

**C++程序设计实验报告**



实验题目 面向对象的程序设计

学 号 201806061108

姓 名 胡皓睿

班 级 计实1801

提交日期 2019.4.7

1. 实验目的：

1. 掌握一个多文件程序的编译。

2. 掌握用成员函数实现某项功能。

3. 掌握利用构造函数对类对象进行初始化。

1. 实验内容：

1.阅读下面的程序，分析其执行过程，写出输出结果。

#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;

}

2. 将第1题程序分别作以下修改，分析所修改部分的含义以及编译和运行的情况。

（1）○1程序代码

#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;

}

(3）○1程序代码

#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);

Student \*p=&stud;

p->display();

p->change(101,80.5);

p->display();

return 0;

}

（4）○1程序代码

#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);

const Student \*p=&stud;

p->display();

p->change(101,80.5);

p->display();

return 0;

（5）○1程序代码

#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);

Student \*const p=&stud;

p->display();

p->change(101,80.5);

p->display();

return 0;

}

3. 商店销售某一商品，每天公布统一的折扣（discount）。同时允许销售人员在销售时灵活掌握售价（price），在此基础上，一次购10件以上者，还可以享受9.8折优惠。现已知当天3个销售员销售情况为：

销货员号（num） 销货件数（quantity） 销货单价（price）

101 5 23.5

102 12 24.56

103 100 21.5

请编写程序，计算出当日此商品的总销售款sum以及每件商品的平均售价。要求用静态数据成员和静态成员函数。

4. 将例9.13程序中的display函数不放在Time类中，而作为类外的普通函数，然后分别在Time和Date类中将display声明为友元函数。在主函数中调用display函数，display函数分别引用Time和Date两个类的对象的私有数据，输出年、月、日和时、分、秒。

