Advanced Software Engineering

ASSIGNMENT 6.1

Aspect-Oriented Software Development Using AspectJ

DUE DATE

Tuesday, October 14, 2025 at 11:59 PM CT

POINTS

100 Points

In this assignment, you will modify a MoneyTransfer package. By completing this assignment, you will gain experience with Aspect-Oriented Software Development (AOSD) using AspectJ.

PREPARE

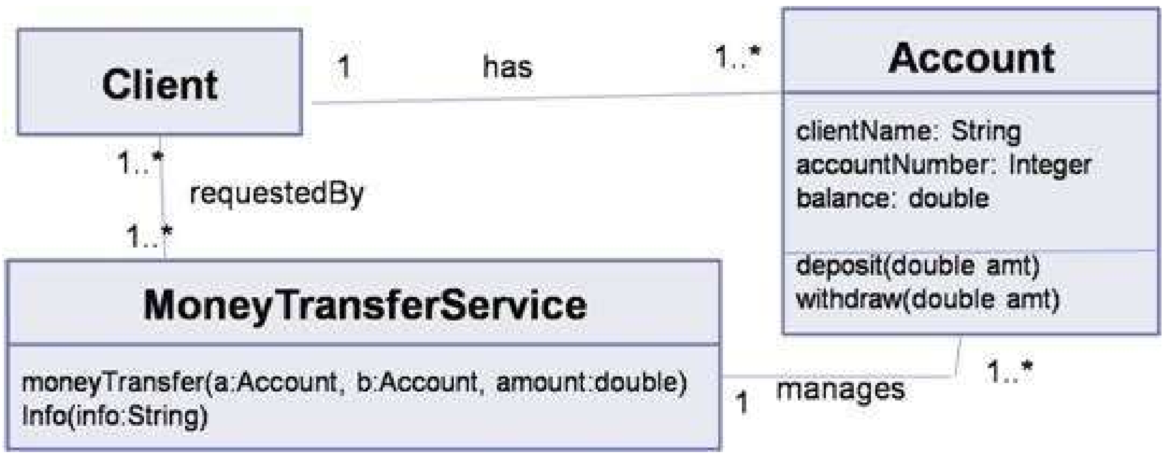
Complete the following tasks before attempting to complete this assignment.

- Complete all assigned readings on the **Week 6 Overview page (https://baylor.instructure.com/courses/242214/pages/week-6-overview)**
- Review all Week 6 lessons
- Read ***I want my AOP!, Part 1* *(https://www.infoworld.com/article/2073918/i-want-my-aop---part-1.html)*** by Ramnivas Laddad
- Download the latest stable release of AspectJ from the **Eclipse Foundation's AspectJ GitHub (https://eclipse.dev/aspectj/)** page
- Download the **MoneyTransfer package (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/MoneyTransfer.zip)**

You may also choose to review the following additional AspectJ resources:

- **AspectJ web site (http://www.eclipse.org/aspectj/)**
- **Programmers Guide (http://www.eclipse.org/aspectj/doc/released/progguide/index.html)**
- **Quick Reference (PDF) (http://www.eclipse.org/aspectj/doc/released/quick5.pdf)** (an excelled guide to teh language syntax)
- Refer to either resource for compilation instructions:
  - **Development Environment Guide (http://www.eclipse.org/aspectj/doc/released/devguide/index.html)**
  - **ajc compiler page (http://www.eclipse.org/aspectj/doc/released/devguide/ajc-ref.html)**

CLASS DIAGRAM



The above class diagram represents objects involved in a money transfer service. A money transfer from an account **a** to another account **b**, involves withdrawing money from account a and depositing money into account b. Implementations of these classes are presented in **Client.java (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/Client.java.txt)**, **Account.java (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/Account.java.txt)**, **AroundAdviceAspect.java, (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/AroundAdviceAspect.java.txt)** and

MoneyTransferService.java (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/MoneyTransferService.java.txt). These classes are part of the **MoneyTransfer package**.

The **MoneyTransfer package** includes the following AspectJ aspect samples:

1. The **MoneyTransferServiceAspect.java (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/MoneyTransferServiceAspect.java.txt)** contains AspectJ aspects that:
   - Prints a message indicating the start of a money transfer transaction.
   - Prints a message indicating the end of a money transfer transaction.
   - Double the deposit amount for each deposit transaction *using a second call to the deposit method for each money transfer.*
2. The **AroundAdviceAspect.java (https://courseapps.onlinecs.baylor.edu/media/CSI5354/Files/week6/AroundAdviceAspect.java.txt)** contains AspectJ aspects that:
   - Adds a method to the MoneyTransferService class to log each transaction and uses an around advice to log each transaction.

INSTRUCTIONS

For *each* of the following, write a new aspect with relevant JUnit tests:

1. Double the amount of each deposit by doubling the values of the parameter supplied to the deposit method.
2. Double the amount of each deposit by adding the deposit amount to the balance in the account.
3. Add a money transfer benefit for the customer by adding 10% to the deposit amount for each money transfer.
4. Implement a service fee by charging a fixed fee, (e.g., $1) to each account involved in a money transfer.
5. Prints a message every time the account balance is set.
6. Prints a message every time a method executes.
7. Grant a 10% bonus for each money transfer greater than $5,000 and 5% for amounts between $1,000 and $5,000. The bonus must be credited to the account from which funds are withdrawn.
8. Reduce the amount withdrawn by 50% for every third money transfer.
9. Write three more aspects of your own choice. At least one of them should include one or more inter-type declarations. Bonus points will be given when your aspects deal with more critical crosscutting properties (such as security, performance) and/or take the best use of various AspectJ features (such as w.r.t. the type of advices/pointcuts).

Note: No change is allowed for the following files: Account.java, MoneyTransferService.java, AroundAdviceAspect.java, and MoneyTransferServiceAspect.java.

Submit the following in one zip file:

- A summary of your aspect creation and testing
- All new aspect source files with tests

Submission Guidelines

Title: Assignment Submission Guidelines

This content is not printable; please log into your course to view this content.

Start Assignment

- **Due** Tuesday by 9:59pm
- **Points** 100
- **Submitting** a file upload