Threads

Multi-tasking:

Executing serval tasks simultaneously at a time is called multi-tasking.

Multi-tasking has been divided into two processes

1. Process based

Executing serval tasks simultaneously at a time, but that tasks should be separate independent process called process based multi-tasking.

Best suitable for OS level not programmatic level.

1. Thread based.

Executing server task simultaneously at a time, where each task is separate independent part of same program such type of process is called thread-based process.

Advantages

Either thread based or process based, main agenda is, instead of completing single task one by one, by using multi -tasking we can complete multiple tasks simultaneously to reduce response time of system and to improve the performance with in less time.

Where we can use multi-threading?

In web servers, like G-mail

Many request will come to Gmail server, if request are execute one by one means we need to wait for entire life, so here implemented multi thread concept and request will execute simultaneously.

Thread?

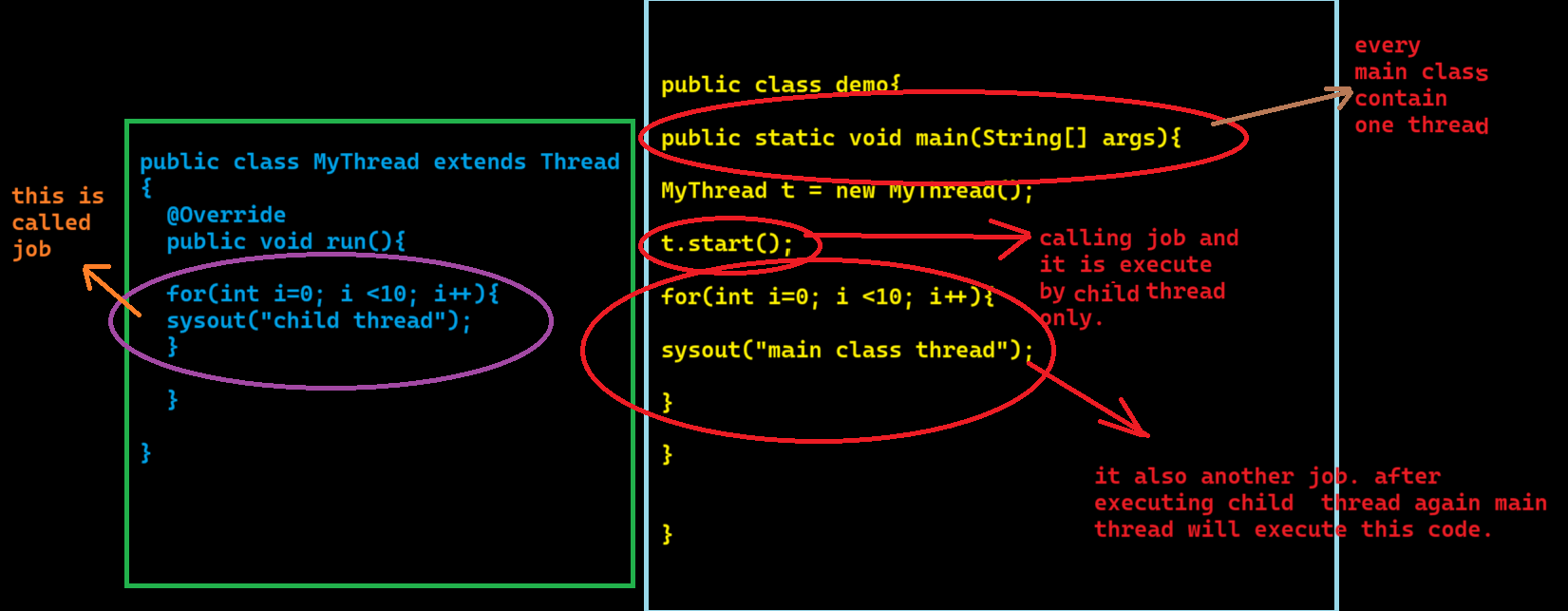
A flow of execution is nothing but a thread. For every execution there will be a separate job. If we have single flow then it is single thread, if we have multiple flows then it is multi thread. Every main method is a one thread in java.

Thread can be created two ways.

1. By extending thread class.

We need to override the run() method.

1. By implementing Runnable interface.



Threads schedule:

If multiple threads are getting ready to execute, then thread scheduler will execute simultaneously. We can’t expect thread execution order. It is varied from JVM to JVM .

Note: we can’t except particular output in this case.

Difference between start () and run ()?

If I call start () method, one more thread will be created and that thread will be responsible to run the code inside the run () method.

