EXERCISE 8

**Implementing Basic AOP with Spring**

**Scenario:** The library management application requires basic AOP functionality to separate cross-cutting concerns like logging and transaction management.

**Steps:**

1. **Define an Aspect:**

@Aspect

@Component

public class LoggingAspect {

@Before("execution(\* com.library.service.\*.\*(..))")

public void logBefore(JoinPoint joinPoint) {

System.out.println("Executing: " + joinPoint.getSignature().getName());

}

@After("execution(\* com.library.service.\*.\*(..))")

public void logAfter(JoinPoint joinPoint) {

System.out.println("Finished: " + joinPoint.getSignature().getName());

}

}

1. **Create Advice Methods:**
   * Define advice methods in LoggingAspect for logging before and after method execution.
2. **Configure the Aspect:**
   * Update applicationContext.xml to register the aspect and enable AspectJ auto-proxying.

<aop:aspectj-autoproxy/>

<bean id="loggingAspect" class="com.library.aspect.LoggingAspect"/>

1. **Test the Aspect:**
   * Run the LibraryManagementApplication main class to verify the AOP functionality.