CSEN160 OO Analysis, Design and Programming Fall 2025

Lab 8 write JUnit tests for Bank Management System

1. **Accept** the **Lab 8** assignment using GitHub Classroom link with your Github account.

2. Clone your GitHub Classroom repository using IntelliJ IDEA.

3. Complete **Lab 8 JUnit tests** as described in the comments and the requirements in this file.

4. Use **JUnit tests** to verify your work.

6. Use **Git** to submit your code to your GitHub Classroom repository.

7. After completing all the steps above, be prepared to **demo** your code to me.

In this lab, you will use JUnit to test a Bank application.

All the exercises in this lab should be included in two packages:

- **edu.scu.csen160.lab8.code**
- **edu.scu.csen160.lab8.test**

Objective

Study the given Java application for managing bank accounts with functionalities like account creation, balance checking, deposits, withdrawals, and fund transfers. Write comprehensive unit tests using JUnit to verify the correctness of the implementation.

Lab Requirements

1. Tests for BankAccount (6 tests)

1.1. testDeposit

Description: This test checks that depositing a valid amount correctly increases the account balance.

1.2. testDepositNegative

Description: This test checks that depositing an invalid negative amount is properly handled by the account.

1.3. testWithdraw

Description: This test checks that withdrawing a valid amount correctly decreases the account balance.

1.4. testWithdrawInsufficientFunds

Description: This test checks that withdrawing more than the current balance is handled correctly by the account.

1.5. testTransfer

Description: This test checks that transferring money updates both the sender and receiver balances correctly.

1.6. testTransferToNullAccount

Description: This test checks that transferring money to a null target account is handled properly.

2. Tests for Bank (4 tests)

2.1. testCreateAccount

Description: This test checks that a new account is created and stored with the correct initial balance.

2.2. testCreateAccountWithDuplicateId

Description: This test checks that creating an account with a duplicate ID is handled correctly by the bank.

2.3. testDeleteAccount

Description: This test checks that after deleting an account, it can no longer be retrieved.

2.4. testGetTotalFunds

Description: This test checks that the bank correctly reports the total balance of all accounts.