:- Spring Scheduling :-

To execute a task simultaneously by container (without user interface) based on "Period of Time" or "Point of Time" scheduling are used.

- Here period of Time indicates Time gap.
  No starting and ending hours or days.
- Point of Time indicates start and end date and time.

Example - Period of Time - 1 week, 6 Months, 3 days, 8 sec, 6 hours, i.e. [Time gap]

Point of Time - 11.00 AM, 12th Jan, 1:32:41 PM [Exact time]

* To do scheduling write one method, it should be public, void and have no params. and apply @Scheduled over method.

** At AppConfig level add @EnableScheduling.

→ Then this method will be called by container automatically.

→ To activate this procen using

(a) XML Configuration → <task : annotation-driven>

[use task schema in beans tag]

(b) Java Configuration → @ Enabled Scheduling

[write in AppConfig class]

1. FixedDelay - time in milli sec.

→ This is a concept of Period of Time.

After certain time gap b/w method will be called by container. (works in loop).

example

#1 Create on maven project

#2 Add below dependency in pom.xml.

Spring - context - 5.1.4.RELEASE

#3 AppConfig.

```
package com.app.config;

@Configuration
@ComponantScan ("com.app")
@Enable Scheduling
public class AppConfig { }
```

-ii- Fixed Rate :- It indicates max wait time for next method call including method execution time.

Where as fixedDelay gives exact gap b/w last method end to next method call

Case #1 → Fixed Rate > method execution time

then wait = FixedRate - method exe. time

e.g.

FixedRate = 5 sec, method time = 3 sec

→ wait = 2 sec

m1	wait	m1	wait	m1	----
3sec	2sec	3sec	2sec	3sec	

Case #2 FixedRate <= method exec time

e.g. then wait = 0 sec i.e. no wait call again n agn

e.g. fixedRate = 3 sec, method time = 5 sec

m1	m1	m1	m1	----
5sec	5sec	5 sec	5 sec	

-!- Cron Expressions :- This concept is used to indicates Point of time or Period of time.

Format

sec	MIN	HRS	DAY	MONTH	WEEK
0-59	0-59	0-23	1-31	1-12	SON-SAT

eg: 0 0 10 * * *

→ Every day morning 10:00:00 AM do task

#2 20 * * * * *

→ every minute 20th sec do task

eg: 9:01:20 , next 9:02:20, next 9:03:20

#3 0 0 9,10,16 * * *

→ every day 9:00:00 AM, 10:00:00 AM and 4:00:00 PM

→ Common (,) indicates multiple values.