

山西工程技术学院

实验报告 1

(2023 -2024 学年第 1 学期)

课程名称: 面向对象程序设计实训

专业班级

学 号:

学生姓名:

任课教师:

2023 年 11 月

实验名称	输入输出、选择结构、循环结构			指导教师	
实验类型	验证型	实验学时	4	实验时间	8.29
一、实验目的 (1) 掌握 C++ 输入输出的使用; (2) 掌握 if 语句的三种形式及其应用; (3) 掌握 switch 语句的应用, 区分语句中有 break 语句和无 break 语句的不同;					
二、实验环境 Dev C++;					
三、实验内容 (1) 编程计算销售公司销售利润个人奖金提成;输入一个数, 编程, 计算从 0 到该数的所有数的阶乘并输出。 <pre> #include <iostream> using namespace std; int main() { double p; cout<<"please enter the amunt:"<<endl; cin>>p; int earn; if(p<=100000) { earn = p * 0.1; } else if(p>100000&&p<=200000) { earn = 10000+((p-100000)*0.075); } else if(p>200000&&p<400000) { earn = 10000+(100000*0.075)+((p-200000)*0.05); } else if(p>400000&&p<600000) { earn = 10000+(100000*0.075)+(200000*0.05)+((p-400000)*0.03); } </pre>					

```
else if(p>600000)
{
    earn = 10000+(100000*0.075)+(200000*0.05)+(200000*0.03)+((p-600000)*0.015);
}
cout << earn << endl;
return 0;}
```

500000

30500

四、实验小结

- (1) 该程序运用了三种 if 的结构 if, else if, else 来进行程序的判断。
- (2) 使用了一些数学的知识，需要有一定的逻辑思维和数学基础。
- (3) 要掌握 C++与 C 语言输入输出的区别，并熟练运用并且掌握。

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--

实验名称	函数			指导教师	
实验类型	验证型	实验学时	4	实验时间	9.12

三、实验目的

- (1) 掌握函数定义的一般形式；
- (2) 掌握函数参数和函数值的概念，掌握函数参数的传递方式；
- (3) 掌握函数调用的形式及执行过程；
- (4) 理解函数的嵌套调用和递归调用；

四、实验环境

Dev C++;

三、实验内容

(1) 输入几个字符，通过定义一个对字符进行加密处理的函数，在主函数中反复调用，输出加密后的字符。

```
#include <iostream>
using namespace std;
int main()
{
    int a,b;
    int offset = 3;
    char input[5];
    char ch[5];
    for(a=0;a<5;a++)
    {
        cin>>input[a];
        ch[a]=input[a];
    }
    for(a=0;a<5;a++)
    {
        if (ch[a] >= 'a' && ch[a] <= 'z')
        {
            ch[a] = (ch[a] - 'a' + 3) % 26 + 'a';
        }
        else if (ch[a] >= 'A' && ch[a] <= 'Z')
        {
            ch[a] = (ch[a] - 'A' + 3) % 26 + 'A';
        }
    }
}
```

```
cout<<"加密后的字符为： ";
    for(a=0;a<5;a++)
    {
        printf("  %c",ch[a]);
    }
    return 0;
}
```

