class X {

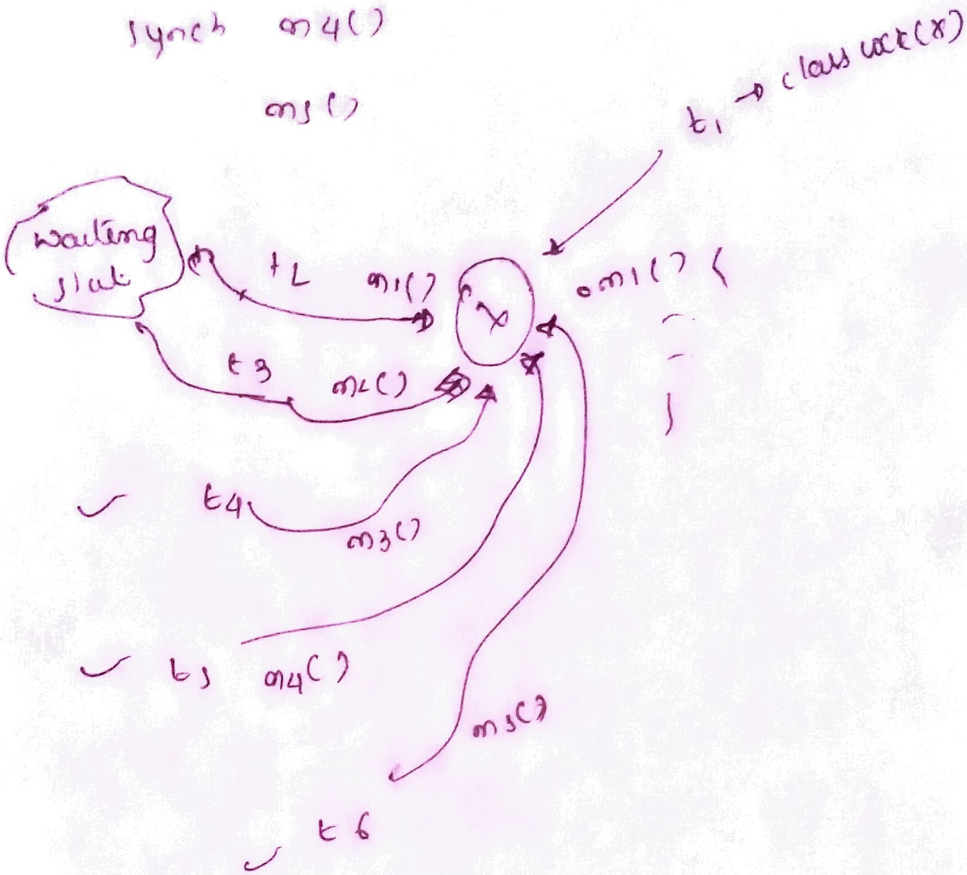
static synch m1()

static synch m2()

static m3()

synch m4()

m5()



class Display {

public void display() {

for(int i=1; i<=10; i++){

System.out.println(i);

try{

Thread.sleep(2000);

}

catch(InterruptedException e){ }

}

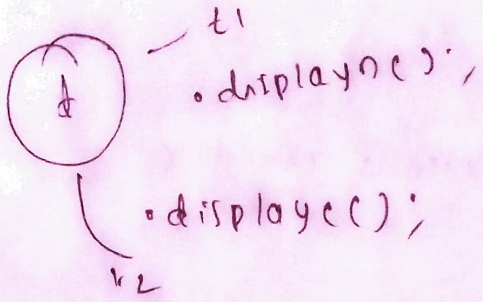
}

```
public void displayC() {
    for(int i 65; i< 78; i++){
    System.out.(char)i);
      try{
        Thread.sleep(2000);
      }
      catch(InterruptedException e){ }
    }
  }

class MyThread1 extends Thread {
    Display d;
    MyThread1(Display d) {
      this.d = d;
    }
    public void run() {
      d.displayn();
    }
}

class MyThread2 extends Thread {
    Display d;
    MyThread2(Display d) {
      this.d = d;
    }
    public void run() {
      d.displayc();
    }
}
```

```
class SynchronizedDemo {
psvm(String[] args){
Display d = new Display();
Mythread1 t1 = new Mythread1(d);
Mythread2 t2 = new Mythread2(d);
t1.start();
t2.start();
```



