**WTHAT IS THREAD**

A thread can be defined as a lightweight process which can execute multiple pieces codes in parallel.

**How to create thread?**

1.By Extending Thread class

2.Implementing Runnable Interface

**THREAD STATES**

**1.New Born State**

The programmer creates the thread type object.Once thread type object is created,it is stored inside heap memory and the object is created as inactive thread until the programmer will call the start() method.

Only start() method can be called on a new thread; otherwise, an **IllegalThreadStateException** will be thrown.

**2.Runnable**

When the programmer uses thread type object to call the start() method,the JVM will now the run() method and execute the functionality of the thread

Runnable state means a thread is ready for execution. When the start() method is called on a new thread, thread enters into a runnable state.  
In runnable state, thread is ready for execution and is waiting for availability of the processor (CPU time). That is, thread has joined queue (line) of threads that are waiting for execution.

If all threads have equal priority, CPU allocates time slots for thread execution on the basis of first-come, first-serve manner. The process of allocating time to threads is known as **time slicing**. A thread can come into runnable state from running, waiting, or new states.

**3.Running**

In running state, processor gives its time to the thread for execution and executes its run method. This is the state where thread performs its actual functions. A thread can come into running state only from runnable state.

a) When sleep() method is invoked on a thread to sleep for specified time period, the thread is out of queue during this time period. The thread again reenters into the runnable state as soon as this time period is elapsed.

b) When a thread is suspended using suspend() method for some time in order to satisfy some conditions. A suspended thread can be revived by using resume() method.

c) When wait() method is called on a thread to wait for some time. The thread in wait state can be run again using notify() or notifyAll() method.

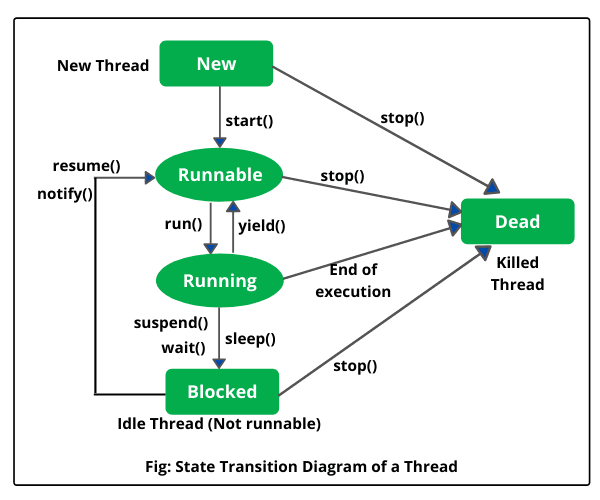
**4.Blocked(Non-runnable state)**

During execution of run() method,if programmer uses sleep() method then the state of thread changes from runnable state to blocked state and thread remains in blocked state for a specified duration of time provided in the sleep() method.Once specified duration is completed it goes back from blocked state to runnable state and continues the execution of run() method

**5.Dead**

Once the execution of run() method is completed,the JVM will call the stop() method to change the state from runnable state to dead state.

**THREAD LIFECYCLE**



**COMMONLY USED METHODS OF THREAD CLASS**

start()🡪It is used to start the lifecycle of a thread,when programmer calls run method,jvm internally calls run() and executes functionality of thread

run()🡪this method contains empty functionality.Hence programmer must override the run() method & provide the functionality of thread

stop()🡪JVM uses the stop() method once the execution of run() method is completed.

currentThread()🡪returns the memory address of the current thread object

sleep()🡪It is a static method present in the Thread class.It is used to make a current thread inactive for specified duration of time, after the specified duration the thread becomes active.It will generate an exception called Interrupted Exception.

setName() 🡪used by the programmer to provide the name for the current thread

getName()🡪used by the programmer to obtain the name(provided using setName() of the thread

isAlive()🡪Check if the thread is alive

getPriority()🡪It returns the priority of the thread

**WHAT IS SYNCHRONIZATION**

* synchronization is used to protect access to resources that are accessed concurrently.
* When multiple Threads access a common resource, the data of the common resource will undergo Data corruption.To overcome this problem,Thread synchronization is used.
* Thread synchronization can be achieved by using a keyword “Synchronized” on the method declaration.

SAMPLE PROGRAM FOR THREAD IMPLEMENTATION

//Using Runnable Interface

**public** **class** ThreadDemo **implements** Runnable

{

**public** **void** run()

{

**for**(**int** i=1;i<4;i++)

{

System.***out***.println("hey threa1 started");

}

}

**public** **static** **void** main(String[] args)

{

ThreadDemo td1 = **new** ThreadDemo();

Thread t1= **new** Thread(td1);

t1.start();

ThreadDemo td2 = **new** ThreadDemo();

Thread t2 = **new** Thread(td2);

t2.start();

}

}

//Using Thread Class

**public** **class** ThreadExample **extends** Thread {

**public** **void** run()

{

**for**(**int** i=0;i<10;i++)

{

System.***out***.println("Keep Praising");

}

}

**public** **static** **void** main(String[] args)

{

ThreadExample te = **new** ThreadExample();

te.setName("Java Thread");

te.start();

String name = te.getName();

System.***out***.println(name);

**boolean** state = te.isAlive();

System.***out***.println(state);

}

}