#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class Account {

private:

int accountNumber;

string accountHolderName;

double balance;

public:

void createAccount() {

cout << "\nEnter Account Number: ";

cin >> accountNumber;

cout << "\nEnter Account Holder Name: ";

cin.ignore();

getline(cin, accountHolderName);

cout << "\nEnter Initial Balance: ";

cin >> balance;

cout << "\nAccount Created Successfully!\n";

}

void showAccount() const {

cout << "\nAccount Number: " << accountNumber;

cout << "\nAccount Holder Name: " << accountHolderName;

cout << "\nBalance: " << balance << "\n";

}

void deposit() {

double amount;

cout << "\nEnter Amount to Deposit: ";

cin >> amount;

balance += amount;

cout << "\nDeposited Successfully!\n";

}

void withdraw() {

double amount;

cout << "\nEnter Amount to Withdraw: ";

cin >> amount;

if (amount <= balance) {

balance -= amount;

cout << "\nWithdrawn Successfully!\n";

} else {

cout << "\nInsufficient Balance!\n";

}

}

int getAccountNumber() const {

return accountNumber;

}

};

void createAccount();

void showAccount();

void depositOrWithdraw(bool deposit);

int main() {

int choice;

do {

cout << "\n\n\tBANK MANAGEMENT SYSTEM";

cout << "\n\n\t1. Create Account";

cout << "\n\t2. View Account";

cout << "\n\t3. Deposit";

cout << "\n\t4. Withdraw";

cout << "\n\t5. Exit";

cout << "\n\n\tSelect Option (1-5): ";

cin >> choice;

switch (choice) {

case 1:

createAccount();

break;

case 2:

showAccount();

break;

case 3:

depositOrWithdraw(true);

break;

case 4:

depositOrWithdraw(false);

break;

case 5:

cout << "\nThank you for using the Bank Management System.\n";

break;

default:

cout << "\nInvalid Option! Please try again.\n";

}

} while (choice != 5);

return 0;

}

void createAccount() {

Account account;

ofstream outFile("account.dat", ios::binary | ios::app);

account.createAccount();

outFile.write(reinterpret\_cast<char\*>(&account), sizeof(Account));

outFile.close();

}

void showAccount() {

Account account;

int accountNumber;

bool found = false;

cout << "\nEnter Account Number: ";

cin >> accountNumber;

ifstream inFile("account.dat", ios::binary);

while (inFile.read(reinterpret\_cast<char\*>(&account), sizeof(Account))) {

if (account.getAccountNumber() == accountNumber) {

account.showAccount();

found = true;

break;

}

}

inFile.close();

if (!found)

cout << "\nAccount not found!\n";

}

void depositOrWithdraw(bool deposit) {

Account account;

int accountNumber;

bool found = false;

cout << "\nEnter Account Number: ";

cin >> accountNumber;

fstream file("account.dat", ios::binary | ios::in | ios::out);

while (file.read(reinterpret\_cast<char\*>(&account), sizeof(Account))) {

if (account.getAccountNumber() == accountNumber) {

if (deposit) {

account.deposit();

} else {

account.withdraw();

}

file.seekp(-static\_cast<int>(sizeof(Account)), ios::cur);

file.write(reinterpret\_cast<char\*>(&account), sizeof(Account));

found = true;

break;

}

}

file.close();

if (!found)

cout << "\nAccount not found!\n";

}

