

# C++程序设计实验报告



姓名	薛旗	学号	20155362
班级	软信-1503	指导教师	马毅
实验名称	C++程序设计		
开设学期	大二第一学期		
实验时间	第14周——第17周		
评定成绩	评定人签字		
	评定日期		

东北大学软件学院

2016年

## 一、实验目的

### 实验一：C++程序设计基础

- (1) 学习编写简单的 C++ 程序，并掌握 C++ 程序的基本格式与规范。
- (2) 理解 C++ 程序结构的特点。
- (3) 掌握函数的定义和调用方法。
- (4) 练习重载函数的使用。
- (5) 练习函数模板的使用。
- (6) 练习使用系统函数。
- (7) 学习使用 VC++ 的 debug 调试功能，使用 step into 追踪到函数内部。

### 实验二：类与对象

- (1) 学习类与对象的定义。
- (2) 掌握类与对象的使用方法。
- (3) 学习类和对象的声明。
- (4) 学习具有不同属性的数据成员与成员函数的访问方式。
- (5) 理解构造函数和析构函数的定义与执行过程。
- (6) 学会类的聚集使用方法。

### 实验三：多态性

- (1) 掌握运算符重载的方法。
- (2) 学习使用虚函数实现动态多态性。

### 实验四：文件

- (1) 熟悉流类库中常用的类及其成员函数的用法。
- (2) 学习标准输入输出及格式控制。
- (3) 学习对文件的应用方法（二进制文件、文本文件）。

## 二、实验内容与实验步骤

### 实验一：

- (1) 输入并运行所给的参考程序 1，并将程序中的注释部分也输入计算机，体会和理解程序的基本格式规范。
- (2) 编写一个函数把华氏温度转换为摄氏温度，转换公式为  $C = (F - 32) * 5/9$ 。
- (3) 编写重载函数 Max1 可分别求取两个整数，三个整数，两个双精度数，三个双精度数的最大值。
- (4) 使用重载函数模板重新实现上小题中的函数 Max1。
- (5) 使用系统函数 pow(x,y) 计算  $x^y$  的值，注意包含头文件 math.h。

(6) 用递归的方法编写函数求 Fibonacci 级数，观察递归调用的过程。

## 实验二：

(1) 定义一个 CPU 类，包含等级(rank)、频率(frequency)、电压(voltage)等属性，有两个公有成员函数 run、stop，其中 rank 为枚举类型 CPU\_Rank，定义为 enum CPU\_Rank(P1 = 1, P2, P3, P4, P5, P6, P7)，frequency 为单位是 MHz 的整型数，voltage 为浮点型的电压值。观察构造函数和析构函数的调用顺序。

(2) 定义一个简单的 Computer 类，有数据成员芯片(CPU)，内存(ram)，光驱(cdrom)等等，有两个公有成员函数 run、stop，CPU 为 CPU 类的一个对象，ram 为 RAM 类的一个对象，cdrom 为 CDROM 类的一个对象，定义并实现这个类。

(3) 设计一个用于人事管理的 People（人员）类。考虑到通用性，这里只抽象出所有类型人员都具有的属性：number（编号），sex（性别），birthday（出生日期），id（身份证号）等等。其中“出生日期”定义为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数。

## 实验三：

(1) 定义 Point 类，有坐标 \_x, \_y 两个成员变量；对 Point 类重载“++”（自增）、“--”（自减）运算符，实现对坐标值的改变。

(2) 定义一个车(vehicle)基类，有 Run、Stop 等成员函数，由此派生出自行车(bicycle)类、汽车(motorcar)类，从 bicycle 和 motorcar 派生出摩托车(motorcycle)类，它们都有 Run、Stop 等成员函数。观察虚函数的作用。

(3) 对实验二中的 people 类重载“==”运算符和“=”运算符，“==”运算符判断两个 people 类对象的 id 属性是否相等；“=”运算符实现 people 类对象的赋值操作。

