中山大學
SUN YAT-SEN UNIVERSITY

程序设计II实验

电子与信息工程学院（微电子学院）

庞志勇 高级实验师 助教：焦涵 、桂海田 博士研究生

PPT大纲

- 上次课实验总结
- 本次课的实验介绍

要做自编教材哪些实验？

第4章 面向过程编程实验

- 实验一、VC6使用与cout输出程序设计
- 实验二、数据类型、常量、变量、表达式
- 实验三、输入输出流
- 实验四、选择结构程序设计
- 实验五 循环结构程序设计
- 实验六 控制结构综合实验
- 实验七 函数实验
- 实验八 作用域、生存期及函数实验
- 实验九 数组实验
- 实验十 指针实验
- 实验十一 结构体（记录）实验

课程设计I

第5章 面向对象编程实验

- 实验一 类与对象
- 实验二 函数重载与运算符重载
- 实验三 继承与派生
- 实验四 多态性与虚函数
- 实验五 模板与STL
- 实验六 流类库与文件操作
- 实验七 异常处理

课程设计II

实验内容

- 实验内容根据理论课内容进行调整
- 每次课实验内容可以根据自己学习情况动态调整
- 课程设计可以自选其他内容

实验一 类与对象

一. 实验目的

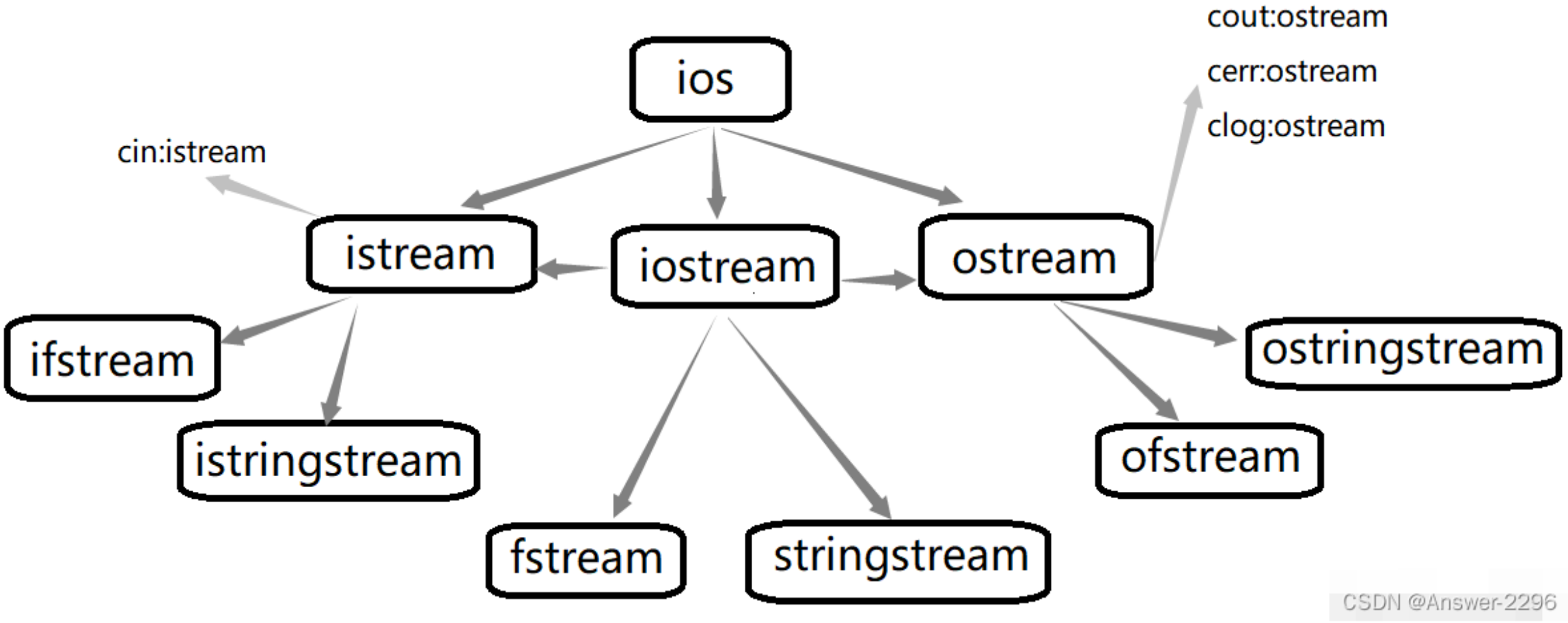
1. 理解面向对象的程序设计的特点，理解类的封装性和信息隐藏。
2. 掌握类、类数据成员、类成员函数的定义方式。
3. 理解类成员的Private、public、protect访问控制方式。
4. 掌握对象的定义创建和操作对象成员的方法。
5. 理解构造函数和析构函数的定义与使用VC++的debug调试观察执行过程。
6. 掌握重载构造函数的方法。
7. 了解拷贝构造函数的定义方法。
8. 理解掌握类的组合。