9.13：有一个日期（Date）类的对象和一个时间（Time）类的对象，均已指定了内容，要求一次输出其中的日期和时间。

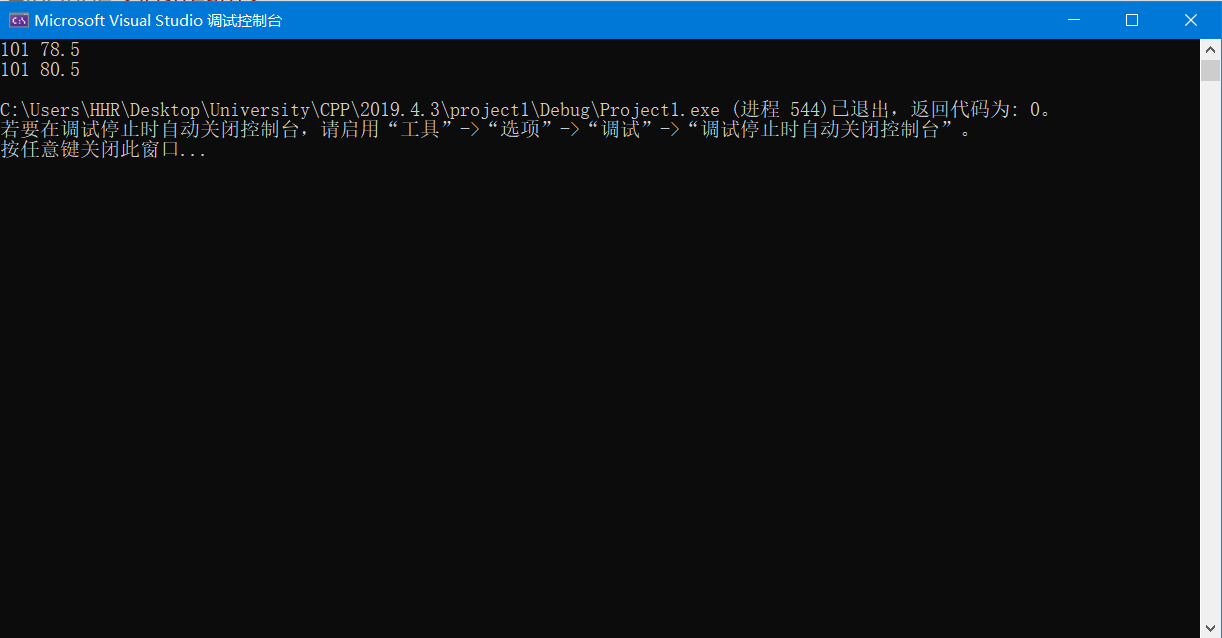
5. 将例9.14改写为在类模板外定义各成员函数。

9.14：声明一个类模板，利用它分别实现两个整数、浮点数和字符的比较，求出大数和小数。

三、实验过程

实验1：

直接运行代码，即可获得结果



实验2：

#include <iostream>

using namespace std;

class Student

{

public:

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

const void change(int n, float s)

{

num = n;

score = s;

}

const void display()

{

cout << num << " " << score << endl;

}

public:

int num;

float score;

};

int main()

{

const Student stud(101, 78.5);

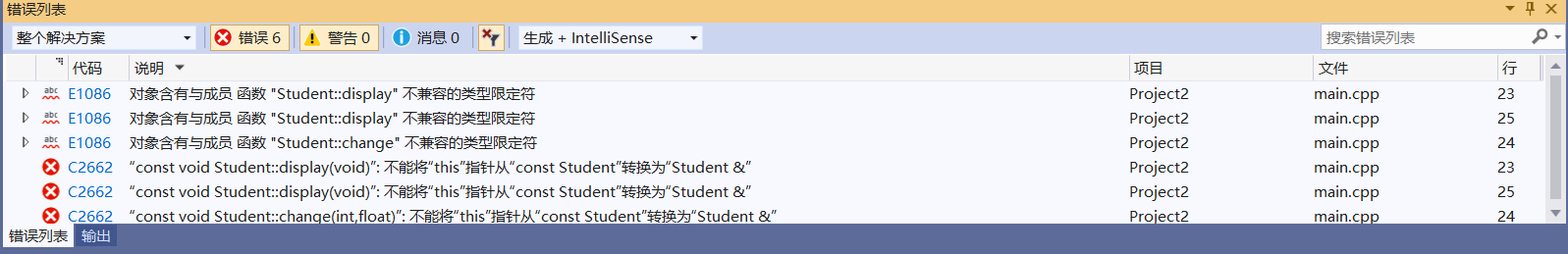
stud.display();

stud.change(101, 80.5);

stud.display();

return 0;

}



实验3

代码如下：

#include <iostream>

using namespace std;

class p

{

private:

int num;

int quantity;

float price;

public:

static float discount;

void input()

{

cin >> num >> quantity >> price;

}

float money()

{

return float(quantity) \* price\* discount / 10;

}

void inputdiscount()

{

cin >> discount;

}

};

float p::discount = 9.8;

int main()

{

p people[3];

float sum = 0;

for (int x = 0; x < 3; x++)

{

people[x].input();

}

people[0].inputdiscount();

for (int x = 0; x < 3; x++)

{

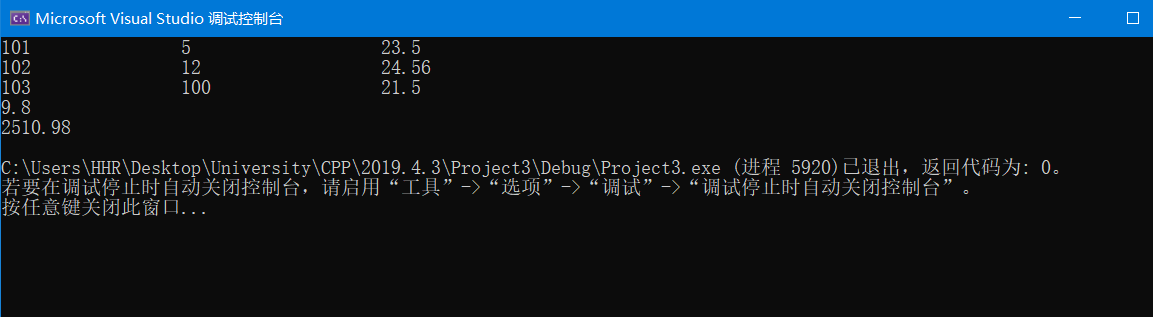
sum += people[x].money();