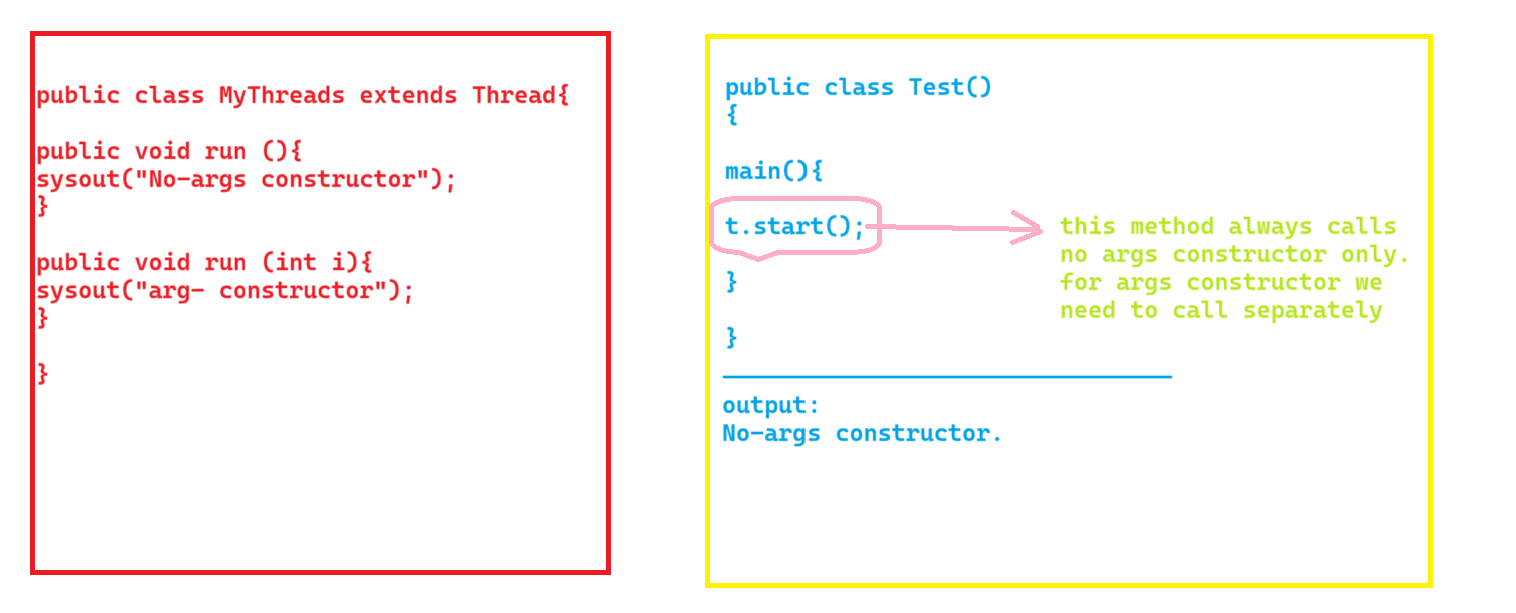
If in call run () method, I will execute as a normal method using main thread (psvm method ()) without creating thread.

Important of start () method?

1. By calling start method, we can register thread with thread scheduler.
2. Perform all other mandatory activities. With out executing start () method there is no chance of starting new thread in java. Due to this start () method is consider as heart of multi- threading.
3. Invoke run () method.

Overloading of run ()

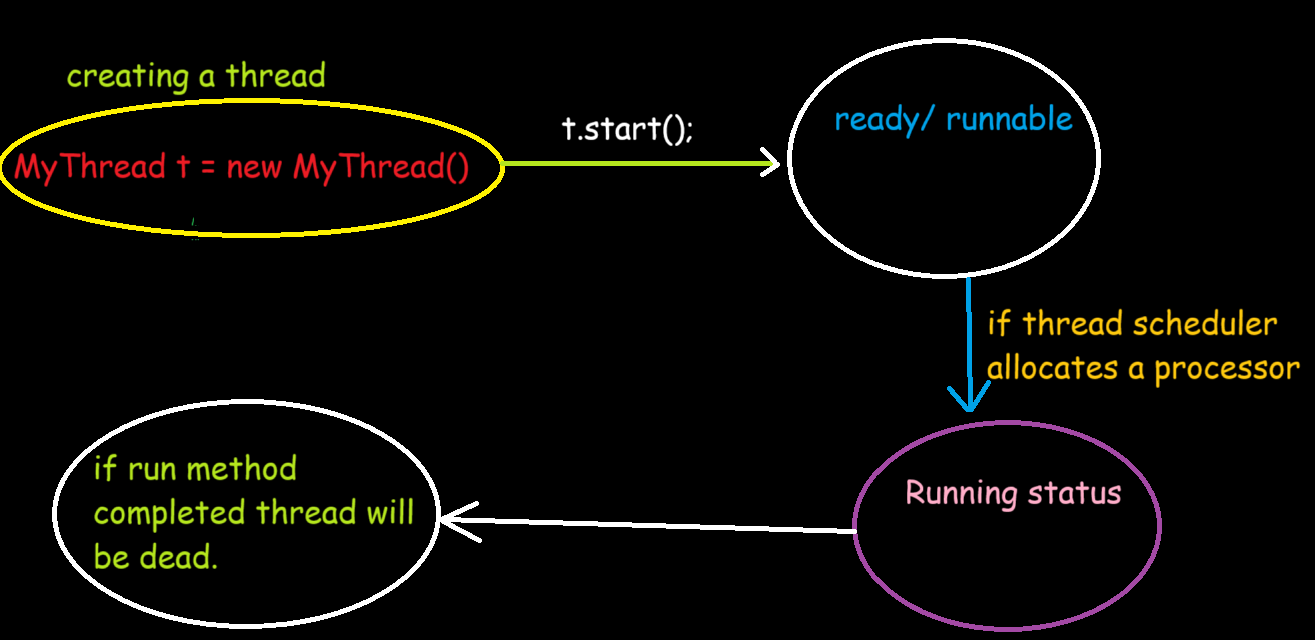
There is a concept of overload in threads. But start () method always calls a no args method only. Overloaded method we always calls a explicitly like a normal method call.



