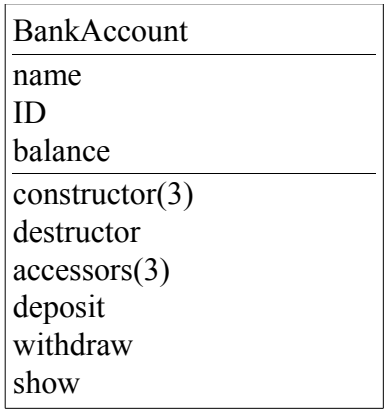
CSC240 - Sample Object Definition - C++




```
#include <iostream>
using namespace std;

// BankAccount.h

class BankAccount
{
    char *_name;
    int _ID;
    double _balance;

  public:

    BankAccount( char name[], int ID, double balance );
    ~BankAccount();

    char *getName();  // accessors
    int getID();
    double getBalance();

    void deposit( double amount );  // mutators
    int withdraw( double amount );

    void show();
};
```

```
// BankAccount.cpp

BankAccount::BankAccount( char newName[],
            int newID, double newBalance )
{
  _name = strnewcpy( newName );
  _ID = newID;
  _balance = newBalance;
}

BankAccount::~BankAccount( )
{
  delete[] _name;
}

char *BankAccount::getName( )
{
  char *nameCopy;
  nameCopy = strnewcpy( _name );
  return nameCopy;
}

int BankAccount::getID( )
{
  return _ID;
}

double BankAccount::getBalance()
{
  return _balance;
}

void BankAccount::deposit( double amount )
{
  _balance = _balance + amount;
  if ( amount > 10000 )
    cout << endl <<
    "Notify Federal Authorities..." << endl << endl;
}

int BankAccount::withdraw( double amount )
{
  int success;
  if ( amount <= _balance )
  {
    _balance = _balance - amount;
    success = 1;  // true
  }
  else
    success = 0;  // false
  return success;
}

void BankAccount::show()
{
  cout << endl;
  cout << "   name: " << _name << endl;
  cout << "     ID: " << _ID << endl;
  cout << "balance: " << _balance << endl;
  cout << endl;
}
```

```
// application.cpp

void strcpy( char dest[], char source[] );
int strlen( char str[] );
char *strnewcpy( char source[] );

int main(  )
{
  char name[81];
  int ID;
  double initialBalance, deposit, withdrawal, amount;

  BankAccount fredAccount( "Fred FlintStone", 1011, 1000 );
  BankAccount *testAccount;

  cout << "Testing Fred's account" << endl << endl;

  fredAccount.deposit( 100 );
  if ( fredAccount.withdraw( 50 ) )
    cout << "withdraw successful" << endl;
  else
    cout << "withdraw fails" << endl;
  fredAccount.show();

  cout << "Testing another Bank Account" << endl << endl;

  cout << "Enter customer's name (no spaces): ";
  cin >> name;

  cout << "Enter account ID: ";
  cin >> ID;

  cout << "Enter initial balance: ";
  cin >> initialBalance;

  testAccount = new BankAccount( name, ID, initialBalance );
  testAccount->show();


  cout << "Enter amount to deposit: ";
  cin >> deposit;
  testAccount->deposit( deposit );
  testAccount->show();

  cout << "Enter amount to withdraw: ";
  cin >> withdrawal;
  if ( testAccount->withdraw( withdrawal ) )
    cout << "withdraw successful" << endl;
  else
    cout << "withdraw fails" << endl;
  testAccount->show();

  cout << endl << "Transferring from Fred to you..." << endl << endl;

  amount = fredAccount.getBalance();
  fredAccount.withdraw( amount );
  testAccount->deposit( amount );

  fredAccount.show();
  testAccount->show();


  delete testAccount;
}
```

```
void strcpy( char dest[], char source[] )
{
  int ix;

  for ( ix = 0; source[ix] != '\0'; ++ix )
    dest[ix] = source[ix];

  dest[ix] = '\0';
}

int strlen( char str[] )
{
  int ix;
  for ( ix = 0; str[ix] != '\0'; ++ix );
  return ix;
}

char *strnewcpy( char source[] )
{
  int size;
  char *dest;
  size = strlen( source );
  dest = new char[ size + 1 ];
  strcpy( dest, source );
  return dest;
}
```

```
// BankAccount.cpp

BankAccount::BankAccount( char newName[],
            int newID, double newBalance )
{
  _name = strnewcpy( newName );
  _ID = newID;
  _balance = newBalance;
}

BankAccount::~BankAccount( )
{
  delete[] _name;
}

char *BankAccount::getName( )
{
  char *nameCopy;
  nameCopy = strnewcpy( _name );
  return nameCopy;
}

int BankAccount::getID( )
{
  return _ID;
}

double BankAccount::getBalance()
{
  return _balance;
}

void BankAccount::deposit( double amount )
{
  _balance = _balance + amount;
  if ( amount > 10000 )
    cout << endl <<
    "Notify Federal Authorities..." << endl << endl;
}

int BankAccount::withdraw( double amount )
{
  int success;
  if ( amount <= _balance )
  {
    _balance = _balance - amount;
    success = 1;  // true
  }
  else
    success = 0;  // false
  return success;
}

void BankAccount::show()
{
  cout << endl;
  cout << "   name: " << _name << endl;
  cout << "     ID: " << _ID << endl;
  cout << "balance: " << _balance << endl;
  cout << endl;
}
```