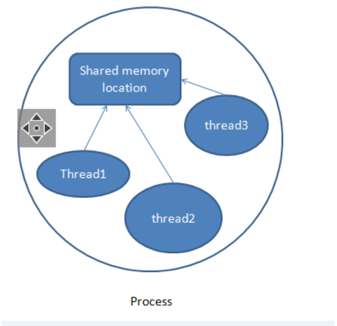
Java Interview Q&A on Java Multi-Threading

- ***By SivaReddy***

1) What is Thread and Process?
   - A process has a self-contained execution environment
   - Threads are sometimes called lightweight processes

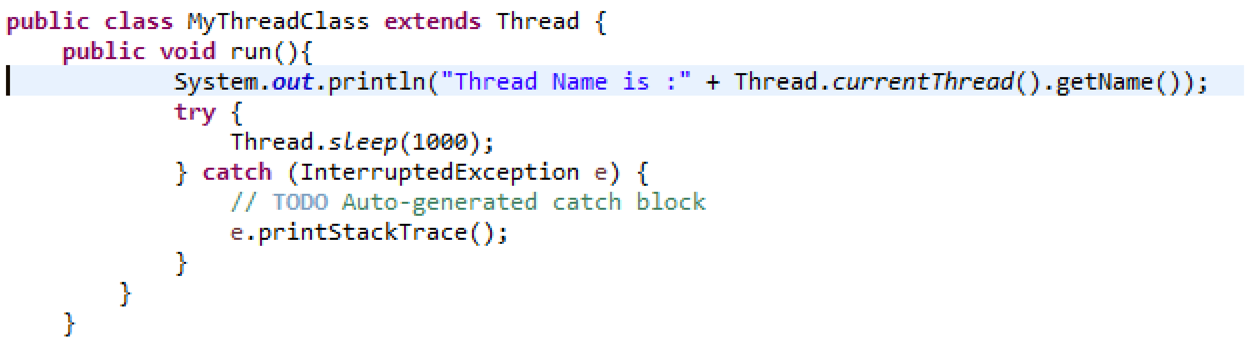


2) What is Multi-Threading and multi-process?
   - Multiprocessing - the simultaneous execution of two or more programs or instruction sequences by separate CPUs under integrated control
   - Multithreading in java is a process of executing multiple threads simultaneously.

3) How to create threads in java?

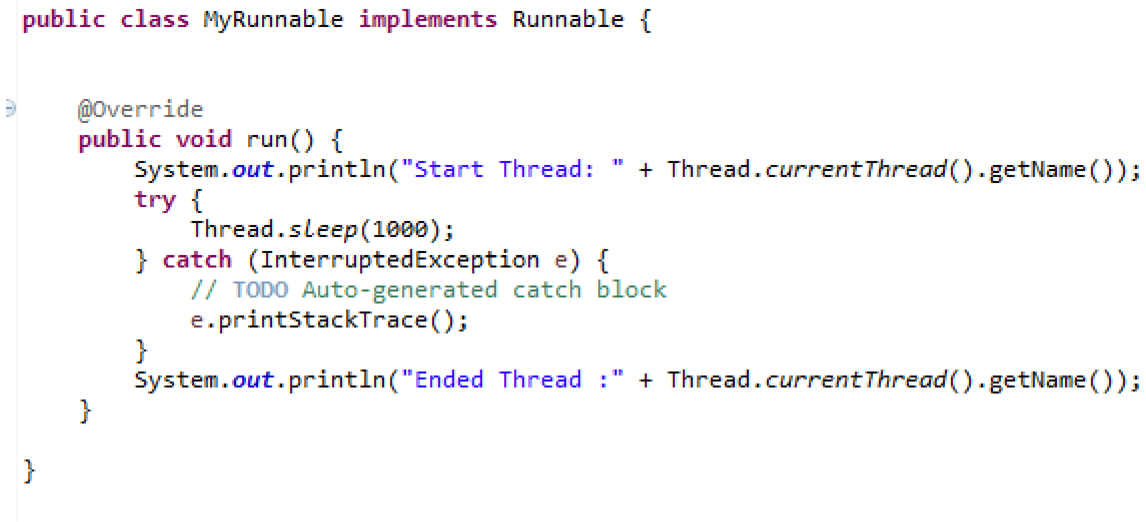
   You can create threads in java two ways