## 实验四：

(1) 编写程序，用二进制方式读写指定的文件 a.bin，在每一行前加行号，写入另一个二进制文件 b.bin 中，从 b.bin 中读取数据输出到屏幕。

(2) 使用实验二中的人员类数组，输入数据并显示出来，重载“<<”运算符把此数组的内容写入磁盘文件，再重载“>>”运算符显示出文件内容。

(3) 设计一个留言的类，实现以下功能：

a. 程序第一次运行时，建立一个名为 C:\msg. ini 的文本文件，并把用户输入的信息存入该文件。

b. 以后每次运行时，都先读取该文件的内容并显示给用户，然后由用户输入新的信息，退出时将新的信息存入这个文档。文件的内容，既可以是最新的信息，也可以包括以前所有的信息，请自己选择。

### 三、实验过程与分析

#### (1)核心代码:

##### 实验一:

##### Lab1\_1: 华氏温度转换为摄氏温度

```
float Convert(float TempFer)
{
    float C;
    float F = TempFer;
    C = (F - 32) * 5 / 9;
    return C;
}
```

##### Lab1\_2 : 重载函数 Max1

```
int Max1(int x, int y);
double Max1(double x, double y);
int Max1(int x, int y, int z);
double Max1(double x, double y, double z);
```

##### lab1\_3 使用重载函数模板实现函数 Max1

```
template<typename T>
T max1(T x, T y) { return (x > y) ? x : y; }
template<typename T>
T max1(T x, T y, T z)
{
    return (x > y ? (x > z ? x : z) : (y > z ? y : z));
}
```

##### Lab1\_5 用递归的方法编写函数求 Fibonacci 级数

```
int fib(int n)
{
    int x;
    if (n <= 2)
        return 1;
    if (n > 2)
        x = fib(n - 1) + fib(n - 2);
    return x;
}
```

## 实验二：

### Lab2\_1: 定义 CPU 类

```
enum CPU_Rank { P1 = 1, P2, P3, P4, P5, P6, P7 };
class CPU
{
private:
    CPU_Rank rank;
    int frequency;
    float voltage;
public:
    CPU()
    {
        cout << "*构造CPU*\n" << endl;
        rank = P2;
        frequency = 66666;
        voltage = 6;
    }
    ~CPU()
    {
        cout << "*析构CPU*" << endl;
    }
    void run()
    {
        cout << "#CPU开始运行#" << endl;
        cout << "CPU等级为: " << rank << endl;
        cout << "CPU频率为: " << frequency << " MHz" << endl;
        cout << "CPU电压为: " << voltage << " V" << endl;
    }
    void stop()
    {
        cout << "#CPU停止运行#" << endl;
    }
};
```

### Lab\_2 定义 Computer 类

```
class Computer
{
private:
    CPU cpu;
    RAM ram;
    CDROM cdrom;
public:
```

```

Computer()
{
    cout << "*构造计算机*\n" << endl;
}
~Computer()
{
    cout << "\n*析构计算机" << endl;
}
void run()
{
    cout << "#计算机开始运行#\n" << endl;
    cpu.run();
    ram.run();
    cdrom.run();
}
void stop()
{
    cdrom.stop();
    ram.stop();
    cpu.stop();
    cout << "\n#计算机停止运行#" << endl;
}
};

```

### Lab2\_3 人士管理 Person 类

```

class Person
{
private:
    int num;
    char sex;
    Date birthday;
    int id;
public:
    Person() {}
    Person(int n, int y, int m, int d, int a, char s = 'm') :birthday(y, m, d)
    {
        num = n;
        sex = s;
        id = a;
    }
    Person(Person& p)
    {
        num = p.num;
        sex = p.sex;
    }
};

```

```

        birthday = p.birthday;
        id = p.id;
    }
    void input()
    {
        cout << "编号: ";
        cin >> num;
        cout << "性别(m/f): ";
        cin >> sex;
        cout << "生日: ";
        birthday.set();
        cout << "身份证号: ";
        cin >> id;
        cout << endl;
    }
    void output()
    {
        cout << "编号: " << num << endl;
        cout << "性别: " << sex << endl;
        cout << "身份证号: " << id << endl;
        cout << "生日: ";
        birthday.display();
        cout << endl;
    }
    ~Person() {}
};

```

### 实验三:

#### Lab5\_1 重载运算符

```

Point Point::operator ++(int)
{
    x += 1;
    y += 1;
    return *this;
}
Point &Point::operator ++()
{
    x += 1;
    y += 1;
    return *this;
}
Point Point::operator --(int)

