Assignment\_06-08-2024

Author: ThanhTH

Date: 07/08/2024

Question: Explain friend function and friend class real time example with banking applicationExplain friend function and friend class real time example with banking application

A friend function or friend class is a function or class that is granted access to the private and protected members of another class. This is useful when we want to allow a specific function or class to access the internal state of an object without making the internal state public.

Consider a banking application where we have a BankAccount class that represents a customer's bank account. The BankAccount class has private members accountNumber and balance, which we don't want to make public for security reasons.

However, we want to allow a Transaction class to access the balance member of BankAccount to perform deposit and withdrawal operations. We also want to allow a PrintStatement function to access the accountNumber and balance members of BankAccount to print the account statement.

***Example***

#include <iostream>

class Transaction;

class BankAccount

{

private:

    int accountNumber;

    double balance;

    // Declare PrintStatement as a friend function

    // This means PrintStatement can access private members of BankAccount

    friend void PrintStatement(const BankAccount &account);

    // Declare Transaction class as a friend class

    // This means Transaction class can access private members of BankAccount

    friend class Transaction;

public:

    BankAccount(int accountNumber, double balance)

        : accountNumber(accountNumber), balance(balance) {}

};

// Friend function to print account statement

// Can access private members of BankAccount because it's a friend function

void PrintStatement(const BankAccount &account)

{

    std::cout << "Account Number: " << account.accountNumber << std::endl;

    std::cout << "Balance: " << account.balance << std::endl;

}

// Transaction class

// Can access private members of BankAccount because it's a friend class

class Transaction

{

public:

    void deposit(BankAccount &account, double amount)

    {

        // Accessing private member balance of BankAccount

        account.balance += amount;

    }

    void withdraw(BankAccount &account, double amount)

    {

        // Accessing private member balance of BankAccount

        if (account.balance >= amount)

        {

            account.balance -= amount;

        }

        else

        {

            std::cout << "Insufficient balance!" << std::endl;

        }

    }

};

int main()

{

    BankAccount account(12345, 1000.0);

    PrintStatement(account);

    Transaction transaction;

    transaction.deposit(account, 500.0);

    PrintStatement(account);

    transaction.withdraw(account, 200.0);

    PrintStatement(account);

    return 0;

}

Output:

