中国矿业大学计算机学院实验报告

课程名称	高级语言程序设计		实验名称	高级语言程序设计实践	
班级	信息安全 2019-1 班	姓名	李春阳	学号	10193657
仪器组号			实验日期	2020.12.8	
实验报告要求：1.实验目的 2.实验内容（题目描述，源代码，运行截图，调试情况） 3.实验体会					

一、实验目的

1. 认识了解 c++基本语法。
2. 掌握条件语句和循环语句
3. 掌握数组和字符串的表达。
4. 掌握类的基本用法。

二、实验内容

1、第一题

1.1 题目描述

A. 问题描述

自定义一个复数类型 Complex，其中含有若干成员函数，使用该类可以完成复数的加法以及对复数的输出。请完成类定义，并编制主函数，说明 Complex 类对象，对定义的各成员函数进行调用。

```
class Complex {
        double real; //复数实部
        double imag; //复数虚部
public:
        Complex (); //无参构造函数，将复数对象的实部和虚部均置为0
        Complex (double r, double i); //有参构造函数，设置对象的实部和虚部
        Complex AddCom(Complex c2); //调用者对象与对象 c2 相加，返回 Complex 类对象
        void OutCom () ; //输出调用者对象的有关数据（各分量）
};
```

具体 要求如下：

1、实现有参构造函数 Complex (double r, double i);

2、实现 Complex AddCom(Complex c2); 调用者对象与对象 c2 相加，返回 Complex 类对象

3、实现 void OutCom () ;实现输出调用者对象的有关数据分量（一定要输出虚部的符号 i），如果该数为纯虚数时,不需要输出实部，当虚部为 0 时，不需要输出虚部。

4．编制主函数 main，作用有参函数说明类对象 cx,cy，使用 Complex 调用 AddCom 实现复数加法，并将相加的结果调用 OutCom 方法以复数的形式输出。

B. 输入

输入包括 a,b,c,d 四个整数，第一个复数为 a+bi，第二个复数为 c+di

C. 输出

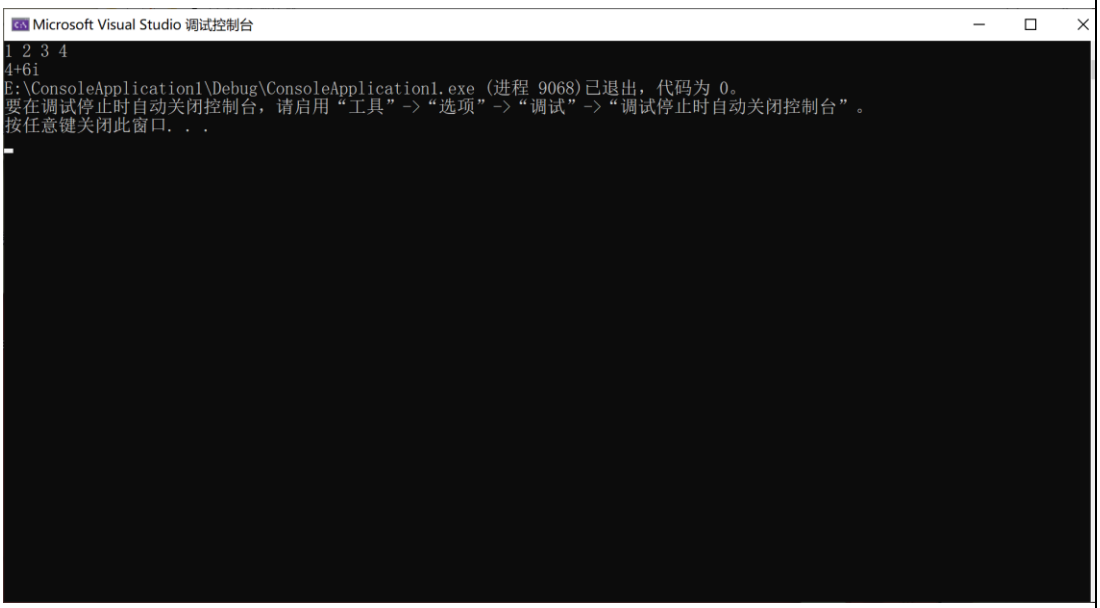
复数

1.2 源代码

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

class Complex
{
private:
    double real;
    double imag;

public:
    Complex()
    {
        real = 0;
        imag = 0;
    }
    Complex(double, double);
    Complex AddCom(Complex);
    void OutCom();
};


Complex::Complex(double r, double i)
{
    real = r;
    imag = i;
}


Complex Complex::AddCom(Complex c2)
{
    (*this).imag += c2.imag;
    (*this).real += c2.real;
    return *this;
}


void Complex::OutCom()
{
    if (real == 0)
```

```
        cout << imag << "i";
    else if (imag == 0)
        cout << real;
    else
        cout << real << "+" << imag << "i";
}


int main()
{
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    Complex cx(a, b);
    Complex cy(c, d);
    cx.AddCom(cy);
    cx.OutCom();
    return 0;
}
```

1.3 运行截图



1.4 调试情况

2、第二题

2.1 题目描述

A. 问题描述、

自定义一个简单日期类 DateType，它具有数据成员 y、m、d，用来表示当前日期的年、月、日。

而后设计该类欲实现（完成）的功能，进而设计出相应的类成员函数。

```
class DateType {
//自定义的日期类 DateType
        int y,m,d; //数据成员，表示当前日期的年、月、日
public:
        DateType(int y0=1, int m0=1, int d0=1);
        //构造函数，设定年、月、日；并设置参数默认值
        void IncrementDay(); //增加 1 天
        bool Equal(DateType dt2); //判断二日期是否相等
        void PrintDate(); //屏幕输出日期对象的有关数据（年、月、日）
};
```

1、完成有参构造函数

2、完成 void IncrementDay()函数，计算天数加 1 后的日期。

3、完成 Equal(DateType dt2)函数，判断两个日期是否相等。

4、完成打印输出函数 PrintDate()，在屏幕上输出日期对象的有关数据（年、月、日）

5、编写并完成主函数，实现输入包含六个整数，说明 DateType 类对象 dt1,dt2,分别是 dt1 和 dt2 的年月日。请先输出 dt1，dt2，然后判断 dt1，dt2 是否相等，再对 dt1,dt2 分别增加一天。最后输出 dt1，dt2。

注意：在 IncrementDay 成员函数中，当对日期增加 1 天后，要注意所谓的“进位”问题：首先算出本“日”所在的月份具有的天数 N（注意闰年与平年的 2 月份天数不一样），若加 1 之后的“日”数值超过所在的月份具有的天数 N 时，“进位”到月，而月份若超过 12 时还要“进位”到年等。

B. 输入

两个日期的月、日、年

C. 输出

相关要求显示

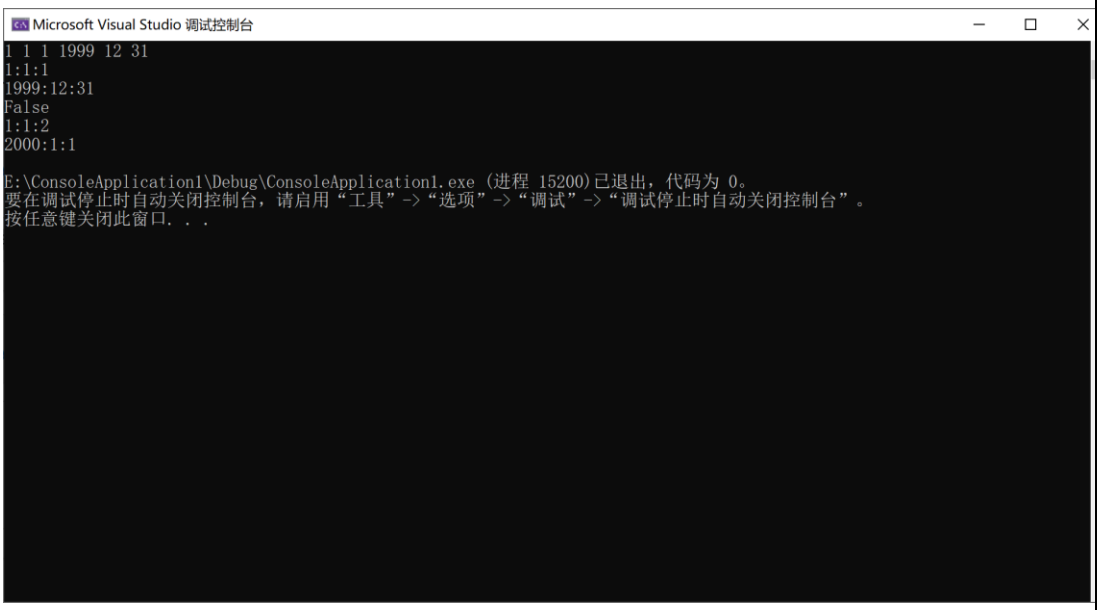
2.2 源代码