abced

加密后的字符： d e f h g

五、实验小结

（1）学会运用数组进行字符的转码解码；同时掌握一维数组的定义和用法；用数组存入字符。

（2）同时掌握函数参数和函数值的概念，掌握函数参数的传递方式；

（3）掌握函数调用的形式及执行过程，理解函数的嵌套调用和递归调用，运用嵌套和循环来减小运算的时间，以及我们书写代码的长度，

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--

实验名称	类和对象			指导教师	
实验类型	验证型	实验学时	4	实验时间	9.26
一、实验目的 (1) 掌握类的定义方法；掌握类的数据成员和成员函数的定义方法； (2) 掌握类的构造函数和析构函数的作用及使用方法。					
二、实验环境 Dev C++;					
三、实验内容 (1) 定义一个学生类，其中有 8 个数据成员有学号、姓名、性别、专业、年级、年龄、住址、电话，以及若干成员函数。编写函数使用这个类，实现对学生数据的赋值和输出。 <pre> #include<iostream> #include<string.h> using namespace std; //Student 类 class Student { public: Student(); Student(string ID, string name, char sex, string birthday, int grade, string stu_class, string institute, string major); Student(const Student& obj); ~Student(void); void SetInfo(string ID, string name, char sex, string birthday, int grade, string stu_class, string institute, string major); void Show(); int Check_input(); private: </pre>					

```

string ID;
    string name;
    char sex;
    string birthday;
    int grade;
    string stu_class;
    string institute;
    string major;
};
Student::Student(void)
{
    cout<<"object is being created by constructor without parameters!"<<endl;
}
Student::Student(string ID, string name, char sex, string birthday, int grade, string
stu_class, string institute, string
major):ID(ID),name(name),sex(sex),birthday(birthday),grade(grade),stu_class(stu_c
lass),institute(institute),major(major)
{
    cout<<"object is being created by constructor with paramters!"<<endl;
}
Student::Student(const Student& obj)
{
    ID=obj.ID;
    name=obj.name;
    sex=obj.sex;
    birthday=obj.birthday;
    grade=obj.grade;
    stu_class=obj.stu_class;
    institute=obj.institute;
    major=obj.major;
}
Student::~~Student()
{
    cout<<"Student object is being deleted\n"<<endl;
}
void Student::SetInfo(string ID, string name, char sex, string birthday, int grade,
string stu_class, string institute, string major)
{
    this->ID=ID;
    this->name=name;
    this->sex=sex;
    this->birthday=birthday;
    this->grade=grade;
    this->stu class=stu class;

```

```

        this->institute=institute;
        this->major=major;
    }
void Student::Show()
{
    cout<<"学生"<<name<<"的基本信息如下:"<<endl;
    cout<<"\t 学号: "<<ID<<endl;
    cout<<"\t 性别: "<<sex<<endl;
    cout<<"\t 出生日期: "<<birthday<<endl;
    cout<<"\t 年级: "<<grade<<endl;
    cout<<"\t 班级: "<<stu_class<<endl;
    cout<<"\t 院系: "<<institute<<endl;
    cout<<"\t 专业: "<<major<<"\n";
}
int Student::Check_input()
{
    int length;
    length=ID.length();
    if(length!=7){
        cout<<"输入错误: 学号格式不正确!"<<endl;
        if( !(sex=='F' || sex=='M' || sex=='f' || sex=='m') ){
            cout<<"输入错误: 性别格式不正确!"<<endl;
        }
        return -1;
    }
    return 1;
}
int main()
{
    string ID;
    string name;
    char sex;
    string birthday;
    string stu_class;
    string institute;
    string major;
    int grade;
    int n;
    char flag;
    cout<<"请输入要输入信息的学生个数: ";
    cin>>n;
    Student *stu[n];
    for(int i=0;i<n;i++){
        cout<<endl<<"请输入学生"<<i+1<<"的信息: "<<endl;

```



```

        cout<<"学号： ";
        cin>>ID;
        cout<<flush<<"姓名： ";
        cin>>name;
        cout<<flush<<"性别(F/M)： ";
        cin>>sex;
        cout<<flush<<"出生日期 Year-month： ";
        cin>>birthday;
        cout<<flush<<"年级： ";
        cin>>grade;
        cout<<flush<<"班级： ";
        cin>>stu_class;
        cout<<flush<<"院系： ";
        cin>>institute;
        cout<<flush<<"专业： ";
        cin>>major;
        stu[i]=new Student(ID,name,sex,birthday,grade,stu_class,institute,major);
        if(stu[i]->Check_input()<0){
            cout<<endl<<"是否需要对学生"<<name<<"的信息进行修改？ Y/N"<<endl;
            cin>>flag;
            if(flag=='Y' || flag=='y'){
                cout<<"请输入修改后学生的信息： "<<endl;
                cout<<"学号： ";
                cin>>ID;
                cout<<"姓名： ";
                cin>>name;
                cout<<"性别(F/M)： ";
                cin>>sex;
                cout<<"出生日期 Year-month： ";
                cin>>birthday;
                cout<<"年级： ";
                cin>>grade;
                cout<<"班级： ";
                cin>>stu_class;
                cout<<"院系： ";
                cin>>institute;
                cout<<"专业： ";
                cin>>major;

                stu[i]->SetInfo(ID,name,sex,birthday,grade,stu_class,institute,major);
            }
        }
    }
}