二. 实验原理

前面实验我们已经用到的“类和对象”

- Cin
- Cout
- Ifstream
- string

C++【cin/cout 输入流/输出流】深度挖掘



https://blog.csdn.net/weixin_63267854/article/details/124742641

C++【cin/cout 输入流/输出流】深度挖掘

```
class ios // 基类ios的声明部分
{
public:
    enum fmtFlags
    { // 枚举类型fmtFlags：定义用作格式标记的枚举常量
    skipws = 0x0001, // 输入时跳过空白字符（空格、制表符、回车或换行符）
    left = 0x0002, // 输出时左对齐，不足部分在右侧补填充字符
};
inline int width(int _i); // 设置下一个数据在输入/输出时的位数，内联函数
inline int eof() const; // 返回输入/输出流是否结束，内联    virtual ~ios(); // 虚析构函数
protected:
    streambuf* bp; // 流缓冲区指针
    ios( ); // 无参构造函数
private:
    ios( const ios& ); // 拷贝构造函数
    ios& operator=( const ios& ); // 重载赋值运算符
    // ..... 以下代码省略
} ;
```

https://blog.csdn.net/weixin_63267854/article/details/124742641

C++【cin/cout 输入流/输出流】深度挖掘

输入过程：计算机输入的数据先被存到内存缓存区，当按下回车键的时候，cin再从内存缓冲区读出数据，进格式转换，再赋值给相应的变量。


```
namespace std//命名空间
{
    istream cin;  //键盘对象cin
    ostream cout;//显示器对象cout
}
```

输出过程：先将输出数据存放到内存缓存区，当缓存区满的时候，再将数据输出到显示器中

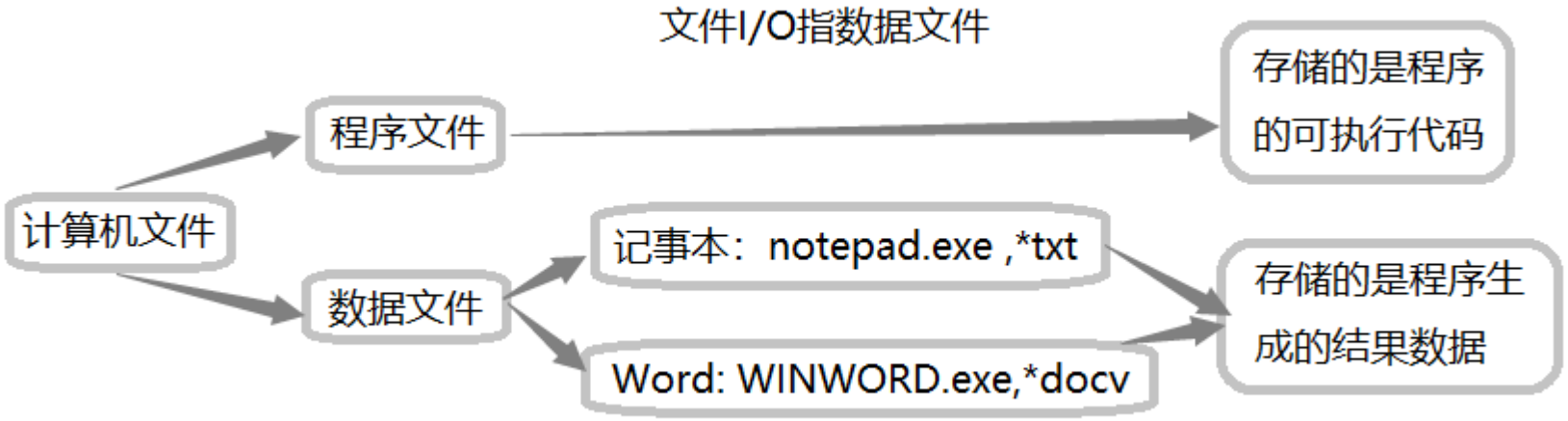
https://blog.csdn.net/weixin_63267854/article/details/124742641

