start();
              t2.start();
        }
    }
```

o/p

```
T1 = 101
T2 = 101
T2 = 102
T1 = 102
  :
```

```
Class A
{
Synchronized void addnew (int i)
{
Thread t = Thread.currentThread();
for (int n=1 ; n<=5; n++)
{
S.O.P (t.getName()+"—"+
(i+n));
}
}
```

```
class B extends Thread
{
A a1 = new A()
public void run()
{ a1.addnew(100);
}
```

```
class mainclass
{
P.S.V.M (S a[])
{
B b = new B();
Thread t1 = new Thread (b);
Thread t2 = new Thread (b);
t1.setName ("T1");
t2.setName ("T2");
t1.start();
t2.start();
}
}
```

on use of Synchronized keyword.

↓

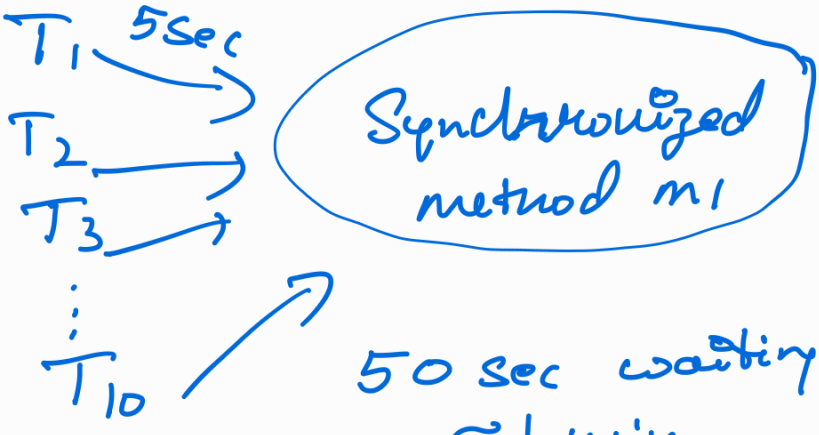
one Thread at a time + object

o/p.

```
T1 = 101
T1 = 102
T2 = 101
T2 = 102
  ⋮
```

Problem of Synchronization

- Increases the waiting time



- as only thread at a time, it will decrease the slm performance.

* Thus Synchronization is not advisable.

To avoid entire method to be synchronized, we can have

Synchronized Block

↓

Few lines of code to be synchronized.

Syntax :

```
Synchronized (object)
{
    ==
}
```

Programs

VR → Synch.java
Synch1.java
Synchronized Block.java (*)

Inter Thread Com"

VR → Deadlock.java
Suspend Resume.java

wait() notify() notifyAll() → Ex ?