```
#include<iostream>

using  namespace  std;

int  mon[13] = { 0,31,28,31,30,31,30,31,31,30,31,30,31 };

class  DateType
{
private:
    int y;
    int m;
    int d;
public:
    //构造函数，设定年、月、日；并设置参数默认值
    DateType(int  y0 = 1, int  m0 = 1, int  d0 = 1)
    {
        y = y0;
        m = m0;
        d = d0;
    }
    void  IncrementDay();//增加 1 天
    bool  Equal(DateType  d2);//判断二日期是否相等
    void  PrintDate()//屏幕输出日期对象的有关数据（年、月、日）
    {
        cout << y << ":" << m << ":" << d << endl;
    }
};


void  DateType::IncrementDay()
{
    if (m == 2)
    {
        if (!(((y % 4) != 0) || ((y % 100) == 0) && ((y % 400) != 0)))
        {
            if ((d + 1) <= 29)
                d++;
            else
            {
                m++;
                d = 1;
            }
        }
```

```
        else
        {
            if ((d + 1) <= 28)
                d++;
            else
            {
                m++;
                d = 1;
            }
        }
    }
    else
    {
        if ((d + 1) <= mon[m])
            d++;
        else if ((m + 1) <= 12)
        {
            m++;
            d = 1;
        }
        else
        {
            y++;
            m = 1;
            d = 1;
        }
    }
}


bool DateType::Equal(DateType  d2)
{
    if (y == d2.y && m == d2.m && d == d2.d)
        return true;
    else return false;
}

int  main()
{
    int  d1_y, d1_m, d1_d, d2_y, d2_m, d2_d;
    cin >> d1_y >> d1_m >> d1_d >> d2_y >> d2_m >> d2_d;

    DateType  dt1(d1_y, d1_m, d1_d), dt2(d2_y, d2_m, d2_d);
    dt1.PrintDate();
```

```
    dt2.PrintDate();

    if (dt1.Equal(dt2))
        cout << "True" << endl;
    else
        cout << "False" << endl;

    dt1.IncrementDay();
    dt2.IncrementDay();
    dt1.PrintDate();
    dt2.PrintDate();
    return 0;
}
```

2.3 运行截图



2.4 调试情况

3、第三题

3.1 题目描述

A. 问题描述

设计一个学生类（CStudent），其私有数据成员：注册号、姓名、数学、外语、计算机课程的成绩。公有成员函数是：求三门课总成绩的函数 Sum；求三门课平均成绩的函数 Average；显示学生数据信息的函数 Display；设置学生数据信息的函数 SetData。

1. 可按如下样式设计 CStudent 类的各数据成员以及成员函数

class CStudent { //学生类 CStudent