}

cout << sum << endl;

return 0;

}



实验4

#include <iostream>

using namespace std;

class Date;

class Time

{

public:

Time(int, int, int);

friend void display(const Date&,const Time&);

private:

int hour;

int minute;

int second;

};

class Date

{

public:

Date(int, int, int);

friend void display(const Date&,const Time&);

private:

int month;

int day;

int year;

};

Time::Time(int h, int m, int s)

{

hour = h;

minute = m;

second = s;

}

void display(const Date& d,const Time&t)

{

cout << d.month << "/" << d.day << "/" << d.year << endl;

cout << t.hour << ":" << t.minute << ":" << t.second << endl;

}

Date::Date(int m, int d, int y)

{

month = m;

day = d;

year = y;

}

int main()

{

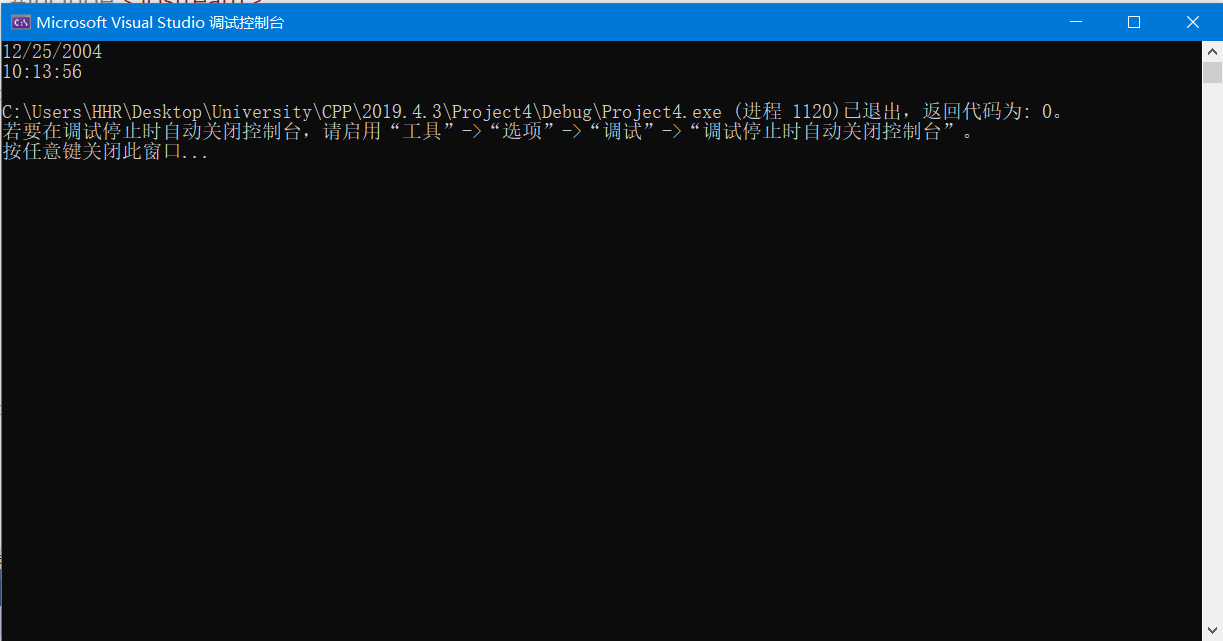
Time t1(10, 13, 56);

Date d1(12, 25, 2004);

display(d1,t1);

return 0;

