Git repo : https://github.com/ashokitschool/SBMS-39.git

==========
Project development
==========

=> Project means collection of classes

=> In a project we can see below types of classes

1) POJO classes (DTO / VO / Binding / Command) : To represent data

2) Component Classes (controller/service/dao/util) : logic

Controller --> logic to handle req & resp

Service --> business logic (validation, otp, reports)

DAO --> logic to communicate with DB

Utils --> email ops, pwd ops, date ops...

==========
How one java class method can access another java class method ?
==========

1) Inheritence (IS-A)

2) Composition (HAS-A)

Note: With IS-A and HAS-A our classes will become tightly coupled.

=> To overcome this tightly coupling we need to develop our classes with strategy design pattern (GOF).

1) Favour composition over inheritence

2) Code with interfaces

3) Code should be open for extension

==========
What is dependency injection ?
==========

=> The process of injecting dependent obj into target obj is called as dependency injection (DI).

=> We can perform this DI in 3 ways

1) Field Injection
2) Setter Injection
3) Constructor Injection

=> Injecting dependent obj into target class variable directley is called as FI.

=> Injecting dependent obj into target class obj using target class constructor is called as CI.

=> Injecting dependent obj into target class obj using target class setter method is called as SI.

Note: In normal java apps, programmers are responsible to perform DI.

Note: If we use spring core then IOC container will take care of Dependency Injection.

==========
What is IOC ?
==========

=> Inversion of control

=> IOC is a principle which is responsible to manage and colloborate dependencies among the objects in the application.

=> IOC will take care of Dependency Injection.

1) What is IOC container

2) How many ways we can start IOC

3) Dependency Injection

4) How many ways are there for DI

5) What is CI

6) What is SI

7) What is FI

8) What is Spring Bean ?

```
===============
STS IDE Setup
===============
```

=> Download sts ide jar using below link

https://cdn.spring.io/spring-tools/release/STS4/4.21.1.RELEASE/dist/e4.30/spring-tool-suite-4-4.21.1.RELEASE-e4.30.0-win32.win32.x86_64.self-extracting.jar

=> Extract jar file using below command

```
java -jar <jar-file-name>
```

=> Go to STS folder and open IDE using .exe file

```
===================================
First app development using spring
===================================
```

1) Create Maven project (Simple) using IDE (Eclipse/STS/IntelliJ)

2) Add 'spring-context' dependency in project pom.xml file

```
URL : www.mvnrepository.com
```

3) Create Required java classes using strategy design pattern

4) Create Spring Beans Configuration file

5) Create Test class to test our app with IOC container

```
------------------------------------------------------------
<property /> : Represents setter injection

<constructor-arg /> : Represents constructor injection
------------------------------------------------------------

============
Bean Scopes
```

```
============

=> Scope will decide when to create obj for spring bean and how many objects should be created for
spring bean.

=> We have 4 types of scopes in spring

                1) singleton (default)
                2) prototype
                3) request
                4) session

=> Singleton means only one object will be created
                (at the time of ioc startup)

=> Prototype means everytime new object will be created when we call getBean ( ) method.


Note: request and session scopes are used in spring web mvc module.


Note: IOC will perform Eager loading for singleton beans and it will perform lazy loading for
prototype beans.

=================
Bean life cycle
=================

Thread Life Cycle  : will be managed by thread schedular

Servlets Life Cycle : will be managed by Servlet Container

Spring Bean Life Cycle : will be managed by IOC container


init-method : After bean obj creation

destory-method : When ioc is closed

<bean id="eng" class="in.ashokit.Engine"
                                init-method="m1"
                                destory-method="m2" />

-----------------------------------------------------------------

============
Autowiring
============

If we perform DI using property or constructor-arg tags then it is called as Manual wiring.

We will use ref attribute to specify dependent bean

        Ex: <property name="eng" ref="de"/>


-----------------------------

1) Manual wiring (programmer should specify what is dependent)

2) Auto wiring  (IOC will identify dependent bean obj)

=> TO use Autowiring configuration changes are required

=> we have Autowiring modes

                1) byName (identify dependent based on bean name)

                2) byType (identify dependent based on bean type)
```

```
                3) constructor

                4) none (default)


=====================
Collection Injection
=====================

public class Student {

        private Integer id;

        private String name;

        private List courses;

        private Set hobbies;

}


java.lang.Object ====> default super class

public String toString(){

 return this.getClass().getName()+"@"+
                Integer.toHexString(this.hashCode());
}

in.ashokit.Student@797lhkl



in.ashokit.Student@302552ec



how to inject object  (ref)

how to inject value (int, string....)  (value)

how to inject collections (list, set, map)


1) What is Spring Bean
2) Bean Scopes
3) Bean Life Cycle
4) Autowiring (byName, byType, Constructor)
5) Collection Injection (list, set, map)
6) What is IOC
7) Dependency Injection
8) Setter Injection
9) Constructor Injection
10) Field Injection
11) Strategy Design Pattern
12) What is Bean Configuration file
```