**湖 北 大 学**

计算机与信息工程学院

2022——2023学年度

第二学期

**学 生 实 验 报 告 册**

学生姓名：

班 级：

学 号：

课程名称：

任课老师：

学生实验守则

1、学生在规定的时间内进行实验，不得无故缺席或迟到。

2、学生在每次实验前对排定要做的实验应进行预习，并按要求作好预习报告。

3、每次实验前，必须交上次实验报告和本次实验预习报告，并经指导教师提问、检查同意后，才可进行本次实验。

4、学生进入实验室指定位置后，首先根据仪器清单核对自己使用的仪器是否有缺少或损坏，发现问题及时向指导教师报告，严禁擅自动用别组仪器。

5、实验时必须有实事求是、严肃认真的科学态度，严格遵守仪器操作规程和注意事项。

6、实验完毕应将实验数据交给指导教师检查，合格后，整理复原好仪器设备，方可离开实验室。

7、保持实验室肃静和整洁，不得大声喧哗，乱丢垃圾和吃东西。

8、学生在实验过程中，由于不遵守操作规程或未经许可，擅自进行实验而造成事故、损坏仪器设备，应及时报告，并填写损坏清单，按院有关规定进行赔偿。

实 验 报 告 单

**实验名称： 1**

同组人： 实验课时：

实验室： 报告日期：

**一、实验目的：**

熟练掌握类和对象，熟悉类中的构造函数，析构函数，对象数组，对象指针，共用数据的保护，静态成员和友元的概念。

**二、实验内容：**

**第1题：**建立一个对象数组，内放5个学生的数据(学号、成绩)，用指针指向数组首元素，输出第1,3,5个学生的数据。

**第2题：**阅读下面程序4-1，分析其执行过程，写出输出结果。

程序4-1：

#include <iostream>

using namespace std;

class Student

{public:

Student(int n,float s):num(n),score(s){ }

void change(int n,float s){num=n;score=s;}

void display( ){cout<<num<<″ ″<<score<<endl;}

private:

int num;

float score;

};

int main( )

{Student stud(101,78.5);

stud.display( );

stud.change(101,80.5);

stud.display( );

return 0;

**第3题：**将第2题的程序4-1分别作以下修改，分析所修改部分的含义以及编译和运行的情况。

(1) 将main函数第2行改为const Student stud(101,78.5);

(2) 在(1)的基础上修改程序，使之能正常运行，用change函数修改数据成员num和score的值。

(3) 将main函数改为

int main( )

{Student stud(101,78.5);

Student \*p=&stud;

p->display( );

p->change(101,80.5);

p->display( );

return 0;}

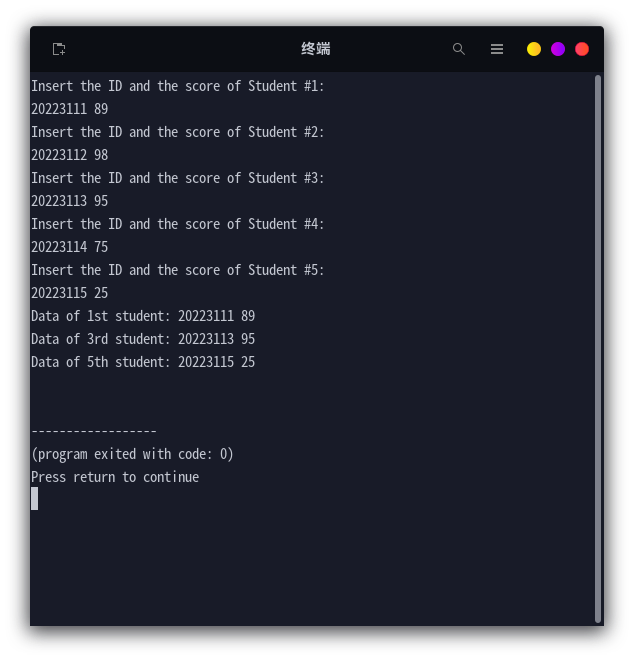
其他部分仍同第2题的程序。

(4) 在(3)的基础上将main函数第3行改为

const Student \*p=&stud;

(5) 再把main函数第3行改为Student \*const p=&stud;

**第4题：**修改第2题的程序4-1，增加一个fun函数，改写main函数。在main函数中调用fun函数，在fun函数中调用change和display函数。在fun函数中使用对象的引用(Student &)作为形参。



**三、实验结果：**

4-1.

