

PRACTICE PROGRAMS ON INTER-THREAD COMMUNICATION (SYNCHRONIZATION)

1. Create a class Table including a method printable which prints multiplication table of a given value . Create two threads which prints multiplication table of 5 and 100 by calling the same function on same object. Test the threads

```
class Table
{
    void printTable(int n)
    {
        synchronized(this)
        {
            for(int i=1;i<=10;i++)
            {
                System.out.println(+n+"*" +i+"="+ (n*i));
                try
                {
                    Thread.sleep(400);
                }
                catch(Exception e)
                {
                    System.out.println(e);
                }
            }
        }
    }
}
```

```
class Mythread1 extends Thread
```

```
{  
    Table t;  
    Mythread1(Table t)  
    {  
        this.t=t;  
    }  
    public void run()  
    {  
        t.printTable(5);  
    }  
}
```

```
class Mythread2 extends Thread
```

```
{  
    Table t;  
    Mythread2(Table t)  
    {  
        this.t=t;  
    }  
    public void run()  
    {  
        t.printTable(100);  
    }  
}
```

```
class multiplication_table
```

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

```
{  
    Table obj = new Table();  
    Mythread1 thread1 = new Mythread1(obj);  
    Mythread2 thread2 = new Mythread2(obj);  
    thread1.start();  
    thread2.start();  
}  
}
```

Command Prompt

```
Microsoft Windows [Version 10.0.19041.685]  
(c) 2020 Microsoft Corporation. All rights reserved.  
  
C:\Users\SAKSHI>cd C:\Users\SAKSHI\JAVA PROGRAMS  
  
C:\Users\SAKSHI\JAVA PROGRAMS>javac multiplication_table.java  
  
C:\Users\SAKSHI\JAVA PROGRAMS>java multiplication_table  
5*1=5  
5*2=10  
5*3=15  
5*4=20  
5*5=25  
5*6=30  
5*7=35  
5*8=40  
5*9=45  
5*10=50  
100*1=100  
100*2=200  
100*3=300  
100*4=400  
100*5=500  
100*6=600  
100*7=700  
100*8=800  
100*9=900  
100*10=1000  
  
C:\Users\SAKSHI\JAVA PROGRAMS>
```

2.Create two class Car_Owner and Car_Mechanic . Make them threads. The two threads tries to access an object of Car_queue class. Car_queue involves get and put function based on the availability of mechanic. Car_owner should put the request for car service and Car mechanic should get the request posed by car owner one at a time.

```
class Car_Service
```

```
{  
    public static void main(String args[])  
    {  
        Car_Queue q= new Car_Queue();  
        new Car_Owner(q);  
        new Car_Mechanic(q);  
        System.out.println("Press CTRL+C to stop.\n");  
    }  
}
```

```
class Car_Queue
```

```
{  
    int n;  
    boolean valueSet = false;  
    synchronized int get()  
    {  
        while(!valueSet)  
        try  
        {  
            wait();  
        }  
        catch(InterruptedException e)  
        {  
            System.out.println("InterruptedException caught");  
        }  
    }  
}
```

```

    }
    System.out.println("Service Provided:"+n);
    valueSet = false;
    notify();
    return n;
}
synchronized void put(int n)
{
    while(valueSet)
    try
    {
        wait();
    }
    catch(InterruptedException e)
    {
        System.out.println("InterruptedException caught");
    }
    this.n=n;
    valueSet = true;
    System.out.println("Order placed:"+n);
    notify();
}
}

```

```

class Car_Mechanic implements Runnable
{
    Car_Queue q;
    Car_Mechanic(Car_Queue q)

```

```

    {
        this.q=q;
        new Thread(this,"Car_Mechanic").start();
    }
    public void run()
    {
        int i=0;
        while(true)
        {
            q.put(i++);
        }
    }
}

```

class Car_Owner implements Runnable

```

{
    Car_Queue q;
    Car_Owner(Car_Queue q)
    {
        this.q=q;
        new Thread(this,"Car_Owner").start();
    }
    public void run()
    {
        while(true)
        {
            q.get();
        }
    }
}

```

```
}  
}
```

Command Prompt

```
C:\Users\SAKSHI>cd C:\Users\SAKSHI\JAVA PROGRAMS  
  
C:\Users\SAKSHI\JAVA PROGRAMS>javac Car_Service.java  
  
C:\Users\SAKSHI\JAVA PROGRAMS>java Car_Service  
Press CTRL+C to stop.  
  
Order placed:0  
Service Provided:0  
Order placed:1  
Service Provided:1  
Order placed:2  
Service Provided:2  
Order placed:3  
Service Provided:3  
Order placed:4  
Service Provided:4  
Order placed:5  
Service Provided:5  
Order placed:6  
Service Provided:6  
Order placed:7  
Service Provided:7  
Order placed:8  
Service Provided:8  
Order placed:9  
Service Provided:9  
Order placed:10  
Service Provided:10  
Order placed:11  
Service Provided:11  
Order placed:12  
Service Provided:12  
Order placed:13  
Service Provided:13  
Order placed:14  
Service Provided:14  
Order placed:15  
Service Provided:15  
Order placed:16  
Service Provided:16  
Order placed:17  
Service Provided:17  
Order placed:18  
Service Provided:18  
Order placed:19  
Service Provided:19  
Order placed:20  
Service Provided:20  
Order placed:21
```

Command Prompt

```
Service Provided:20
Order placed:21
Service Provided:21
Order placed:22
Service Provided:22
Order placed:23
Service Provided:23
Order placed:24
Service Provided:24
Order placed:25
Service Provided:25
Order placed:26
Service Provided:26
Order placed:27
Service Provided:27
Order placed:28
Service Provided:28
Order placed:29
Service Provided:29
Order placed:30
Service Provided:30
Order placed:31
Service Provided:31
Order placed:32
Service Provided:32
Order placed:33
Service Provided:33
Order placed:34
Service Provided:34
Order placed:35
Service Provided:35
Order placed:36
Service Provided:36
Order placed:37
Service Provided:37
Order placed:38
Service Provided:38
Order placed:39
Service Provided:39
Order placed:40
Service Provided:40
Order placed:41
Service Provided:41
Order placed:42
Service Provided:42
Order placed:43
Service Provided:43
Order placed:44
```

```
C:\Users\SAKSHI\JAVA PROGRAMS>
```