C++【cin/cout 输入流/输出流】深度挖掘

```
#include<iostream>
using namespace std;
int main()
{
        const char* name[] = { "自行车","电池" };
        double price[] = { 325.5,65.3 };
        cout << "商品名称单价" << endl;
   cout.flags(ios::left);
   //格式标记设置左对齐，等价于格式操纵符cout <<
setiosflags(ios::left);
        cout.fill('#');//输出时位数不足部分补#
        for (int n = 0; n < 2; n++)
        {
                cout.width(8); //设置输出宽度
                cout << name[n];cout << " ";
                cout.width(8);  //设置输出宽度
                cout << price[n];cout << endl;
        }
}
```

```
#include <iostream>
#include <iomanip>
using namespace std;
int main( )
{
    double x = 12.345678;
    cout << x << endl; // 默认：12.3457
    cout << setiosflags( ios::fixed ); // 定点表示法
    cout << setprecision( 2 ); // 保留2位小数
    cout << x << endl; // 显示结果：12.35
    cout << resetiosflags(ios::fixed); //取消定点格式
    cout << setiosflags(ios::scientific); //科学表示法
    cout << setprecision( 8 ); // 保留8位小数
    cout << x << endl; //显示结果：1.23456780e+001
    return 0;
}
```

文件I/O




```
class ofstream : public ostream // 公有继承通用输出类ostream
{
public:
    ofstream(); // 无参构造函数
    ofstream(const char *, int =ios::out); // 有参构造函数
    ~ofstream(); // 析构函数
    void open(const char *, int =ios::out); // 打开文件
    bool is_open() const; // 检查文件是否正确打开
    void close(); // 关闭文件
    // ..... 以下代码省略
};
```

文件I/O

```
// 输出文本文件  和cout用法对应
#include <iostream>
#include <iomanip>
#include <fstream>
using namespace std;
int main()
{
    const char* name [] = { "手电筒", "电池" };
    double price[] = { 75.825, 4.1 };
    int n;
    // 使用显示器对象cout将数据输出到显示器
    cout << "商品名称 单价\n" << setiosflags(ios::left);
    for (n = 0; n < 2; n++)
    {
        cout << setw(8) << name[n]; cout << "";
        cout << setw(6) << price[n]; cout << endl;
    }
```

```
    // 使用文件输出流类ofstream的文件对象fout将数据输出到文本文件"price.txt"
    ofstream fout; // 文件输出流类ofstream对象fout(定义文件输出对象)
    fout.open("price.txt"); // 打开文件“price.txt”，如文件不存在则创建新文件
    fout << "商品名称 单价\n" << setiosflags(ios::left); // 标题行
    for (n = 0; n < 2; n++)
    {
        fout << setw(8) << name[n]; fout << "";
        fout << setw(6) << price[n]; fout << endl;
    }
    fout.close(); // 关闭所打开的文件“price.txt”
    return 0;
}
```

