

Customer Menu Manual Test Case Designs

Test Case 1: `validDepositTest`

Objective: Verify that a valid deposit amount is successfully processed.

Test Step	Expected Result
Call <code>accountA.deposit(500)</code> .	No output message about invalid deposit.

Test Case 2: `invalidDepositTest`

Objective: Ensure that an invalid deposit amount (e.g., 0 or negative) is rejected.

Test Step	Expected Result
Call <code>accountA.deposit(0)</code> .	Output contains: "Amount to deposit should be positive." .

Test Case 3: `validWithdrawalTest`

Objective: Verify that a valid withdrawal amount is successfully processed.

Test Step	Expected Result
Deposit 1000 into <code>accountA</code> .	Deposit is successful.
Call <code>accountA.withdraw(500)</code> .	No output message about insufficient balance or invalid amount.

Test Case 4: `invalidWithdrawalTest`

Objective: Ensure that an invalid withdrawal amount or insufficient funds scenario is handled properly.

Test Step	Expected Result
Attempt to withdraw 500 from accountA without sufficient funds.	Output contains: "Insufficient balance or invalid amount." .

Test Case 5: validTransferTest

Objective: Verify that a valid transfer between accounts is successfully processed.

Test Step	Expected Result
Deposit 1000 into accountA .	Deposit is successful.
Transfer 500 from accountA to accountB .	No output message about insufficient balance or invalid amount.

Test Case 6: invalidTransferTest

Objective: Ensure that an invalid transfer (e.g., insufficient funds or invalid amount) is handled properly.

Test Step	Expected Result
Deposit 1000 into accountA .	Deposit is successful.
Attempt to transfer 5000 from accountA to accountB .	Output contains: "Insufficient balance or invalid amount." .

All-DU-Paths Coverage

1. **Deposit Functionality:** Covered both valid and invalid deposit paths, ensuring proper message outputs for invalid inputs.
2. **Withdrawal Functionality:** Ensured valid and invalid scenarios are tested, including edge cases like insufficient funds.

3. **Transfer Functionality:** Covered valid and invalid transfer paths to ensure proper handling of inter-account transactions.