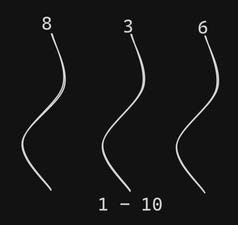
Multithreading

⇒ Thread class Methods :-
1. thread status
2. naming a thread
3. daemon thread
4. thread priority
5. Sleeping a Thread
6. Yielding a Thread
7. Joining a Thread
8. Interrupting a Thread

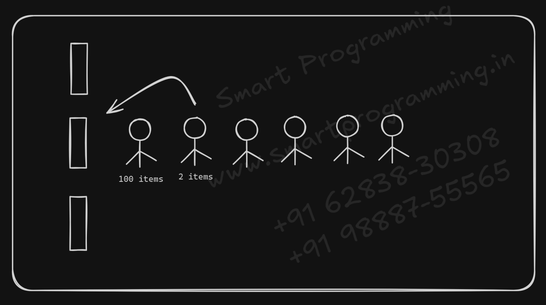
⇒ Thread Priority :-
→ It is the integer value through which thread scheduler decides which thread it has to execute first and then which thread to execute next
→ Thread priority ranges from 1 - 10
→ there are 3 pre-defined thread-priorities :-
1. MIN_PRIORITY - 1
2. NORM_PRIORITY - 5
3. MAX_PRIORITY - 10
→ By default every thread priority is 5 (NORM_PRIORITY)
→ Priority methods :-
1. setPriority(-)
2. getPriority()



→ Thread priority provides the hint to thread schedular which thread to execute first and then which thread, but it totally depends on thread schedular to accept that hint or not.

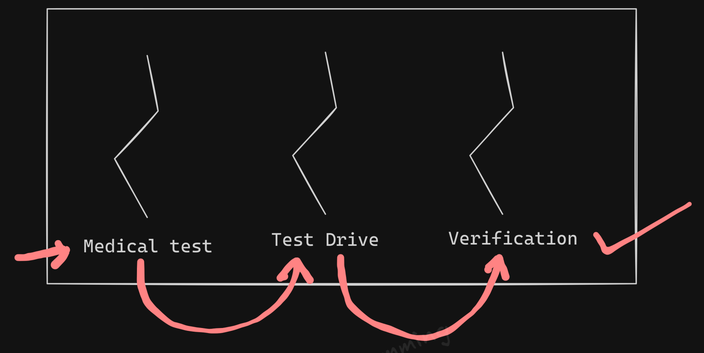
⇒ Sleeping a Thread :-
→ We are pausing the thread execution for some period of time.
→ Points :-
1. sleep() method is overloaded
2. sleep() method throws an Exception, so we have to use try-catch block or throws keyword
→ use :-
1. PPT
2. Animations
3. Digital Clock
4. Blinking blubs
etc

⇒ Yielding a Thread :-
→ This method will pause the current running thread and provide the chance to another thread to execute
→ Real World example :-



→ yield() provides the hint to thread schedular that I want to pause and provide chance to another thread, but it totally depends on thread schedular to accept that hint or not.
→ Its output is also depends on Thread Schedular so output can be changed

⇒ Joining a Thread :-
→ This method will wait for another thread to complete its task and starts after that thread.
→ Real World Example :
= Licence Department



⇒ Thread Interruption :-
→ This is the process by which a signal is provided to stop the thread gracefully.
→ Three methods :-
1. interrupt()
2. isInterrupted()
3. interrupted()

Interview Questions

⇒ Thread Priority
1. What is thread priority in Java?
2. What is the default priority of a thread?
3. What are the valid priority values in Java?
4. How do you set and get the priority of a thread?
5. What is the purpose of the constants MIN_PRIORITY, NORM_PRIORITY, and MAX_PRIORITY?
6. Does setting thread priority guarantee execution order? Why or why not?
7. How does the JVM handle threads with the same priority?
8. Can thread priorities behave differently on different platforms?

⇒ Thread Sleep
9. What does the sleep() method do in Java?
10. How is Thread.sleep() different from Object.wait()?
11. Does calling sleep() release the lock if the thread is synchronized?
12. What happens if we give a negative time to Thread.sleep()?
13. What exceptions must be handled when using sleep()?
14. Can we wake a sleeping thread manually?

⇒ Thread Join
15. What is the purpose of the join() method in Java?
16. How does join() affect the execution of threads?
17. What is the difference between join() and sleep()?
18. What happens if a thread calls join() on itself?
19. Can we specify a timeout with join()? What happens if the timeout is reached?
20. What exception can join() throw?

⇒ Thread Yield
21. What does the yield() method do?
22. Does yield() guarantee that another thread will run? Why or why not?
23. What is the difference between yield() and sleep()?
24. In what scenarios might yield() be useful?
25. How does thread scheduling impact the behavior of yield()?

⇒ Thread Interruption
26. What is the purpose of interrupt() in Java?
27. How does a thread check if it has been interrupted?
28. What is the difference between isInterrupted() and interrupted()?
29. What happens if we call interrupt() on a thread that is sleeping?
30. Does interrupt() stop a thread immediately? Why or why not?
31. How should a thread respond to an interruption signal?
32. Can interruption be used as a mechanism to stop threads?
33. What happens if you call interrupt() on a thread that is not in a blocking state?