文件I/O

例子1文本文件输入 与cin用法对应

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
        char name[20];
        double price;
        // 使用文件输入流类ifstream的文件对象fin从文本文件“price.txt”中输入数据
        ifstream fin; // 文件输入流类ifstream对象fin需要程序员自己定义
        fin.open("price.txt"); // 打开文件“price.txt”，刚才运行创立的price.txt
        fin.getline(name, 19); // 读出标题行
        cout << name << endl; // 显示所读出的标题行，显示结果：商品名称 单价
        for (int n = 0; n < 2; n++)
        {
                fin >> name >> price; // 从文件“price.txt”中读取商品名称和单价
                cout << name << ", " << price << endl; // 显示商品名称和单价，验证输入结果
        }
        fin.close(); // 关闭所打开的文件“price.txt”
        return 0;
}
```

充分利用网络资源进行学习

下面的列表中还列出了其他一些 C++ 类和对象相关的概念，可以点击相应的链接进行学习。用生活中简单易懂的“对象”进行编程练习。

概念	描述
类成员函数	类的成员函数是指那些把定义和原型写在类定义内部的函数，就像类定义中的其他变量一样。
类访问修饰符	类成员可以被定义为 public、private 或 protected。默认情况下是定义为 private。
构造函数 & 析构函数	类的构造函数是一种特殊的函数，在创建一个新的对象时调用。类的析构函数也是一种特殊的函数，在删除所创建的对象时调用。
C++ 拷贝构造函数	拷贝构造函数，是一种特殊的构造函数，它在创建对象时，是使用同一类中之前创建的对象来初始化新创建的对象。
C++ 友元函数	**友元函数**可以访问类的 private 和 protected 成员。
C++ 内联函数	通过内联函数，编译器试图在调用函数的地方扩展函数体中的代码。
C++ 中的 this 指针	每个对象都有一个特殊的指针 **this**，它指向对象本身。
C++ 中指向类的指针	指向类的指针方式如同指向结构的指针。实际上，类可以看成是一个带有函数的结构。
C++ 类的静态成员	类的数据成员和函数成员都可以被声明为静态的。

https://www.runoob.com/cplusplus/cpp-classes-objects.html

三、 实验内容1

1.用C++编写程序：请定义一个矩形类(Rectangle)，私有数据成员为矩形的长度(len)和宽度(wid)，缺省构造函数置len和wid为0，有参构造函数置len和wid为对应形参的值，另外还包括求矩形周长、求矩形面积、取矩形长度和宽度、修改矩形长度和宽度为对应形参的值、输出矩形尺寸等公有成员函数。要求输出矩形尺寸的格式为“length：长度，width：宽度”。编写主函数对定义的类进行测试。

```
// classes example
#include <iostream>
using namespace std;

class Rectangle {
    int width, height;
  public:
    void set_values (int,int);
    int area() {return width*height;}
};

void Rectangle::set_values (int x, int y) {
  width = x;
  height = y;
}
int main () {
  Rectangle rect;
  rect.set_values (3,4);
  cout << "area: " << rect.area();
  return 0;
}
```

```
// example: class constructor
#include <iostream>
using namespace std;

class Rectangle {
    int width, height;
  public:
    Rectangle (int,int);//构造函数
    int area () {return (width*height);}
};

Rectangle::Rectangle (int a, int b) {
  width = a;
  height = b;
}

int main () {
  Rectangle rect (3,4);
  Rectangle rectb (5,6);
  cout << "rect area: " << rect.area() << endl;
  cout << "rectb area: " << rectb.area() << endl;
  return 0;
}
```

三、 实验内容2

定义一个类来表示银行帐户。包括以下成员:

数据成员: 1） 存款人姓名2）账号3）账户类型4）账户余额。

成员函数: 1） 用string作为构造函数姓名参数2）存入金额3）检查余额后提取金额4）显示姓名和余额。

编写一个主程序来测试程序。