```

```

{
    x -= 1;
    y -= 1;
    return *this;
}

Point &Point::operator --()
{
    x -= 1;
    y -= 1;
    return *this;
}

```

### Lab5\_2 成员函数调用

```

class Vehicle
{
public:
    virtual void Run() { cout << "基类 Vehicle::Run 函数调用\n"; }
    virtual void Stop() { cout << "基类 Vehicle::Stop 函数调用\n"; }
};

class motorcar :public Vehicle {
public:
    void Run() { cout << "派生类 motorcar::Run 函数调用\n"; }
    virtual void Stop() { cout << "motorcar::Stop 函数调用\n"; }
};

class bicycle :public Vehicle {
public:
    virtual void Run() { cout << "派生类 bicycle::Run 函数调用\n"; }
    void Stop() { cout << "派生类 bicycle::Stop 函数调用\n"; }
};

class motorcycle :public bicycle, public motorcar {
public:
    void Run() { cout << "派生类 motorcycle::Run 函数调用\n"; }
    void Stop() { cout << "派生类 mototrcycle::Stop 函数调用\n"; }
};

```

### Lab5\_3 "==" 和 "=" 运算符重载

```

bool Person::operator==(Person &p)
{
    return id == p.id;
}

Person & Person::operator = (Person &p)

```



```

{
    num = p.num;
    sex = p.sex;
    birthday = p.birthday;
    id = p.id;
    return *this;
}

```

## 实验四：

### Baseform1:二进制文件读写

```

ifstream fin;
ofstream fout;
int num_line = 1;
char temp, inFileName[128];
string buffer;
if (argc == 1)
{
    printf("请用格式:SetLineNum filename\n");
    exit(0);
}
else
{
    strcpy_s(inFileName, argv[1]);
}
cout << "正打开输入文件...\n";
fin.open(inFileName);
if (fin.fail())
{
    cout << "输入文件打开失败!\n";
    exit(1);
}
cout << "正打开输出文件...\n";
fout.open("output.txt");
if (fout.fail())
{
    cout << "输出文件打开失败!\n";
    exit(1);
}

```

### Baseform2: " <<" 和 ">>" 运算符重载

```

ostream& operator<<(ostream& os, People& p)
{

```

```

char s[100];
ifstream Output("abc.txt");
os << endl << "人员信息: " << endl;
while (!Output.eof())
{
    Output.getline(s, sizeof(s));
    os << s << endl;
}
return os;
}
istream& operator >> (istream& is, People& p)
{
    ofstream Input("abc.txt", ios::app);
    cout << "请输入人员信息: " << endl;
    cout << "编号: ";
    is >> p.number;
    Input << "编号: " << p.number << endl;
    cout << "性别: ";
    is >> p.sex;
    Input << "性别: " << p.sex << endl;
    cout << "出生日期: ";
    p.birthday.shuru();
    Input << "出生日期: " << p.birthday.year << "." << p.birthday.month << "." <<
p.birthday.day << endl;
    cout << "身份证号: ";
    is >> p.id;
    Input << "身份证号: " << p.id << endl;
    return is;
}

```

### Baseform3:留言类

```

class MailMan
{
public:
    bool bHaveMsg;
    void OutMsg();
    void ReadMsgFromFile();
    void SaveMsgToFile();
    void InMsg();
    MailMan();
    virtual ~MailMan();
private:
    char Buf[1024];
    fstream fs;
}

```

};

## (2) 故障分析:

a. 在进行实验一时, 输入参考程 1 在编译时提示报错, 经过试验后发现, 将语句 `#include <iostream. h>` 改为 `#include <iostream>` 即可正常运行

b. 在实验 lab1\_3 时, 使用函数模板实现求任意类型数的最大值出错, 后来发现自己只声明了一次 `template<typename T>`。通过学习, 知道在每一次使用函数模板之前, 都要声明一下 `template<typename T>`

c. 用自定义的 .h 文件时, 正确格式应为 `#include "test.h"`, 而不是 `#include<test.h>`

## (3) 测试数据:

实验一:

