南京邮电大学

实 验 报 告

（ **2023 / 2024 学年 第 二 学期**）

课程名称	面向对象程序设计及 C++
实验名称	实验一： 类和对象的定义及使用
实验时间	2024 年 4 月 7 日
指导单位	计算机学院、软件学院、网络空间安全学院
指导教师	吴家皋

学生姓名	于明宏	班级学号	B23041011
学院(系)	计软网安学院	专 业	信息安全

实 验 报 告

实验名称	类和对象的定义及使用		
实验类型	验证	实验学时	2

一、 实验目的和要求

（1）掌握类与对象的定义与使用方法，理解面向对象方法中通过对象间传递消息的工作机制。

（2）正确掌握类的不同属性成员的使用方法。

（3）掌握构造函数与析构函数的概念，理解构造函数与析构函数的执行过程。

（4）掌握友元函数和友元类的定义和使用。

（5）基本掌握指针和引用作为函数参数的应用。

二、实验环境(实验设备)

硬件: 微型计算机

软件: Windows 操作系统、Microsoft Visual Studio 2010

三、实验原理及内容

实验题目 1 定义一个借书证类 BookCard，在该类定义中包括如下内容。

（1）私有数据成员：

string id; //借书证学生的学号

string stuName; //借书证学生的姓名

int number; //所借书的数量

（2）公有成员函数：

构造函数 //用来初始化 3 个数据成员，是否带默认参数值参考结果来分析

void display() //显示借书证的 3 个数据成员的信息

bool borrow() //已借书数量不足 10 则将数量加 1，数量达到 10 则直接返回 false

主函数及 f()函数代码如下。请结合输出结果完成程序。

```
void f(BookCard &bk)
{
    if (!bk.borrow())
    {
       bk.display();
       cout<<"you have borrowed 10 books,can not borrow any more!"<<endl;
    }
else
    bk.display();
```

```
}
int main()
{
    BookCard bk1("B20190620","东平",10),bk2;
    f(bk1);
    f(bk2);
    return 0;
}
```

程序的运行结果为：

```
B20190620 东平 10
you have borrowed 10 books,can not borrow any more!
B19010250 雪峰 4
```

参考实验教材中相应的实验指导，完成源程序代码如下：（中文五号宋体，英文五号 Consolas 字体，单倍行距)

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

class BookCard {
private:
    string id;
    string stuName;
    int number;

public:
    BookCard(string id = "", string stuName = "", int number = 0) {
        this->id = id;
        this->stuName = stuName;
        this->number = number;
    }

    void display() {
        cout << id << " " << stuName << " " << number << endl;
    }

    bool borrow() {
```

```
        if (number < 10) {
            number++;
            return true;
        }
        else {
            return false;
        }
    }
};

void f(BookCard& bk) {
    if (!bk.borrow()) {
        bk.display();
        cout << "you have borrowed 10 books, can not borrow any more!" << endl;
    }
    else {
        bk.display();
    }
}

int main() {
    BookCard bk1("B20190620", "东平", 10), bk2("B19010250", "雪峰", 4);
    f(bk1);
    f(bk2);
    return 0;
}
```

实验题目 2 定义一个时间类 Time，有三个私有成员变量 Hour、Minute、Second，定义构造函数、析构函数以及用于改变、获取、输出时间信息的公有函数，主函数中定义时间对象，并通过调用各种成员函数完成时间的设定、改变、获取、输出等功能。

① 按要求完成类的定义与实现。

② 修改数据成员的访问方式，观察编译结果。

③ 在 Time 类中定义一个成员函数，用于实现时间增加一秒的功能，主函数中通过对象调用该函数，并输出增加一秒后的时间信息。

实 验 报 告

④ 定义一个普通函数。

```
void f(Time    t)
{       t. PrintTime( );
}
```

在 Time 类中增加拷贝构造函数的定义，主函数中调用该函数，运用调试工具跟踪，分析整个程序调用构造函数（包括拷贝构造函数）和析构函数的次数；再将 f 函数的形式参数分别修改为引用参数和指针参数（此时函数代码修改为{t-> PrintTime( );},主函数中调用，再分析此时调用构造函数和析构函数的次数。

参考实验教材中相应的实验指导完成程序，并回答相关问题。完成后的源程序代码如下：(中文五号宋体，英文五号 Consolas 字体，单倍行距)

