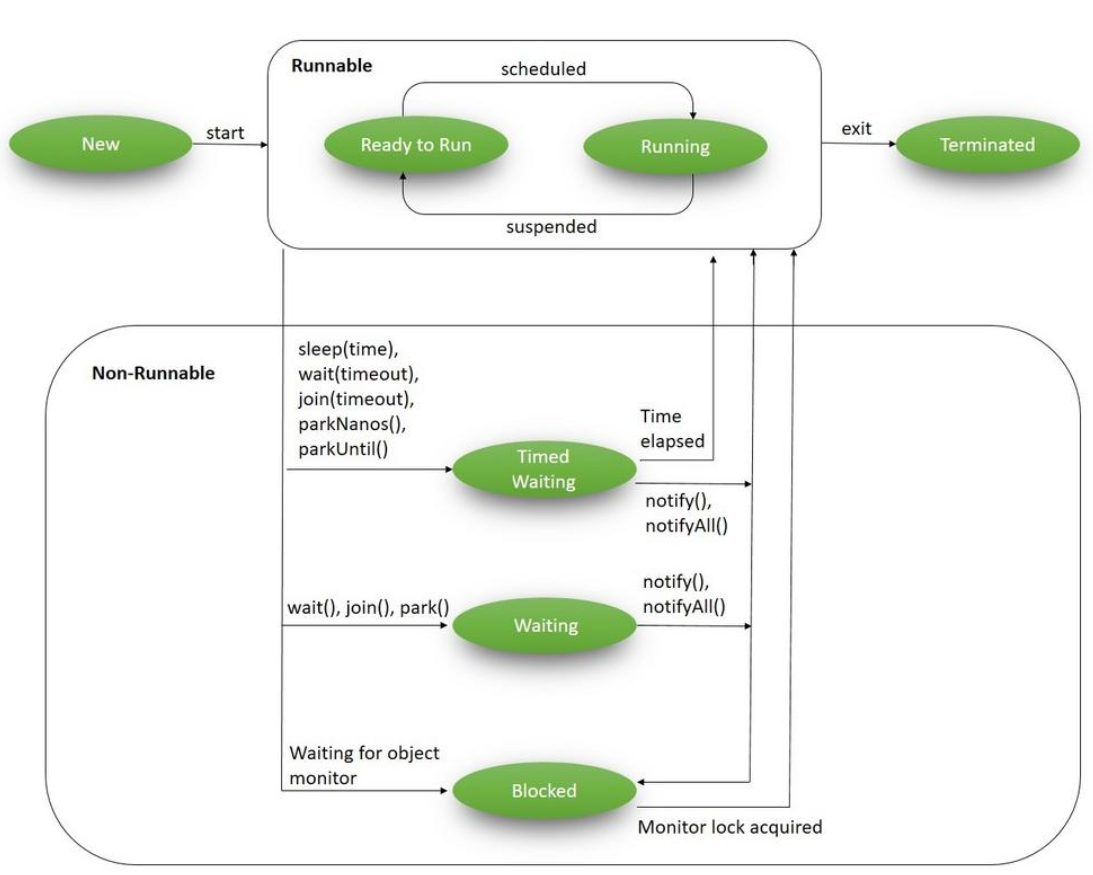
**Task 1: Mcq interview questions on Multi-threading**

**1.What is thread lifecycle and mention the different states of a thread’s lifecycle?** As we Know a thread is well known for independent execution , during lifecycle a thread can move from different state.

1. New
2. Active
3. Blocked / Waiting
4. Timed Waiting
5. Terminated

**2. What is the implementation of thread cycle?**

****

**3. What is the start() and run() method of Thread class?**

**start()**: In simple words, the start() method is used to start or begin the execution of a newly created thread. When the start() method is called, a new thread is created and this newly created thread executes the task that is kept in the run() method. One can call the start() method only once.    
  
**run()**: In simple words, the run() method is used to start or begin the execution of the same thread. When the run() method is called, no new thread is created as in the case of the start() method. This method is executed by the current thread. One can call the run() method multiple times.

Example: class starting implements Runnable{  
 @Override  
 public void run() {  
 // No processing in this block  
 }  
}  
  
public class NewState {  
 public static void main(String[] args) throws InterruptedException {  
 starting r=new starting();// child class  
  
 Thread t=new Thread(r);  
  
 System.*out*.println(t.getState());  
 }

### 4. What’s the purpose of the join() method?

**join()** method is generally used to pause the execution of a current thread unless and until the specified thread on which join is called is dead or completed. To stop a thread from running until another thread gets ended, this method can be used. It joins the start of a thread execution to the end of another thread’s execution. It is considered the final method of a thread class.

Ex: class D extends Thread{  
 public void run(){  
 String n=Thread.*currentThread*().getName();  
 try{  
 for (int i=0;i<=3;i++)  
 {  
 System.*out*.println(n);  
 Thread.*sleep*(1000);  
 }  
 }  
 catch (InterruptedException i){  
  
 }  
 }  
}  
  
public class joinExample {  
 public static void main(String[] args){  
 D t1=new D();  
 D t2=new D();  
 D t3=new D();  
  
 t1.setName("Thread 1");  
 t2.setName("Thread 2");  
 t3.setName("Thread 3");  
  
 t1.start();  
 try{  
 t1.join();// by this join execute the thread completely  
 }  
 catch (InterruptedException i){  
  
 }  
 t2.start();  
 String n=Thread.*currentThread*().getName();  
 for (int i=0;i<=3;i++)  
 {  
 System.*out*.println(n);  
 }  
 t3.start();  
  
 }  
}

**5. What is the difference between and sleeping and yield?**

Sleep( ): It is static method of thread class, which throws checked exception i.e Interrupted Exception .

The main purpose of sleep method to put a thread into temporary waiting state.

**Ex:** class C extends Thread{  
 public void run(){  
 String n=Thread.*currentThread*().getName();  
 try{  
 for (int i=0;i<=3;i++)  
 {  
 System.*out*.println(n);  
 Thread.*sleep*(1000);  
 }  
 }  
 catch (InterruptedException i){  
  
 }  
 }  
}  
  
public class sleepExample {  
 public static void main(String[] args){  
 C t1=new C();  
 C t2=new C();  
 C t3=new C();  
  
 t1.setName("Thread 1");  
 t2.setName("Thread 2");  
 t3.setName("Thread 3");  
  
 t1.start();  
 t2.start();  
 t3.start();  
 }  
}

**Yield ( ) :**

Yield method of thread class that allow us to run another thread which has same priority by pause it current thread**.**

**Example:**

class thread1 extends Thread{  
 public void run(){  
 String n1=Thread.*currentThread*().getName();  
 int n=n1.length();  
 for(int i=0;i<n;i++)  
 {  
 System.*out*.println(n1);  
 Thread.*yield*();// excute in the last  
 }  
 }  
}  
  
class thread2 extends Thread{  
 public void run(){  
 String n1=Thread.*currentThread*().getName();  
 int n=n1.length();  
 for(int i=0;i<n;i++)  
 {  
 System.*out*.println(n1);  
 }  
 }  
}  
public class yeilsExample {  
 public static void main(String[] arg){  
  
 thread1 t1=new thread1();  
 thread2 t2=new thread2();  
  
 t1.setName("Hello");  
 t2.setName("x2");  
  
 t1.start();  
 t2.start();  
 }  
}