```
Enter Details:
-----------------------
Accout No. 45789123

Name : Mangala

Account Type : Saving

Balance : 50000

Enter Deposit Amount = 5000

Enter Withdraw Amount = 10000

-----------------------
Accout No. : 45789123
Name : Mangala
Account Type : Saving
Balance : 45000
```

- UML(Unified Modeling Language) 统一建模语言，又称标准建模语言。是用来对软件密集系统进行可视化建模的一种语言。UML类图，属于UML图中的一种，是在面向对象语言用中用来表示一个类的图形

- 设计一个类Account，要将mName、mSN、mBalance均作为其成员数据，将deposit、withdraw、getBalance均作为其成员函数。类图UML设计图如右图

- 三个格子从上至下分别表示：
  - 类名（如果是接口，就使用斜体表示）
  - 类的特性/成员变量（一般是类的字段和属性,可没有这一行）
  - 类的操作/成员函数（一般是类的方法或行为）默认构造函数、带参数的构造函数、复制构造函数、析构函数

它们前边的符号有以下几类：“+”表示public，“-”表示private，“#”表示protected

银行账户程序的 Account 类的 UML 图

1.设计一个类Account，要将mName、mSN、mBalance均作为其成员数据，将deposit、withdraw、getBalance均作为其成员函数。

```
class Account
{
public:
        Account(char name[],long num,float amount); //类的有参构造函数
        Account();                                  //类的无参构造函数
        void deposit(float amount);                 //往账户中存款
        int withdraw(float amount);                 //从账户中取款
        float getBalance();                         //查询当前余额
private:
        char mName[20];                                 //银行账户的户名
        long mSN;                                       //本账户的帐号
        float mBalance;                                 //本账户当前的余额
};
```

2、代码中 strcpy（）即字符串复制函数，其原型是char * strcpy（char * dest，const char * src），把含有‘ \0 ’结束符的字符串复制到罗一个地址空间，即把从 src 地址开始且含有 NULL 结束符的字符串复制到以 dest 开始的地址空间，其中 src 和 dest 所指内存区域不可以重叠且 dest 有足够空间容纳 src 的字符串。

```
strcpy(mName, name);     //字符串复制函数
```

3、无参构造函数跟有参构造函数的区别，即在类里定义一个与类名相同的函数，但它的实参列表为空。

```
//类的有参构造函数
Account::Account(char name[], long num, float amount)
{
        strcpy(mName, name);     //字符串复制函数
        mSN = num;
        mBalance = amount;
}
//类的无参构造函数
Account::Account()
{
        cout << "无参函数被调用！"<< endl;
}
```

4、取款操作中，我们要考虑账户透支，即账户余额为0的情况，这里我们定义一个 if - else if 语句，若取款金额大于卡内金额时，return 0此时函数正常终止，而当取款金额小于或等于卡内金额时，return 1此时函数异常退出，即异常退出此函数。

```
//从当前账户中取款
int Account::withdraw(float amount)
{
        if (amount > mBalance)
        {
                return 0;
        }
        else if (amount <= mBalance)
        {
                mBalance = mBalance - amount;
                return 1;                    //return 1代表函数非正常终止
        }
}
```

5、查询账户余额操作中，return 关键字，其作用是返回程序流程的控制权，副作用是返回一个值，这里是返回卡内金额 mBalance 的值。

```
//查询当前账户的余额
float Account::getBalance()
{
        return mBalance;
}
```

https://blog.csdn.net/qq_43085848/article/details/116140649

6、Account A(name, num, amount)这里我们建立了一个账户，即实例化一个对象。

```
int main()
{
        int NO, m;
        char name[20];
        long num;
        float amount;
        cout << "请输入所开账户户名：";
        cin >> name;
        cout << "请输入所开账户帐号：";
        cin >> num;
        cout << "请输入所开账户初始存款金额：";
        cin >> amount;
        Account A(name, num, amount);
        ..........................................
        ..........................................
```

```
cout << "-------------------------------------------
"<<endl;
        cout << "                菜单栏                    "<<endl;
        cout << "1、存款请输入“1”"<<endl;
        cout << ""<<endl;
        cout << "2、取款请输入“2”"<<endl;
        cout << ""<<endl;
        cout << "3、查询账户余额请输入“3”"<<endl;
        cout << ""<<endl;
        cout << "4、退出请输入“4”"<<endl;
        cout << ""<<endl;
        cout << "-------------------------------------------
  "<<endl;
```

请根据上面菜单栏代码，实现具体菜单项功能