```
#include<iostream>
using namespace std;
class Time
{
private:
    int Hour, Minute, Second;
public:
    Time(int h = 0, int m = 0, int s = 0);
    Time(const Time &ob);
    ~Time();
    void ChangeTime(int h, int m, int s);
    int GetHour();
    int GetMinute();
    int GetSecond();
    void PrintTime();
};

Time::Time(int h, int m, int s)
{
    Hour = h;
    Minute = m;
    Second = s;
    cout << "Constructing..." << endl;
```

```
}
Time::Time(const Time& ob)
{
    cout << "Copy constructing..." << endl;
    Hour = ob.Hour;
    Minute = ob.Minute;
    Second = ob.Second;
}
Time::~Time()
{
    cout << "Destructing..." << endl;
}
void Time::ChangeTime(int h, int m, int s)
{
    Hour = h;
    Minute = m;
    Second = s;
}
int Time::GetHour()
{
    return Hour;
}
int Time::GetMinute()
{
    return Minute;
}
int Time::GetSecond()
{
    return Second;
}
void Time::PrintTime()
{
    cout << Hour << ":" << Minute << ":" << Second << endl;
}
```

```
int main()
{
    Time t1;
    t1.PrintTime();
    Time t2(11);
    t2.PrintTime();
    Time t3(11, 45);
    t3.PrintTime();
    Time t4(11, 45, 14);
    t4.PrintTime();

    t1.ChangeTime(23, 59, 59);
    cout << "ChangeTime " << t1.GetHour() << ":" << t1.GetMinute() << ":" <<
t1.GetSecond() << endl;
    t4.ChangeTime(14, 45, 11);
    cout << "ChangeTime " << t4.GetHour() << ":" << t4.GetMinute() << ":" <<
t4.GetSecond() << endl;
}
```

程序的运行结果是：

```
Constructing...
0:0:0
Constructing...
11:0:0
Constructing...
11:45:0
Constructing...
11:45:14
ChangeTime 23:59:59
ChangeTime 14:45:11
Destructing...
Destructing...
Destructing...
Destructing...
```

实 验 报 告

构造函数与析构函数的调用方式及执行顺序是：

调用方式：自动调用。

执行顺序：先执行构造函数，程序结束时执行析构函数；析构函数的调用顺序与构造函数相反。

③取消类中成员函数 IncreaceOneSecond( )的注释标志，将该函数补充完整，注意时间在增加一秒情况下的进位关系。

该函数的代码如下：（中文五号宋体，英文五号 Consolas 字体，单倍行距)

