Lab Assignment 1

Object Oriented Programing (CSC241)
SP25-BCS (Section A)

Assignment Date: 04-03-2025 Due Date: 04-03-2025 (within lab slot)
Maximum Marks: 10

Secure Banking System (CLO-4 Applying)

Apply Object-Oriented Programming concepts to create a banking system that requires a secure PIN verification process for users to access their accounts. The system has a single registered user with a fixed account number and PIN. To access the account:

- The user must enter their account number and PIN via `System.console()`.
- The PIN is validated using a non-static inner class to ensure security.
- If the credentials match, access is granted to view the account details.
- The verification process is handled using a static inner class inside BankingSystem.

Students must implement the following folder and package structure to maintain code organization and modularity.

Step-by-Step Process

1. User enters PIN using System.console().
2. PinValidator's Rules class checks if the PIN is exactly 4 digits.
   `pin.length() == 4 && pin.matches("\\d+")`
3. If valid, the system proceeds to authenticate it against the stored PIN.
4. If invalid, the user gets an error message and must enter a correct format.

Folder & Package Structure

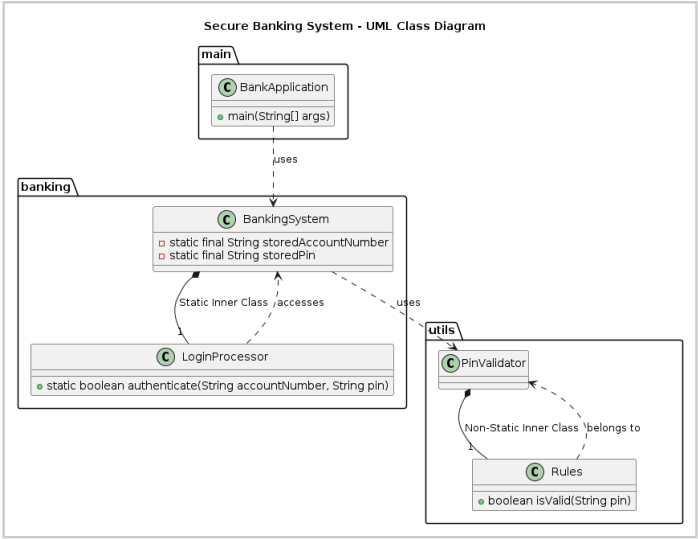
Students must organize their project using the following structure:

```
C:\>tree C:\JavaLabs\A\ /F
Folder PATH listing
Volume serial number is B421-883E
C:\JAVALABS\A
    run.bat

├──out
    ├──banking
            BankingSystem.class

    ├──main
            BankApplication.class

    ├──util
            PinValidator.class

    └──utils
└──src
    ├──banking
            BankingSystem.java

    ├──main
            BankApplication.java

    └──utils
            PinValidator.java
```

Each package should contain the appropriate class implementation with proper modularization and encapsulation.

UML Class Diagram Instructions

Students must carefully follow the provided UML Class Diagram while implementing the banking system. The diagram outlines the class structure, relationships, and method responsibilities, ensuring proper use of static and non-static inner classes. Maintain class names, method signatures, and package structure as shown in the diagram to ensure consistency and clarity in your implementation.



Submission Instructions

Git Repository Upload:

- Students must upload their completed project to a Git repository.
- Once uploaded, the repository link must be submitted via the provided form.

Submission Form:

- Fill out the submission form with your repository details:
  Submit Repository Link Here

Important:

- Once uploaded, do not modify your repository.
- Ensure that your submission follows the correct package structure and naming conventions.