Modern Topics in IT

Assignment 03 : Aspect Oriented Programming

Objective :

The objective of this assignment is to check students' knowledge regarding Aspect Oriented Programming using spring AOP or AspectJ frameworks. Students should be able to use AOP for business application implementation.

Percentage of overall : 12.5%

Assignment created by: Udara Samaratunge

AOP Assignment 03

Guidelines:-

A. Assignment is an open ended assignment. **Creativity is up to you to decide**.

B. You can implement either **web based or standalone** applications with connecting to any database.

C. You can use either **AspectJ or Spring AOP** frameworks.

D. Marks will be allocated for given **deliverables,** and **marking scheme** is given below.

E. You will be given **2nd of April 2016 as deadline.**

F. **Both members must do the coding. (1 member** can implement **service layer** and **other member** can implement **data-access layer**. Both should implement Aspects.

G. If the code has been **directly copied** from the **internet or from a pier at any level**, it will be considered as **a plagiarism act and dealt with accordingly**. (Please note that we will be using software to check for any plagiarism)

Question:

1) You should implement the scenario to persist transactional data in to database. Separate the implementation in layer wise (Service layer & Data access layer) and use the AOP concept to implement logging aspects. You can use either **Spring AOP or AspectJ frameworks** for implementation.

    a) **Service Layer** – **Interface** of business service and **Service implementation class**. (One service implementation is enough) If you implement more than one service extra marks would be given for creativity.

    b) **Data Access Layer** – **Interface** (for CRUD operations) and **Data Access implementation** for create, remove, update and delete transactional data.

    c) You should implement **Before Advice, After Advice and after-throwing advices** (for exception handling). As well represent the behavior of exceptional scenario using **after-throwing advice**.

    d) **AOP Logging Aspect** should work for **both Service-level** method implementation and **Data-Access level** method implementation.

    e) Layers should be well packaged and keep classes in relevant packages. (**com.mtit.service, com.mtit.dataaccess and com.mtit.aspects)**

    f) You can connect this application to any database

Deliverables:

Source codes of implemented scenario. Source code should be packaged and well commented.

Report: (You should provide a report that include below details)

a) Explain the Scenario using a diagram.
b) High level class diagram & sequence diagram of method invocation.
c) Explain implemented aspect methods or point cut expressions
d) Sample screenshots of Logger outputs.

[**Important**: - During the evaluation time if application didn't run successfully, by looking at **screen shots** in the **report,** marks would be given]

Marking Scheme

Feature	Marks
Service layer implementation [interface implementation + ServiceImpl implementation]	20%
Data access layer implementation [interface implementation + DataAccess implementation]	20%
AOP Advices implementation AOP Before advice – 10% AOP after advice – 10% AOP after throwing advice – 10%	30%
Report	20%
Creativity of the scenario (Coding standards & adding class level, method level and single line comments)	10%
Total Marks	**100%**