}



实验5：

#include <iostream>

using namespace std;

template <class numtype>

class Compare

{

public:

Compare(numtype a, numtype b);

numtype max();

numtype min();

private:

numtype x, y;

};

template <class numtype>

Compare<numtype>::Compare(numtype a, numtype b)

{

x = a;

y = b;

}

template <class numtype>

numtype Compare<numtype>::max()

{

return (x > y) ? x : y;

}

template <class numtype>

numtype Compare<numtype>::min()

{

return (x < y) ? x : y;

}

int main()

{

Compare <int> cmp1(3, 7);

cout << cmp1.max() << "........................" << endl;

cout << cmp1.min() << "........................" << endl;

Compare <float> cmp2(45.78, 93.6);

cout << cmp2.max() << "........................" << endl;

cout << cmp2.min() << "........................" << endl;

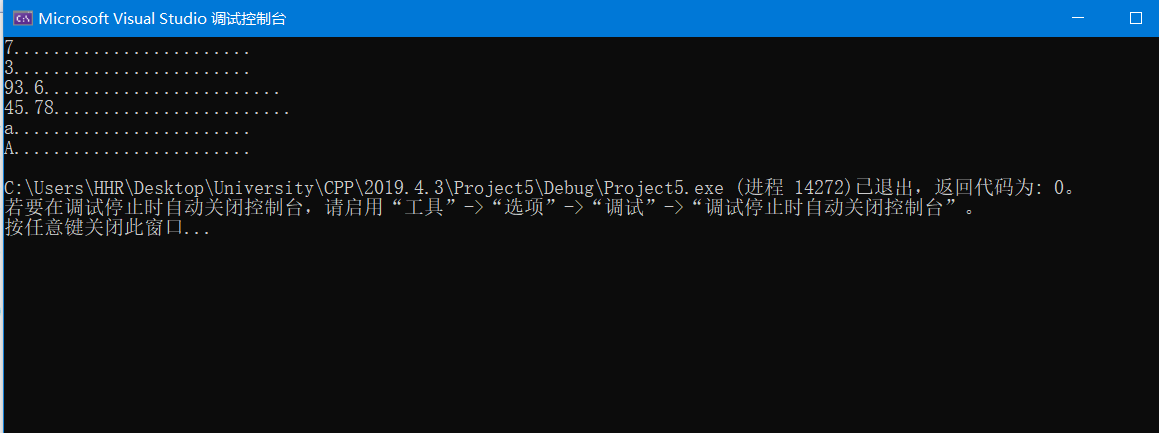
Compare <char> cmp3('a','A');

cout << cmp3.max() << "........................" << endl;

cout << cmp3.min() << "........................" << endl;

return 0;

}



四、小结与收获

应用了类和对象的简单应用，了解了简单的对一个对象进行封装处理，是处理复杂问题的其中一小步，学会了怎样使用类和对象，利用构造函数对类对象进行初始化，构造函数的重载，收获很多，而且要把以前面向过程的思想转移到面向对象，支撑了毕业要求1中的专业知识用于解决复杂软件工程问题。

【毕业要求1】能够将数学、自然科学、工程基础和专业知识用于解决复杂软件工程问题。

支撑指标点1.3 能够将工程基础和专业知识用于求解软件领域复杂工程问题

【毕业要求3】能够设计针对复杂工程问题的解决方案，设计满足特定需求的系统、单元（部件）或工艺流程，并能够在设计环节中体现创新意识，考虑社会、健康、安全、法律、文化以及环境等因素

支撑指标点3.1：掌握解决复杂工程问题的程序设计语言基础。

【毕业要求5】使用现代工具：能够针对复杂工程问题，开发、选择与使用恰当的技术、资源、现代工程工具和信息技术工具，包括对复杂工程问题的预测与模拟，并能够理解其局限性

支撑指标点5.1：了解现代工程工具和信息技术工具的使用方法,了解实际计算机工程系统中各种现代工具的使用现状。