www.flagzue.cn

```
C:\WINDOWS\system32\cmd.exe

-----
*两个数求和*

Enter two numbers:
12 23
The sum is: 35
-----

*华氏温度转换为摄氏温度*

请输入华氏温度: 60
摄氏温度为: 15.5556
-----

*重载函数Max1*

1. 求两个整数的最大值, 请输入两个整数: 4 6
   最大值为: 6
2. 求三个整数的最大值, 请输入三个整数: 2 5 98
   最大值为: 98
3. 求两个双精度数的最大值, 请输入两个双精度数: 1.23 6.23
   最大值为: 6.23
4. 求三个双精度数的最大值, 请输入三个双精度数: 1.2 4.6 3.11
   最大值为: 4.6
-----

*使用重载函数模板实现函数Max1*

1. 求两个整数的最大值, 请输入两个整数: 78 53
   最大值为: 78
2. 求三个整数的最大值, 请输入三个整数: 1 3 56
   最大值为: 56
3. 求两个双精度数的最大值, 请输入两个双精度数: 1.23 5.11
   最大值为: 5.11
4. 求三个双精度数的最大值, 请输入三个双精度数: 2.3 5.4 7.8
   最大值为: 7.8
-----

*使用系统函数pow(x, y)计算x的y次方的值*

请输入两个整数: 2 3
2的3次方为8
-----

*用递归的方法编写函数求Fibonacci级数*

请输入n: 6
所求的Fibonacci级数fib(n)=8
请按任意键继续. . .
```

实验二:

```
C:\WINDOWS\system32\cmd.exe

*定义CPU类，观察构造函数和析构函数的调用顺序*

*构造CPU*

#CPU开始运行#
CPU等级为：2
CPU频率为：66666 MHz
CPU电压为：6 V
#CPU停止运行#
*析构CPU*

-----

*定义Computer类，观察构造函数和析构函数的调用顺序*

*构造CPU*

*构造RAM*

*构造CDROM*

*构造计算机*

#计算机开始运行#

#CPU开始运行#
CPU等级为：2
CPU频率为：66666 MHz
CPU电压为：6 V
#RAM开始运行#
#CDROM开始运行#

#CDROM停止运行#
#RAM停止运行#
#CPU停止运行#

#计算机停止运行#

*析构计算机

*析构CDROM

*析构RAM

*析构CPU*

-----

*设计用于人事管理的Person类*

编号：20150001
性别(m/f)：f
生日：1996 01 01
身份证号：110001199601012345

编号：20150001
性别：f
身份证号：-858993460
生日：1996年1月1日
请按任意键继续. . .
```

实验三：

```
C:\WINDOWS\system32\cmd.exe

-----
*定义Point类，重载运算符，实现坐标值的改变*

x: 3 y: 4
x: 3 y: 4
x: 4 y: 5
x: 3 y: 4
x: 3 y: 4
x: 2 y: 3

-----
*定义vehicle类，通过不同方式调用成员函数*

成员引用方式访问：
基类 Vehicle::Run 函数调用
基类 Vehicle::Stop 函数调用
派生类 bicycle::Run 函数调用
派生类 bicycle::Stop 函数调用
派生类 motorcar::Run 函数调用
motorcar::Stop 函数调用
派生类 motorcycle::Run 函数调用
派生类 mototrcycle::Stop 函数调用

对象指针方式访问：
基类 Vehicle::Run 函数调用
基类 Vehicle::Stop 函数调用
派生类 bicycle::Run 函数调用
派生类 bicycle::Stop 函数调用
派生类 motorcar::Run 函数调用
motorcar::Stop 函数调用
基类 Vehicle::Run 函数调用
基类 Vehicle::Stop 函数调用

-----
*对people类重载'=='运算符和'='运算符*

编号: 1001
性别(m/f): m
生日: 1996 10 01
身份证号: 14260123512365

1
编号: 1001
性别: m
身份证号: -858993460
生日: 1996年10月1日
请按任意键继续. . .
```

实验四:

```
C:\WINDOWS\system32\cmd.exe
请输入人员信息:
编号: 1001
性别: m
出生日期: 1996 10 10
身份证号: 14235462134562112

人员信息:
编号: 1001
性别: m
出生日期: 1996.10.10
身份证号: 14235462134562112

请按任意键继续. . .
```

```
C:\WINDOWS\system32\cmd.exe
留言系统工作中...

-----无信息-----

请输入您要留下的信息, 以“~”结束(Shift + `):
老师您好! 嘻嘻嘻嘻嘻嘻^_^~

信息已写入文件.
留言系统结束运行...
请按任意键继续. . .
```