Override start ()?

If we override start () method then run () method won’t be call. It works like normal method.

Thread life cycle?



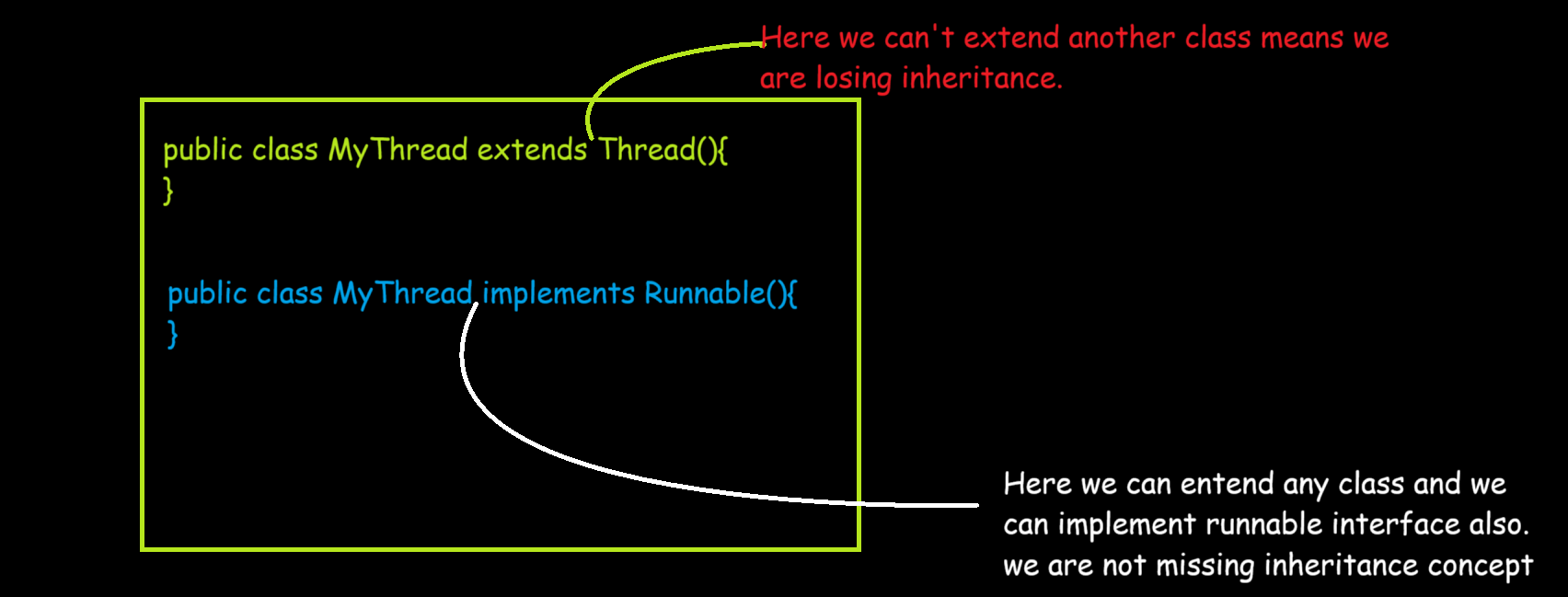
After starting thread can we start thread again?

Once we call start method, the thread will start and if we call again start method we will get illegalThreadStateException error.

Define a thread by implementing Runnable interface?



1. If we extend threads also internally thread implements runnable interface only.



1. It is recommended to create thread by using 2and approach like to implement runnable interface. Because we can use inheritance concept.