```

```

cout<<endl<<"以下是所有学生的基本信息：\n"<<endl;
for(int i=0;i<n;i++){
    stu[i]->Show();
    delete stu[i];
}
return 0;
}

```

请输入要输入信息的学生个数：5

以下是所有学生的基本信息：

学号：2210708130

姓名：郝泓毅

性别（F/M）：M

出生日期2004.7 年级：22

班级：1

院系：计算机系

专业：计算机科学与技术

是否要对学生郝泓毅的信息进行修改？Y/N

Y

请输入修改后的学生信息：

学号：

四、实验小结

（1）注意带参数的构造函数和带默认参数的构造函数的声明与定义；从键盘输入基本信息，调用带参数的构造函数生成学生对象；当输入数据为不合理数据时，提示用户输入错误。

（2）要熟练掌握类的定义方法；掌握类的数据成员和成员函数的定义方法以便更流畅的书写程序。

（3）要学会使用析构函数，构造函数并熟练掌握并且运用。

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--

实验名称	继承与派生			指导教师	
实验类型	综合设计型	实验学时	6	实验时间	10.24

五、实验目的

- (1) 掌握如何通过继承建立类的层次结构;
- (2) 掌握通过类指针操作继承关系中对象的方法;
- (3) 掌握派生类的构造和析构函数的定义和使用;

六、实验环境

Dev C++;

三、实验内容

(1) 新建一个雇员类,有雇员编号、年龄、工资、性别、姓名,输入雇员资料方法,打印雇员资料方法;派生一个教师类,计算教师工资;派生一个实验员工资类,计算实验员工资;派生一个行政人员类,计算行政人员工资。、

```
#include<iostream>
#include<cstring>
#include<string>
using namespace std;
class Student {
public:
    Student(string number,string name,string age,string sex,string
address,string department) {
        m_number = number;
        m_name = name;
        m_age = age;
        m_sex = sex;
        m_address = address;
        m_department = department;
    }
    ~Student() {
    }
    void display() {
        cout << "学号: " << m_number << ", 姓名: " + m_name + ", 年龄: " << m_age
<< ", 性别: " + m_sex << endl;
        cout << "家庭地址: " + m_address + ", 所属院系: " + m_department << endl;
    }
}
```

```

protected:
    string m_number;           //学号
    string m_name;             //姓名
    string m_age;              //年龄
    string m_sex;              //性别
    string m_address;          //家庭地址
    string m_department;       //所属院系
};
class PostGraduate:public Student {
private:
    string m_tutor;            //导师
    string m_allowance;        //津贴
    string m_research;         //研究方向
public:
    PostGraduate(string number, string name, string age, string sex, string address,
string department,string tutor, string allowance, string research)
        :Student(number, name, age, sex, address,
department) ,m_tutor(tutor),m_allowance(allowance),m_research(research) {}
    void display() {}
friend void operator<<(ostream& cout,PostGraduate& p);
};
void operator<<(ostream& cout,PostGraduate& p) {
    cout <<"学号: " + p.m_number + ",姓名: " +p.m_name + ",年龄: " +p.m_age + ",性
别: " + p.m_sex << endl;
    cout <<"家庭地址: " << p.m_address + ", 所属院系: " + p.m_department << endl;
    cout <<"导师: " + p.m_tutor << ",津贴: " <<p.m_allowance << ",研究方向: " +
p.m_research << endl;
}
int main()
{
    Student s1("20190001", "Michael", "19", "Male", "机场路 10 号", "Computer
Science");
    //参数分别为: 学号, 姓名, 年龄, 性别, 家庭地址和所属院系
    s1.display();

    PostGraduate p1("20190001", "Michael", "19", "Male", "机场路 10 号", "Computer
Science",
    "Liu", "1000", "Deep learning");//导师: "Liu", 津贴: "1000", 研究方向: "Deep
learning"
    p1.display();
    cout << p1;
    return 0;
}

