

Account Class

Create the new project *AccountClass*. Click the *Project* tab and select *Add New Item...*. Add a header file named *Account.h* and then repeat the process to add an implementation module named *Account.cpp* to hold C++ code. Use the class *Account*. Use this class description (public and private member function prototypes and private data members) to provide necessary definitions for *Account.h*:

```
#ifndef ACCOUNT_H
#define ACCOUNT_H
#include <string>
using namespace std;
static int accountNumber = 100000; // starting value for account#

class Account
{
public:
    Account(); // default constructor
    Account(string name, double amount); // constructor with parameters
    int getAccountID() const; // ACCESSOR member functions
    string getAccountHolder() const; // return name of account holder
    double getBalance() const; // return account balance
    void setAccountHolder(string name); // MUTATOR member functions
    void setBalance(double amt); // assign amount to balance
    void depositAmount(double amt); // add amount to balance
    void withdrawAmount(double amt); // subtract absolute value of amount from balance
    void displayAccountInfo() const; // display account information
private:
    int getNextAccountNumber() const; // get next account# (pre-increment account number)
    int accountID; // account# identifier
    string accountHolder; // name of account holder
    double balance; // account balance
};
#endif
```

Define a label *ACCOUNT_H* that indicates whether the *Account.h* header file has already been included in the project (it might be included by more than one of the files that make up a Visual Studio C++ project).

Code the implementation file *Account.cpp* to provide the operations of the class member functions declared in the header file. Be sure to include the scope identifier *Account::* in front of each function name in the implementation file to identify each function as part of the *Account* class. Include needed preprocessor directives and *using namespace std* for the code in these routines.

```
#include "stdafx.h"
#include "Account.h"
#include <iostream>
#include <iomanip>
using namespace std;
```

The member functions *set* and *get* values for the three *private* data members of the class. Some helper functions and carry out other useful operations – *displayAccountInfo()* and *getNextAccountNumber()*.

Provide two *constructors* for the *Account* class – a default constructor (no parameters) and a parameterized constructor that accepts the account holder name and starting account balance. Each constructor begins by getting the next account number from a routine that *increments* the number by 1 before returning the value.

```
Account::Account() // default constructor
{
    accountID = getNextAccountNumber(); // get next account#
    accountHolder = "no name";
    balance = 0;
}

Account::Account(string name, double amount) // constructor with two parameters
{
    accountID = getNextAccountNumber(); // get next account#
    accountHolder = name;
    balance = amount;
}

// Accessor (getter) and mutator (setter) functions assign
// values to or return the values of private data members.

// The depositAmount routine adds a passed amount to the
// balance only if the amount to deposit is greater than zero.

// The withdrawAmount function subtracts the absolute value
// (use the fabs() built-in function) of amount from balance.

#include "stdafx.h"
#include <iostream>
#include <iomanip>
#include <string>
#include "Account.h"
using namespace std;

int main()
{
    Account acct01;
    Account acct02("Harold M. Ferguson", 2000);
    Account acct03("Elise Janet Simmons", 3500);
    Account acct04("James Holder", 0);

    cout << endl << "Account information - initial" << endl;
    acct01.displayAccountInfo(); // show account information - initial values
    acct02.displayAccountInfo();
    acct03.displayAccountInfo();
    acct04.displayAccountInfo();

    acct01.setAccountHolder("Mary A. Tarleton");
    acct01.setBalance(542.39);
    acct02.setAccountHolder("James Ellis Holder");
    acct02.setBalance(1990.75);

    cout << endl << "Account information after changes" << endl;
    acct01.displayAccountInfo(); // show account information after changes
    acct02.displayAccountInfo();
    acct03.displayAccountInfo();
    acct04.displayAccountInfo();

    acct01.depositAmount(455); // make deposits
    acct02.depositAmount(-19.95); // negative deposit not allowed - set to zero
    acct03.depositAmount(335.27);
    acct04.depositAmount(95.63);

    cout << endl << "Account information after deposits" << endl;
    acct01.displayAccountInfo(); // show account information after deposits
    acct02.displayAccountInfo();
    acct03.displayAccountInfo();
    acct04.displayAccountInfo();

    acct01.withdrawAmount(37.39);
    acct02.withdrawAmount(-475.25); // withdrawal may be positive or negative (absolute value)
    acct03.withdrawAmount(0.25);
    acct04.withdrawAmount(50.00);

    cout << endl << "Account information after withdrawals" << endl;
    acct01.displayAccountInfo(); // show account information after withdrawals
    acct02.displayAccountInfo();
    acct03.displayAccountInfo();
    acct04.displayAccountInfo();
}
```

Account.h class header file

Open the two Account text files in C:\VCProjects\OOP and copy the contents of AccountHeader.txt to *Account.h* and AccountDriver.txt to *AccountClass.cpp*

```
C:\VCProjects\OOP\AccountClass\Debug\AccountClass.exe
Account information - initial
acct01: 100001 0.00 no name
acct02: 100002 2000.00 Harold M. Ferguson
acct03: 100003 3500.00 Elise Janet Simmons
acct04: 100004 0.00 James Holder

Account information after changes
acct01: 100001 542.39 Mary A. Tarleton
acct02: 100002 2000.00 Harold M. Ferguson
acct03: 100003 3835.27 Elise Janet Simmons
acct04: 100004 1990.75 James Ellis Holder

Account information after deposits
acct01: 100001 997.39 Mary A. Tarleton
acct02: 100002 2000.00 Harold M. Ferguson
acct03: 100003 3835.27 Elise Janet Simmons
acct04: 100004 2085.35 James Ellis Holder

Account information after withdrawals
acct01: 100001 960.00 Mary A. Tarleton
acct02: 100002 1524.75 Harold M. Ferguson
acct03: 100003 3835.02 Elise Janet Simmons
acct04: 100004 2035.35 James Ellis Holder

Press any key to continue . . .
```