   - By extending Thread Class



```
public class MyThreadClass extends Thread {
    public void run(){
            System.out.println("Thread Name is :" + Thread.currentThread().getName());
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }
```

- By implementing Runnable interface



```
public class MyRunnable implements Runnable {


    @Override
    public void run() {
        System.out.println("Start Thread: " + Thread.currentThread().getName());
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        System.out.println("Ended Thread :" + Thread.currentThread().getName());
    }

}
```

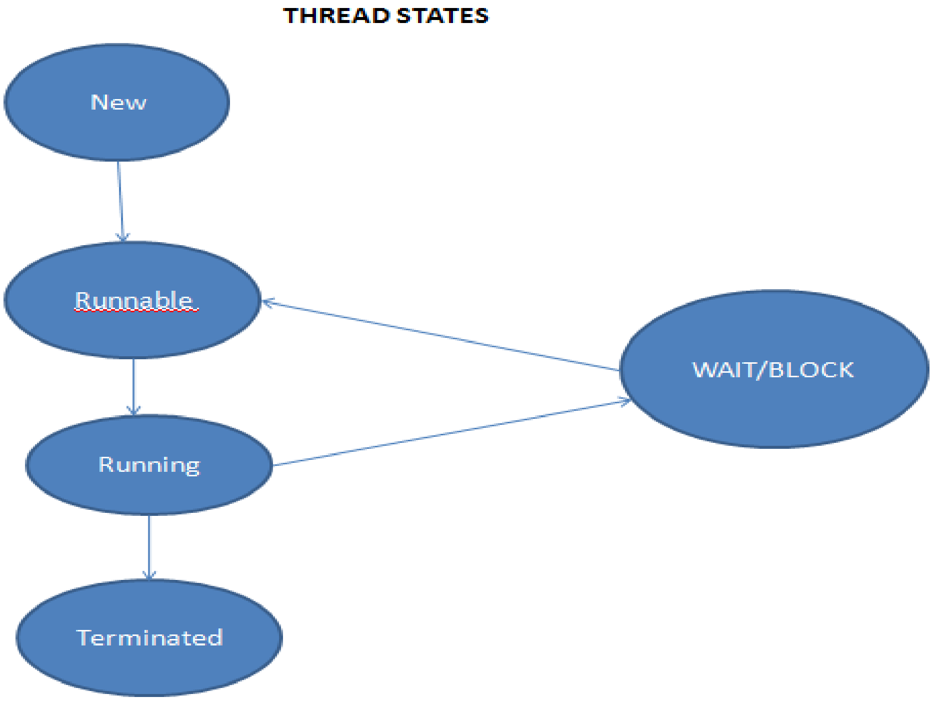
4) When to use Runnable and when to use Thread class, which is better?

Runnable is preferable when:

- You want to use only basic thread class functionality
- If your subclass class already extends some other class

Thread Class: If you want to use complete capabilities of Thread class

5) What is the thread Life cycle?



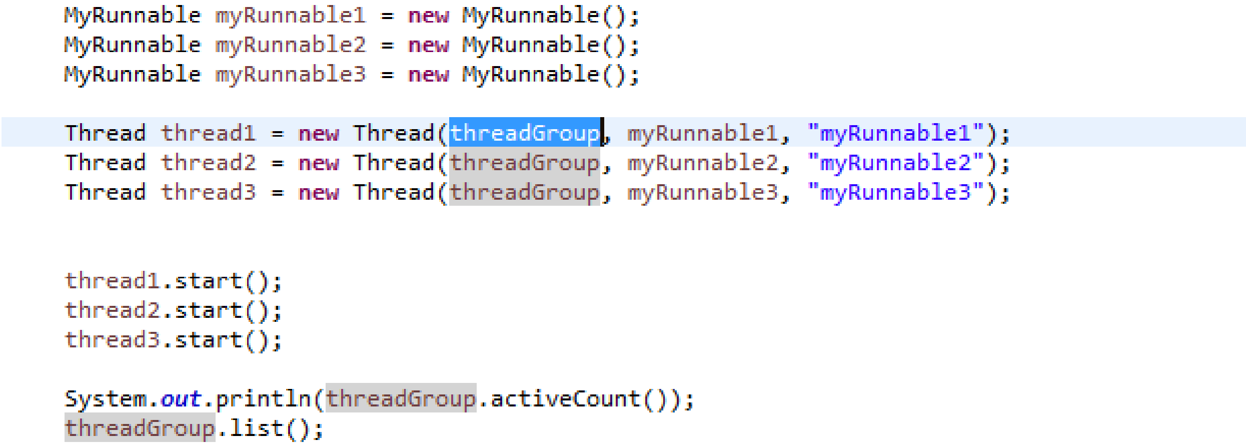
6) What is use of sleep, join, yield, wait, notify, setPriority, setDemon methods?