```

学号：20190001，姓名：Michael，年龄：19，性别：Male
家庭地址：机场路10号，所属院系：Computer Science
学号：20190001,姓名：Michael,年龄：19,性别：Male
家庭地址：机场路10号，所属院系：Computer Science
导师：Liu,津贴：1000,研究方向：Deep learning

六、实验小结

- (1) 自己学习了解继承，并且通过继承建立类的层次结构;使数据形成一个完整的逻辑。
- (2) 要学会熟练运用指针，通过类指针操作继承关系中对象的方法;真正学会自如地使用指针进行各种程序的书写。
- (3) 书写此程序，同时还要掌握派生类的构造和析构函数的定义和使用。

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--

实验名称	综合设计型			指导教师	
实验类型	综合设计型	实验学时	6	实验时间	11.07

七、实验目的

- (1) 掌握运算符重载的规则和使用
- (2) 掌握定义虚函数和抽象基类的方法;

八、实验环境

Visual Studio Code 2019

三、实验内容

(1) 新建一个汽车类，包括数据成员由车名、车主名、购买日期，同时包括一个显示车辆信息的成员函数。

```
#include <iostream>
#include <string>
// 日期类，简化问题，只考虑年月日
class Date {
public:
    int year, month, day;
    Date(int y, int m, int d) : year(y), month(m), day(d) {}
};
// 车的基类
class Vehicle {
protected:
    std::string carName;
    std::string ownerName;
    Date purchaseDate;
public:
    Vehicle(const std::string& name, const std::string& owner, int y, int m, int d)
        : carName(name), ownerName(owner), purchaseDate(y, m, d) {}
    virtual void displayInfo() const {
        std::cout << "Car Name: " << carName << std::endl;
        std::cout << "Owner: " << ownerName << std::endl;
        std::cout << "Purchase Date: " << purchaseDate.year << "-" <<
purchaseDate.month << "-" << purchaseDate.day << std::endl;
    }
};
```

```

// 小汽车类，继承自 Vehicle
class Car : public Vehicle {
private:
    int seatCount;

public:
    Car(const std::string& name, const std::string& owner, int y, int m, int d, int
seats)
        : Vehicle(name, owner, y, m, d), seatCount(seats) {}

    void displayCarInfo() const {
        displayInfo(); // 调用基类的 displayInfo 函数
        std::cout << "Seat Count: " << seatCount << std::endl;
    }
};

// 货车类，继承自 Vehicle
class Truck : public Vehicle {
private:
    double tonnage;

public:
    Truck(const std::string& name, const std::string& owner, int y, int m, int d,
double tonnageVal)
        : Vehicle(name, owner, y, m, d), tonnage(tonnageVal) {}

    void displayTruckInfo() const {
        displayInfo(); // 调用基类的 displayInfo 函数
        std::cout << "Tonnage: " << tonnage << " tons" << std::endl;
    }
};

int main() {
    Car myCar("Toyota", "John", 2020, 5, 10, 5);
    myCar.displayCarInfo();
    std::cout << "-----" << std::endl;
    Truck myTruck("Ford", "Jane", 2022, 2, 15, 10.5);
    myTruck.displayTruckInfo();
    return 0;
}

```

Car Name: Toyota
Owner: John
Purchase Date: 2020-5-10
Seat Count: 5

Car Name: Ford
Owner: Jane
Purchase Date: 2022-2-15
Tonnage: 10.5 tons

七、实验小结

- (1) 这段代码首先定义了一个简单的 Date 类来表示日期。
- (2) 接着，我们定义了 Vehicle 基类，其中包含车名、车主名和购买日期，并有一个函数 displayInfo 用于显示车辆信息。
- (3) 然后我们定义了两个从 Vehicle 派生的子类：Car 和 Truck，它们分别添加了座位数和吨位数据成员，并各自有一个函数用于显示详细信息。

实验成绩		批阅日期		批阅人	
------	--	------	--	-----	--