```
while (1)
        {
            cout << "请输入选择:" << endl;
            cin >> NO;
            switch (NO)       //通过switch循环来判断输入的菜单栏选择对应其相应的操作
            {
            case 1:
                    cout << "请输入存款金额：";
                    cin >> amount;
                    A.deposit(amount);
                    break;                                  //表示跳出该switch语句体
            case 2:
                    cout << "请输入取款金额：";
                    cin >> amount;
                    m = A.withdraw(amount);
                    if (m == 0)
                            cout << "当前账户余额不足！" << endl;
                    else
                            cout << "取款成功！" << endl;
                    break;
            case 3:
                    cout << "当前账户余额为：" << A.getBalance() << endl;
                    break;
```

```
                case 4:
                        cout << "账户已退出！" <<endl;
                        return 0;
                default:
                        cout << "输入错误！"<< endl;    //判断输入菜单
栏是否输入正确
                        exit(0);
                }
                cout << "" <<endl;
        }
}
```

多读多写

- ✓ 同一道程序设计题目，自己设计好之后，多在网上书上找其他人的代码，进行优缺点比较，并且在自己PC上调试运行体验
- ✓ 如果PPT上已经给了程序实现代码，尝试着进行改进。。。

注意前面理论知识综合运用：

1）数据类型

2）数组

3）指针

4）函数声明

5）类

6）构造函数

7）strcpy字符串拷贝函数

```
#include<iostream>
#include<stdio.h>
#include<string.h>

using namespace std;

class bank
{
        int acno;
        char nm[100], acctype[100];
        float bal;
    public:
        bank(int acc_no, char *name, char *acc_type, float balance)  //Parameterized Constructor
        {
                acno=acc_no;
                strcpy(nm, name);
                strcpy(acctype, acc_type);
                bal=balance;
        }
        void deposit();
        void withdraw();
        void display();
};
```

注意理论知识综合运用：

1）类外定义成员函数要加上类前缀

2）友好的交互提示信息

3）联系现实生活实际存钱取钱，进行数学抽象，本例只是简单加减法而已

```
void bank::deposit()   //depositing存款 an amount
{
        int damt1;
        cout<<"\n Enter Deposit Amount = ";
        cin>>damt1;
        bal+=damt1;
}
void bank::withdraw()  //withdrawing取款 an amount
{
        int wamt1;
        cout<<"\n Enter Withdraw Amount = ";
        cin>>wamt1;
        if(wamt1>bal)
                cout<<"\n Cannot Withdraw Amount";
        bal-=wamt1;
}
void bank::display()  //displaying the details
{
        cout<<"\n ----------------------";
        cout<<"\n Accout No. : "<<acno;
        cout<<"\n Name : "<<nm;
        cout<<"\n Account Type : "<<acctype;
        cout<<"\n Balance : "<<bal;
}
```

注意前面理论知识综合运用：

1）变量定义

2）数据输入、输出

3）变量赋值

4）类 相当于数据类型如int等

5）对象 相当于变量

6）对象数据成员函数的访问

体会：在软件开发时，面向对象程序设计，根据设计的需要对现实世界的事物进行抽象，产生类。使用这样的方法解决问题，接近于日常生活和自然的思考方式，势必提高软件开发的效率和质量。

```
int main()
{
        int acc_no;
        char name[100], acc_type[100];
        float balance;
        cout<<"\n Enter Details: \n";
        cout<<"----------------------";
        cout<<"\n Accout No. ";
        cin>>acc_no;
        cout<<"\n Name : ";
        cin>>name;
        cout<<"\n Account Type : ";
        cin>>acc_type;
        cout<<"\n Balance : ";
        cin>>balance;

        bank b1(acc_no, name, acc_type, balance);  //object is
created
        b1.deposit(); //
        b1.withdraw(); // calling member functions
        b1.display(); //
        return 0;
}
```

测试结果

```
Enter Details:
----------------------
Accout No. 45789123

Name : Mangala

Account Type : Saving

Balance : 50000

Enter Deposit Amount = 5000

Enter Withdraw Amount = 10000

----------------------
Accout No. : 45789123
Name : Mangala
Account Type : Saving
Balance : 45000
```

思考：还有哪些改进？

- ✓ 增加利率功能
- ✓ 增加定期存款、活期存款功能

学习的四重境界

知学、好学、会学、乐学

祝大家实验顺利！