

Problem Statement: Banking System with Custom Exception Handling

Background:

You are tasked with designing a simple banking system to manage customer accounts and transactions. The system should allow users to create accounts, deposit money, withdraw money, and transfer money between accounts. However, certain exceptions and errors should be handled in a custom way.

The primary issue to be addressed is the handling of **invalid transactions**, such as:

- **Insufficient funds** during a withdrawal.
- **Invalid account number** during a transaction.
- **Negative deposit amounts**.
- **Exceeding withdrawal limits** (for example, withdrawing more than allowed per day).

These errors should be thrown as custom exceptions so that they can be clearly identified and handled appropriately.

Scenario:

You are given the following requirements for the banking system:

1. Account Class:

- Each account has:
 - **Account Number** (a unique identifier)
 - **Account Holder Name**
 - **Balance** (initially 0)
- **Methods:**
 - **deposit(double amount)**: Deposits a specified amount into the account. If the amount is negative, throw a **NegativeDepositException**.
 - **withdraw(double amount)**: Withdraws a specified amount from the account. If the amount exceeds the balance, throw an **InsufficientFundsException**. If the withdrawal exceeds a daily limit (e.g., \$500), throw a **DailyLimitExceededException**.
 - **transfer(double amount, Account& recipient)**: Transfers money between accounts. If either account is invalid (e.g., account number does not exist), throw an **InvalidAccountException**.

2. Banking System Class:

- This class manages a list of customer accounts.
- **Methods:**
 - **addAccount(Account newAccount)**: Adds a new account to the system.
 - **findAccount(int accountNumber)**: Finds an account by account number. If no account is found, throw an **InvalidAccountException**.

3. Custom Exceptions:

- **NegativeDepositException:** Thrown when a deposit amount is negative.
- **InsufficientFundsException:** Thrown when there are not enough funds in the account to complete a withdrawal.
- **DailyLimitExceededException:** Thrown when the withdrawal amount exceeds the daily limit.
- **InvalidAccountException:** Thrown when an invalid account number is used for a transaction.

Problem Requirements:

1. Implement a **BankAccount** class with the specified methods and exception handling for deposit, withdrawal, and transfer operations.
2. Implement a **BankingSystem** class that keeps track of multiple customer accounts and manages exceptions when trying to access accounts or perform transactions.
3. For each custom exception, provide an appropriate error message when it is thrown.
4. Create a simple program where the user can interact with the system by:
 - Creating an account.
 - Depositing money.
 - Withdrawing money (with checks for insufficient funds and daily withdrawal limit).
 - Transferring money between accounts (with checks for invalid account numbers).
5. Handle all exceptions in a way that the system continues running after an exception is thrown (i.e., no crash).

Example of Expected Behavior:

1. **Creating an account:** Should succeed with a valid account number and name.
2. **Depositing money:** Should succeed unless the deposit amount is negative (in which case, a **NegativeDepositException** is thrown).
3. **Withdrawing money:**
 - Should succeed if there are enough funds and the daily limit is not exceeded.
 - Should throw an **InsufficientFundsException** if there are not enough funds.
 - Should throw a **DailyLimitExceededException** if the withdrawal exceeds the allowed daily limit.
4. **Transferring money:**
 - Should succeed if the accounts are valid and the transfer can be completed.
 - Should throw an **InvalidAccountException** if an invalid account number is used.

Example of Error Scenarios:

- **Negative Deposit:**
 - If a user tries to deposit **-100** into their account, the program throws a **NegativeDepositException** with the message: "Error: Deposit amount cannot be negative!".

- **Insufficient Funds:**

- If a user tries to withdraw **1000** from an account with a balance of **500**, the program throws an **InsufficientFundsException** with the message: **"Error: Insufficient funds in your account!"**.

- **Exceeding Daily Limit:**

- If a user tries to withdraw **600** from their account (with a daily limit of **500**), the program throws a **DailyLimitExceededException** with the message: **"Error: Daily withdrawal limit exceeded!"**.

- **Invalid Account:**

- If a user tries to transfer money to an invalid account (e.g., an account number that doesn't exist), the program throws an **InvalidAccountException** with the message: **"Error: Invalid account number!"**.

Additional Notes:

- Ensure that the system is able to gracefully handle these exceptions without crashing.
 - Consider implementing logging or error messages to help track when exceptions are thrown.
-