```
void Time::IncreaseOneSecond() {
    if (Second < 59)
        Second++;
    else {
        Second = 0;
        if (Minute < 59)
            Minute++;
        else {
            Minute = 0;
            if (Hour < 23)
                Hour++;
            else
                Hour = 0;
        }
    }
}
```

④ 主函数中定义一个 Time 类对象并调用一次 f 函数，观察结果填写下表：

f 函数的原型	主函数中调用 f 的语句	构造函数调用次数	拷贝构造函数调用次数	析构函数调用次数
void f(Time t);	f(t1);	1	1	2
void f(Time &t);	f(t1);	1	0	1
void f(Time *t);	f(&t1);	0	0	1

通过以上结果，关于对象作形式参数、对象引用作形式参数、对象指针作形式参数时构造函数、析构函数的调用次数及顺序，你得到什么结论？

对象作为形式参数调用时，实参的值传给形参，要调用复制构造函数，且形参占内存空间，析构函数调用两次；对象引用作为形式参数调用时，相当于是实参的别名，就是对实参对象进行操作，形参不占内存空间，也不需要调用拷贝构造函数；对象指针作形式参数调用时，不调用

拷贝构造函数，通过指针可以访问实参对象的值，且未再次调用构造函数。

实验题目 3 定义一个 Girl 类和一个 Boy 类，这两个类中都有表示姓名、年龄的私有成员变量，都要定义构造函数、析构函数、输出成员变量信息的公有成员函数。

①根据要求定义相应的类。

②将 Girl 类作为 Boy 类的友元类，在 Girl 类的成员函数 VisitBoy(Boy &)中访问 Boy 类的私有成员，观察程序运行结果。

③在 Boy 类的某成员函数 VisitGirl(Girl &)中试图访问 Girl 类的私有成员，观察编译器给出的错误信息，理解原因。

④主函数中正确定义两个类的对象，调用各自的成员函数实现相应功能。

⑤再将 Boy 类作为 Girl 类的友元类，在 Boy 类的某成员函数 VisitGirl(Girl &)中访问 Girl 类的私有成员，观察编译器给出的信息。

⑥删除两个类中的函数 VisitGirl(Girl &) ，VisitBoy(Boy &)，定义一个顶层函数 VisitBoyGirl(Boy &, Girl &)，作为以上两个类的友元，通过调用该函数输出男孩和女孩的信息。

实验解答：

①定义相应的类，主函数中定义相应的类成员，调用各类的输出函数显示信息。

源程序代码如下：

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

class Boy;
class Girl
{
private:
    string Name;
    int Age;
public:
    Girl(string N = "ABC", int A = 18);
    void Output();
};

class Boy
{
```

```
private:
    string Name;
    int Age;
public:
    Boy(string N = "ABC", int A = 18);
    void Output();
};

Girl::Girl(string N, int A)
{
    Name = N;
    Age = A;
}
void Girl::Output()
{
    cout << "Girl's name: " << Name << endl;
    cout << "Girl's age: " << Age << endl;
}
Boy::Boy(string N, int A)
{
    Name = N;
    Age = A;
}
void Boy::Output()
{
    cout << "Boy's name: " << Name << endl;
    cout << "Boy's age: " << Age << endl;
}

int main()
{
    Girl g("yuxiaojing", 22);
    Boy b("yuminghong", 19);
    g.Output();
```

```
    b.Output();
}
```

程序的运行结果是：

```
Girl's name: yuxiaojing
Girl's age: 22
Boy's name: yuminghong
Boy's age: 19
```

②将 Girl 类作为 Boy 类的友元类， 写出 Girl 类的成员函数 VisitBoy(Boy &)的实现代码。

```
void Girl::VisitBoy(Boy& boy)
{
    cout << "Boy's name:" << boy.Name << endl;
    cout << "Boy's age:" << boy.Age << endl;
}
```

程序的运行结果是：

```
Girl's name: yuxiaojing
Girl's age: 22
Boy's name: yuminghong
Boy's age: 19
Boy's name: yuminghong
Boy's age: 19
```

③在 Boy 类的某成员函数 VisitGirl(Girl &)中试图访问 Girl 类的私有成员，记录编译器给出的错误信息，与②对比，你能得出友元的什么特性？

友元关系是单向的，不具有交换性。

④在上面代码的基础上，在 Girl 类的定义中，增加一行代码：friend Boy; 在主函数中通过 **Boy 类对象. VisitGirl(Girl 类对象)** 的形式输出 Girl 类对象的信息。编译的结果是什么？写出这一步你的主函数代码，要求分别用友元函数 **Girl 类对象. VisitBoy(Boy 类对象);**和 **Boy 类对象. VisitGirl(Girl 类对象) ;**和输出两个类对象的信息。

```
int main() {
    Girl g("yuxiaojing", 22);
    Boy b("yuminghong", 19);
    g.VisitBoy(b);
    b.VisitGirl(g);
    b.Output();
```

```
    g.Output();
    return 0;
}
```