- Sleep method used to sleep the current running the specified time
- Join method used to currently running thread to die

- Wait method is to wait the currently thread until notify the current thread
- Notify is used to wake up the currently wait thread to runnable state
- setPriority used to set the priority for the thread
- setDemon method is used to make current thread as demon

7) What is meant by ThreadSchedular and its advantage?
8) Can we start the thread already started thread again?
9) What are ThreadGroup and Thread Pool?

Thread Group:

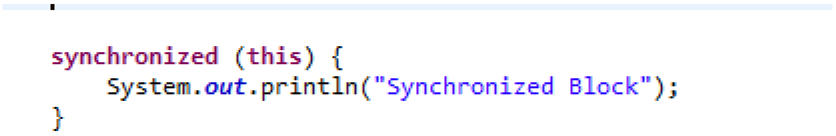


```
MyRunnable myRunnable1 = new MyRunnable();
MyRunnable myRunnable2 = new MyRunnable();
MyRunnable myRunnable3 = new MyRunnable();

Thread thread1 = new Thread(threadGroup, myRunnable1, "myRunnable1");
Thread thread2 = new Thread(threadGroup, myRunnable2, "myRunnable2");
Thread thread3 = new Thread(threadGroup, myRunnable3, "myRunnable3");


thread1.start();
thread2.start();
thread3.start();

System.out.println(threadGroup.activeCount());
threadGroup.list();
```

-
- **Thread Pool**
- ExecutorService executor = Executors.newFixedThreadPool(10); //this will create 10 threads when you execute the program

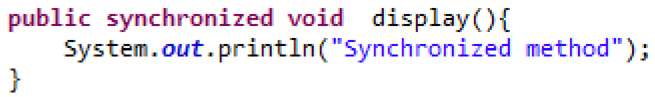
10) What is meant by demon thread?
11) Why wait, notify, notifyAll methods are part of the Object class not part of the Thread class?
12) Can we invoke run method instead of start method?
13) What is Interrupted Exception, IllegalThreadStateException?
14) What is meant by synchronization and advantage of it?
15) When to use synchronize block, method and static synchronization?

Synchronized Block:



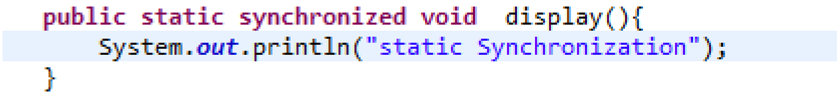
```
synchronized (this) {
    System.out.println("Synchronized Block");
}
```

Synchronized Method:



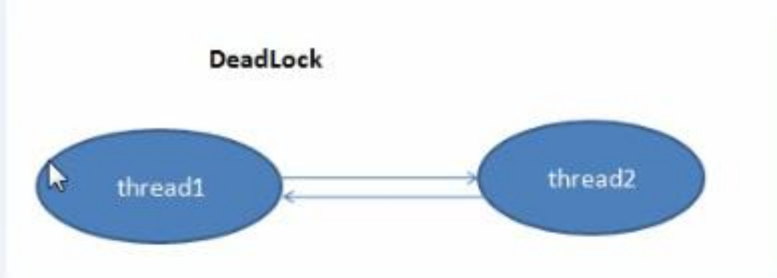
```
public synchronized void  display(){
    System.out.println("Synchronized method");
}
```

Static synchronization:



```
public static synchronized void  display(){
    System.out.println("static Synchronization");
}
```

16) What is deadlock?



17) What is the use of volatile keyword in java?
18) What are Atomic Classes and its advantages?
19) What is meant by Lock?
20) What is the different between Synchronization and Lock?