#include <iostream>

using namespace std;

class Student{

public:

long student\_id;

double score;

};

int main(){

Student \*ptr = new Student[5];

for (int i = 0; i < 5; i++) {

cout << "Insert the ID and the score of Student #" << i+1 <<":"<< endl;

cin >> ptr[i].student\_id >> ptr[i].score;

}

cout << "Data of 1st student: " << ptr->student\_id << " " << ptr->score << endl;

ptr += 2;

cout << "Data of 3rd student: " << ptr->student\_id<< " " << ptr->score << endl;

ptr += 2;

cout << "Data of 5th student: " << ptr->student\_id << " " << ptr->score << endl;

return 0;}

4-2.

#include <iostream>

using namespace std;

class Student

{public:

Student(int n,float s):num(n),score(s){ }

void change(int n,float s){num=n;score=s;}

void display(){cout<<num<<""<<score<<endl;}

private:

int num;

float score;

};

int main( )

{Student stud(101,78.5);

stud.display( );

stud.change(101,80.5);

stud.display( );

return 0;

}

/\* 执行过程：

\* 程序定义了一个名为Student的对象

\* 内部有 change 和 display 两个 public 数据成员函数

\* 和 num 和 score 两个 private 数据成员变量

\* 主函数中首先定义了一个名为 stud 的 Student 类型

\* 初始化其数据为 101, 78.5

\* 然后通过Student中的display成员函数使其显示，显示为101 78.5

\* 之后运行Student中的change成员函数将数据改为101, 80.5

\* Student 对象中的成员函数可以对 Student 内定义的 private 数据进行修改

\* 故再次执行stud.display();函数应输出101 80.5

 \*/

/\*

\* 运行结果：

\* 101 78.5

\* 101 80.5

\*/

4-3.(1)

#include <iostream>

using namespace std;

class Student

{public:

Student(int n,float s):num(n),score(s){ }

void change(int n,float s){num=n;score=s;}

void display(){cout<<num<<""<<score<<endl;}

private:

int num;

float score;

};

int main( )

{const Student stud(101,78.5);

stud.display( );

stud.change(101,80.5);

stud.display( );

return 0;

}

/\* 结果：无法通过编译，提示报错。

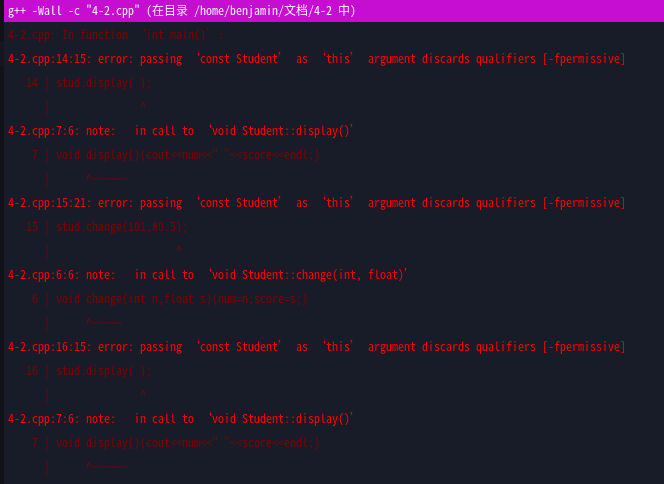
\* 原因：

\* 将main函数中的第二行修改为const Student stud(101,78.5)后，在main函数中定义的是一个Student类型的常量对象 stud