⑤定义一个顶层函数 void VisitBoyGirl(Boy &, Girl &),作为以上两个类的友元函数，主函数中通过调用该函数输出男孩和女孩的信息。写出该友元函数的完整代码，以及主函数的代码。

```
void VisitBoyGirl(Boy& boy, Girl& girl) {
    cout << "Boy's name: " << boy.Name << endl;
    cout << "Boy's age: " << boy.Age << endl;
    cout << "Girl's name: " << girl.Name << endl;
    cout << "Girl's age: " << girl.Age << endl;
}
int main() {
    Girl g("yuxiaojing", 22);
    Boy b("yuminghong", 19);
    VisitBoyGirl(b, g);
    return 0;
}
```

四、实验小结（包括问题和解决方法、心得体会、意见与建议等）

（中文五号宋体，英文五号 Consolas 字体，单倍行距)

(一)实验中遇到的主要问题及解决方法

1.在题目 2 中不改变 main()函数中的对象的定义方式，若取消构造函数中参数的默认值，编译程序错误提示信息及出错原因是：

错误提示信息：不能接受 0 个参数，没有重载函数接受两个参数。

出错原因：实际参数个数不能少于无默认值的形式参数个数。

2.在题目 2 中如果删除类中自定义的构造函数，仅使用系统默认构造函数，再编译，程序错误提示信息及出错原因是：

错误提示信息：没有与参数列表匹配的构造函数。

出错原因：系统默认的构造函数无形式参数。

3.在题目 2 中如果将 main()函数中的输出语句改为：cout<<对象名.Hour<<":"<<对象名.Minute<<":"<<对象名.Second<<endl; 重新编译，会出现什么错误提示？在这种情况下，如果将成员变量的访问属性修改为 public 再编译，结果如何？

错误提示信息：Time::Hour 无法访问 private 成员、Time::Minute 无法访问 private 成员 Time::Second 无法访问 private 成员。

结果：成功运行。

4.其它问题及解决办法

问题：函数声明时指定默认参数后，函数首部再次指定。

解决方法：删除函数首部的默认值，只保留形参。

（二）实验心得

在这次实验中，我学到了类与对象的基本概念及其在面向对象编程中的重要性。通过定义类和创建对象，我能够更清晰地组织和管理程序代码，将数据和操作封装在一起，提高了代码的可维护性和可重用性。我理解了对象之间通过消息传递来进行通信的机制，这为我以后设计更复杂的程序提供了基础。

另外，在实验中我正确掌握了类的不同属性成员的使用方法，包括成员变量和成员函数的定义和调用。特别地，我深入理解了构造函数和析构函数的概念，以及它们在对象生命周期中的作用。构造函数在对象创建时被调用，用于初始化对象的状态，而析构函数则在对象被销毁时执行清理工作，这有助于避免资源泄漏和内存溢出问题。

另一个重要的概念是友元函数和友元类的使用。通过友元函数和友元类，我可以在需要访问类的私有成员时进行授权，而不违反封装的原则。这为我提供了一种灵活的方式来管理程序的访问权限，同时保护数据的安全性。

最后，我还学会了如何在函数参数中使用指针和引用。指针和引用作为参数传递给函数时，可以直接修改实参的值，而不是通过复制参数的副本来操作。这提高了程序的效率，并且在处理大型数据结构时尤其有用。通过这次实验，我对面向对象编程的理解更加深入，并且掌握了一些实用的编程技巧，这对我的编程能力有着积极的影响。

（三）意见与建议（没有可省略）

可以提供更多的时间上机操作，以确保更多程序设计思路得以实现，提升面向对象语言的掌握程度和编程能力。

实 验 报 告

五、支撑毕业要求指标点

信息安全：

1.2-M 掌握计算机软硬件相关工程基础知识，能将其用于分析信息安全领域的相关工程问题。

3.1-H 掌握信息安全领域所涉及的软硬件系统，从数字电路、计算机系统、到各类系统软件的基本理论与设计结构。

六、指导教师评语

	评分项	优秀	良好	中等	合格	不合格
评分细则	遵守实验室规章制度					
	学习态度					
	算法思想准备情况					
	程序设计能力					
	解决问题能力					
	算法设计合理性					
	算法效能评价					
	报告书写认真程度					
	内容详实程度					
	文字表达熟练程度					
	其它评价意见					
	本次实验能力达成评价（总成绩）		批阅人		日期	