Synchronized block

```
synchronized(this){
    =
}
```

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

① To get a lock of current object

② to get lock of particular object b:

```
synchronized (b) {

}
```

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

③ To get class level lock:

```
synchronized ( Display.class ) {

}
```

if a thread got class level lock of Dilplay ~~thread~~ class, then only it is allowed to execute this area

```
class Display {
    public void wish(String name) {
        ;;; take lines of code
        for(int i=0; i<10; i++){
            System.out.println("Good Morning");
            try{
                Thread.sleep(2000);
            }
            catch(InterruptedException e){ }
            System.out.println(name);
        }
        ;;;; take lines of code
    }
}

class MyThread extends Thread {
    Display d;
    String name;
    MyThread(Display d, String name){
        this.d=d;
        this.name=name;
    }
    public void run(){
        for(int i=0;;
        d.wish(name);
    }
}
```

```
class SynchronizedDemo {
    psvm (String[] args) {
        Display d = new Display();
        MyThread t1 = new MyThread(d, "dhoni");
        MyThread t2 = new MyThread(d, "yuvaraj");
        t1.start();
        t2.start();
```

→ if provide regular output

If we created 2 objects

```
Display d1 = new Display();
Display d2 = new Display();
MyThread t1 = new MyThread(d1, "dhoni");
MyThread t2 = new MyThread(d2, "yuvraj");
t1.start();
t2.start();
```

→ irregular output

CDAC Thread Program

In Mythread class place for loop in synchronized(Display.class){

```
Synchronized (Display.class) {
    for(int i=0; i<10; i++){
    sop("Good Morning");
    try{
      Thread.sleep(2000);
    }
    catch(InterruptedException e){ }
     sop(name);
    }
}
```

```
Display d1 = new Display();
Display d2 = new Display();
Mythread t1 = new Mythread(d1, "dhoni");
Mythread t2 = new Mythread(d2, "yuvraj");
t1.start();
t2.start();
```

output will be regular in nature

lock is available for

- object level
- class level

lock is not available for primitives

```
int x = 10;
synchronized (x) {
    ...
}
```

throws compile error

unexpected type found: int
required: reference

lock concept applicable for

- object types
- class types

but not for primitives hence we can't pass primitive type as argument to synchronized block.

otherwise we will get compile type error saying:

unexpected type
found: int
required: reference

FAQs

① What is synchronized keyword where we can apply. ?

- modifier applicable for methods, blocks, but not for variables and classes.

② Explain advantage of synchronized keyword ?

we can resolve data inconsistency problems

③ Explain disadvantage of synchronized keyword ?

It increase waiting time of threads and creates performance problems

④ What is race condition ?

If multiple threads are operating simultaneously on the same java object, there may arise data inconsistency problem that is called as race condition

If multiple threads are operating simultaneously on same java object then there may be chance of data inconsistency problem that is called race condition

we can over come this problem by using synchronized keyword

(5) what is object lock? and when it is required?

Every object in java has a unique lock which is nothing but object level lock

when ever a thread wants to execute instance synchronized method, then the thread require object level lock.

(6) what is class level lock? when it is required?

Every class in java has a unique lock which is nothing but class level lock

when ever if a thread wants to execute static synchronized method then the thread require this level lock

⑦ class level lock V/S object level lock

↓

thread wants to execute static synchronized method then class level lock needed

thread wants to execute instance synchronized method then object level lock needed

⑧ while a thread executing synchronized method on the given object, all the remaining threads are allowed to execute any other synchronized method simultaneously on the same object

- NO

⑨ what is synchronized block

(10) How to declare synchronized block ?

- to get lock of current object

```
synchronized (this) {

}
```

(11) How to declare synchronized block

- to get class level lock

```
synchronized (classname.class) {

}
```

(12) what is the advantage of synchronized block over synchronized method ?

- performance by default will be improved
- waiting time of thread will be reduced

(13) If a thread can acquire multiple locks simultaneously ?

- Yes • from different objects

```
class X {

    public synchronized void m1(){
        // here thread has lock of X object
        Y y = new Y();

        synchronized(y) {
            // here thread has locks of X & Y
            Z z = new Z();

            synchronized(z) {
                // here thread has locks of X, Y, Z
            }
        }
    }
```

```
X x = new X();
x.m1();
```

A thread can aquire multiple locks simultaneously from different objects

(14) what is synchronized statement?

the statement present in synchronized methods and statement present in synchronized block. are called synchronized statements