Bank Account System - OOP Task

Overview

Create a BankAccount class in C++ demonstrating OOP concepts: constructors, encapsulation, and operator overloading.

Class Requirements

Private Attributes:

- accountNumber (int)
- balance (double)
- accountType (string) // (e.g., "Savings", "Current", "Student")

Constructors (use initialization lists):

1. Default - sets default values
2. Parameterized - accepts all three parameters
3. Copy - copies another BankAccount object

Member Functions:

1. deposit(amount) - add money (validate positive amount)
2. withdraw(amount) - remove money (validate positive & sufficient balance)
3. checkBalance() - display current balance
4. transfer(targetAccount, amount) - move money between accounts
5. display() - show all account info

Required Operator Overloads:

1. operator+= - deposit operation

   Usage: account += 500; // Deposits 500 EGP

2. operator-= - withdraw operation

   Usage: account -= 200; // Withdraws 200 EGP

3. **Two comparison operators** (choose from: >, <, ==, >=, <=) - compare balances

Usage Example:

```cpp
if (account1 > account2) {
    cout << "Account 1 has more money" << endl;
}
```

Bonus: operator>> - transfer between accounts

Usage: account1 >> account2; // Transfer from account1 to account2

Validation Rules

- All amounts must be positive
- Withdrawals/transfers need sufficient balance
- Show clear error/success messages

Testing

Your main() should test:

- All three constructors
- All functions with valid and invalid inputs
- All operators
- Edge cases (negative amounts, insufficient funds)