\* 此时stud的成员函数只能调用常量成员函数，而不能调用非常量成员函数。

\* 在main函数中，stud.change(101,80.5)尝试修改常量对象stud的值，但是change函数不是常量成员函数，因此编译器会报错。

\*/



(2)

#include <iostream>

using namespace std;

class Student

{public:

Student(const int n,const float s):num(n),score(s){ }

void change (const int n,const float s) const{num=n; score=s;}

void display() const{cout<<num<<" "<<score<<endl;}

private:

mutable int num;

mutable float score;

};

int main( )

{const Student stud(101,78.5);

stud.display( );

stud.change(101,80.5);

stud.display( );

return 0;

}

/\* 分析：

\* 此程序将原来的几个成员函数改成了常量成员函数

\* 并将num, score改成了mutable量

\*/

/\*

\* 运行结果：

\* 101 78.5

\* 101 80.5

\*/

(3)

#include <iostream>

using namespace std;

class Student

{public:

Student(const int n,const float s):num(n),score(s){ }

void change (const int n,const float s) const{num=n; score=s;}

void display() const{cout<<num<<" "<<score<<endl;}

private:

mutable int num;

mutable float score;

};

int main( )

{Student stud(101,78.5);

Student \*p=&stud;

p->display( );

p->change(101,80.5);

p->display( );

return 0;}



/\*

\* 分析：

\* 相较于原程序，此程序创建了一个指向stud对象的指针p

\* 接着调用了p的display函数输出了num和score的值

\* 然后，调用了p的change函数，将num改为101，将score改为80.5。

\* 最后，再次调用了p的display函数，输出了修改后的num和score的值

\*/

/\*

\* 运行结果：

\* 101 78.5

\* 101 80.5

\*/

（4）

#include <iostream>

using namespace std;

class Student

{public:

Student(const int n,const float s):num(n),score(s){ }

void change (const int n,const float s) const{num=n; score=s;}

void display() const{cout<<num<<" "<<score<<endl;}

private:

mutable int num;

mutable float score;

};

int main( )

{Student stud(101,78.5);

const Student \*p=&stud;

p->display( );

p->change(101,80.5);

p->display( );

return 0;}

/\*

\* 分析：

\* 相较于（3），此程序将指向stud对象的指针p定义为指向常对象的指针

\* 指向常对象的指针也可以指向普通对象，但该对象将被视为常对象

\* main中的第五行语句 p->change(101,80.5); 语句试图修改对象 stud 的值

\* 这意味着，如果change函数中传入的两个参数对应的量为常量，编译器会出现报错

\* 但p指向的常对象stud中，num和score被定义为mutable类型，故因此运行结果会与（3）一致

\*/

（5）

#include <iostream>

using namespace std;

class Student

{public:

Student(const int n,const float s):num(n),score(s){ }

void change (const int n,const float s) const{num=n; score=s;}

void display() const{cout<<num<<" "<<score<<endl;}

private:

mutable int num;

mutable float score;

};

int main( )

{Student stud(101,78.5);

Student \*const p=&stud;

p->display( );

p->change(101,80.5);

p->display( );

return 0;}



/\*

\* 分析：

\* 相较于（3），此程序将指向stud对象的指针p定义一个常量指针

\* 因此不能通过p来修改stud对象的值。

\* 但是，由于Student类中的num和score成员变量被声明为mutable

\* 因此change函数可以修改它们的值，即使在const函数中也可以

\*/

4-4.

#include <iostream>

using namespace std;

class Student

{public:

Student(int n,float s):num(n),score(s){ }

void change(int n,float s){num=n;score=s;}

void display( ){cout<<num<<" "<<score<<endl;}

void fun(int n,float s){display();change(n,s);display();}

private:

int num;

float score;

};

int main( )

{Student stud(101,78.5);

stud.fun(101,80.5);

return 0;}

———————————————————————————————

**成绩：**

批阅教师：

日 期：