## 四、实验结果总结

### (1) 实验思考题:

1. Visual C++ 6.0 环境下是如何管理项目和文件的?  
通过创建项目, 进而在该项目中创建一个或多个文件。
2. C++提供的输入输出操作与 C 语言的输入输出操作的异同点?  
C 语言中 scanf 和 printf 不是类型安全且不具有扩展性; 而 c++是类型安全且有扩展性; C 语言中要把读写变量和控制读写信息分开, 而 c++不需要。
3. C++引入的内联函数与其它函数有何不同?  
内联函数应该放在头文件中定义, 在函数返回类型前加上 inline 关键字。
4. 重载函数时通过什么来区分?  
根据原方法名、各个参数的数据类型、顺序和数量生成新的方法名, 与其他因素无关。
5. 类和结构体的区别是什么?  
结构体中的数据默认方式是 public 的, 而类是 private 的结构体不提供

继承机制，类提供继承机制，实现代码复用类可以实现多态，结构体不支持。

6. 类的成员函数与普通的函数有何区别？

普通函数可以随便调用，并且无法继承和封装，成员函数根据类不同，可以继承和封装。

7. 构造函数和析构函数的作用是什么？

构造函数用于新对象的初始化工作；析构函数用于撤销对象前的清理工作。

8. 类的访问控制权限有何意义？

以实现数据和方法的公有和私密性。

9. 什么是封装？类是如何进行封装的？

封装就是将抽象得到的数据和行为（或功能）相结合，形成一个有机的整体，也就是将数据与操作数据的源代码进行有机的结合，形成“类”，其中数据和函数都是类的成员。

10. 多态性是如何实现的？

多态可以通过模板和函数重载来实现。

11. 虚析构函数的作用是什么？如何发挥作用？

为了正确析构基类指针指向的派生类对象。

12. 运算符重载和函数重载有何异同点？

从意义上说，都是定义了一种新的成员函数；从作用上说，运算符重载是赋予运算符新的含义。

13. 什么是流？使用流有那些好处？

流是 c++ 标准库类的一个类，包括常用的输入输出流。可以简化程序员对电脑的一些操作。

14. C++ 为什么要有自己的输入输出系统？

因为 c++ 是一种面向对象的语言，因此将 I/O 设计成一个类便于用户操作。

15. 随机文件与顺序文件有什么区别？在随机文件中如何确定文件指针的位置？

顺序文件从头到尾进行检索，随机文件可以直接按照数据偏移量定位。

16. C++ 与定义了那些流？他们分别与什么设备相连？

输入输出流连接控制台；文件流连接文件；字符串流连接字符串。

## (2) 心得体会：

通过 C++ 实验课，我理解了 C++ 面向对象编程语言与 C 面向过程编程语言的不同，知道了有关类的知识，掌握了函数的定义和调用方法，学会了重载函数的使用及模板函数的使用。通过练习，我熟悉并掌握了如何利用虚函数实现程序的多态性，学会了如何对各类运算符进行重载，使得各类运算符满足同类之间的运算，使程序更加高效。我认为上机实验是学习程序设计语言必不可少的环节，通过实践，我加深了对知识的理解，获益匪浅。



**教师评语或评价表格：**（任课教师可根据实际情况，做适当调整）

评语及评价表格的字体颜色为红色

评价表格示例：（考核标准与教学大纲中的实验考核标准一致）

考核标准	得分
(1) 正确理解和掌握实验所涉及的概念和原理（10%）；	
(2) 按实验要求合理设计数据结构和程序结构（20%）；	
(3) 能设计测试用例，运行结果正确（20%）；	
(4) 认真记录实验数据，原理及实验结果分析准确（20%）；	
(5) 实验过程中，具有严谨的学习态度和认真、踏实、一丝不苟的科学作风（10%）；	
(6) 所做实验具有一定的创新性（10%）；	
(7) 实验报告规范（10%）。	

www.flagzhe.cn