Spring AOP.

Create spring boot project with dependencies web, dev tools

Add below dependency in pom.xml

```
<dependency>
	<groupId>org.springframework.boot</groupId>
	<artifactId>spring-boot-starter-aop</artifactId>
</dependency>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-
4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<parent>
		<groupId>org.springframework.boot</groupId>
		<artifactId>spring-boot-starter-parent</artifactId>
		<version>2.5.2</version>
		<relativePath /> <!-- lookup parent from repository -->
	</parent>
	<groupId>com.javabykiran</groupId>
	<artifactId>basicMicroserviceProject</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<name>basicMicroserviceProject</name>
	<description>First basic microservice or REST webservice project</description>
	<properties>
		<java.version>1.8</java.version>
	</properties>
	<dependencies>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-web</artifactId>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-aop</artifactId>
		</dependency>

		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-devtools</artifactId>
			<scope>runtime</scope>
			<optional>true</optional>
		</dependency>
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-test</artifactId>
			<scope>test</scope>
		</dependency>
	</dependencies>
```

```
	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```

Create advise as below.

```
package com.javabykiran.advise;

import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.springframework.stereotype.Component;

@Aspect
@Component //We can define bean with this name in configuration class
public class TimingAdvise {

	@Before("execution (* com.javabykiran.controller.StudentController.*())")
	public void printTimeForNow() {
		System.out.println("I am getcalled before any of method..." + new java.util.Date());
	}

}
```


If method has parameters then you need to give 2 dots in method.

```
@Before("execution (* com.javabykiran.controller.StudentController.*(..))")
```

Define Controller or any class where you want to apply advice – in our case it is controller class methods.

```
package com.javabykiran.controller;

import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class StudentController {

	@RequestMapping("firstservice")
	public String welcomeMessage() {
		return "I am in controller testing aop concept";
	}

}
```

Add in main class below annotations.

```
@EnableAspectJAutoProxy
```

Run project you will see advice is getting called before controller method.

Homework:

1. There are four types of advice we used before 3 are remaining so change our advice to below
    a. @After
    b. @Around
    c. @AfterThrowing
2. Create class student and mention m1 method in it.
    a. Do configuration in such a way that before me get called from controller our advice should get executed.

Tutorial to read:

https://www.jbktutorials.com/spring-aop/configure-project-for-spring-aop-with-java-configuration-annotations.php#gsc.tab=0

Download Project:

https://javabykiran.in/springboot/basicMicroserviceProject-AOP-IOC.zip

Homework:

1. Separate Database code from controller to other class and annotate that class with @Repository and autowire that class in controller to call method.
2. Separate session factory code from the configuration file that is starter class to some other class and annotate that class with @Configuration.
3. Above 2 points will separate code between 2 layers.