```
        unsigned long reg_num; //数据成员：注册号
        char name[30]; //数据成员：姓名
        float math, eng, comp; //数据成员：数学、英语、计算机成绩
public: //公有成员函数
        float Sum(); //求三门课总成绩的函数 Sum
        float Average(); //求三门课平均成绩的函数 Average
        Display(); //显示学生数据信息的函数 Display
        SetData (unsigned long r, char* n, float m, float e, float c) ;
        //设置学生数据信息的函数 SetData
};
```

在主函数，通过使用“CStudent stu[150];”的语句，来说明一个 CStudent 类对象的数组 stu，而后通过各对象 stu[i]来处理并求取每一学生的总成绩、平均成绩等。

（1）输入本次欲处理的学生人数 TOTAL（小于等于 150 的正整数）；

（2）输入全班 TOTAL 个学生的有关信息，依次放入对象数组的各元素 stu[i]中（通过使用“stu[i].SetData(...);”形式的语句来实现）；

（3）对全班 TOTAL 个学生，依次通过对象 stu[i]来求出其总成绩、平均成绩等（其中要使用形如“stu[i].Sum()”以及“stu[i].Average()”式样的对成员函数进行调用的语句），并同时求出全班学生总成绩最高者处于 stu 数组的下标位置 idx_max，而后通过使用“stu[idx_max].Display();”来输出该学生有关的全部数据信息。

3．程序执行后的输入输出界面样式可设计为：

TOTAL=3

CStudent 1 : 100001 ma 78 86 90（注意空格）

CStudent 2 : 100002 li 85 91 88

CStudent 3 : 100003 hu 82 89 88

CStudent1.Sum=254,CStudent1.average=84.6667

CStudent2.Sum=264,CStudent2.average=88

CStudent3.Sum=259,CStudent3.average=86.3333

class_Sum_max=264

The infomation of the CStudent with class_Sum_max : 100002 li 85 91 88

B. 输入

TOTAL=3

CStudent 1 : 100001 ma 78 86 90（注意空格）

CStudent 2 : 100002 li 85 91 88

CStudent 3 : 100003 hu 82 89 88

C. 输出

CStudent1.Sum=254,CStudent1.average=84.6667

CStudent2.Sum=264,CStudent2.average=88

CStudent3.Sum=259,CStudent3.average=86.3333

class_Sum_max=264

The infomation of the CStudent with class_Sum_max : 100002 li 85 91 88

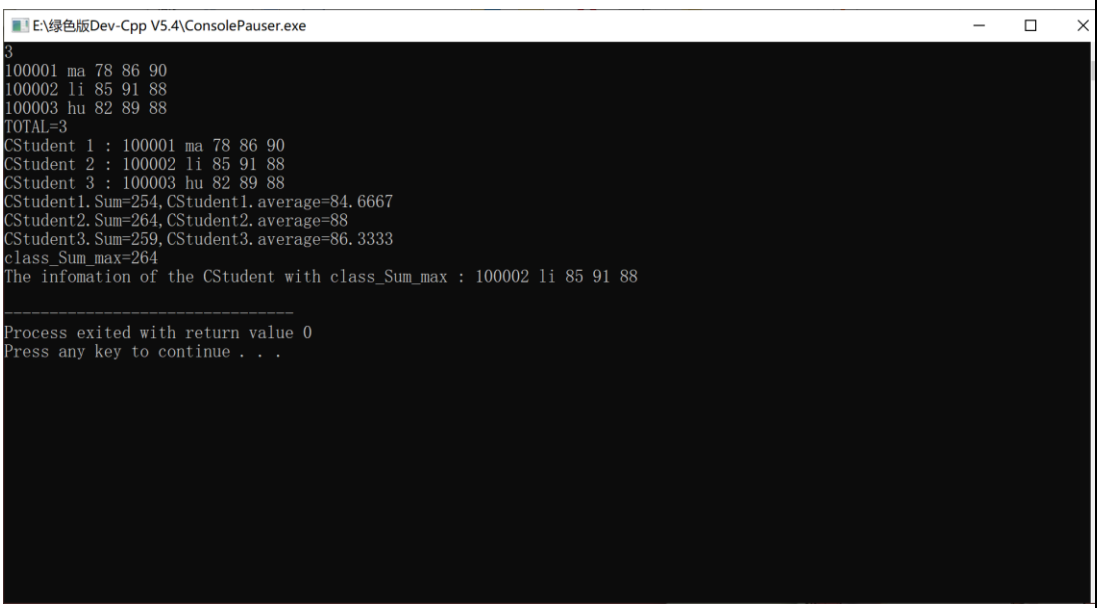
3.2 源代码

```
#include <iostream>
#include <cstring>
#include <string>

using namespace std;

class Cstudent
{
    unsigned reg_num;
    char name[30];
    float math, eng, comp;

public:
    float Sum();
    float Average();
    void Display();
    void setData(unsigned long r, char* n, float m, float e, float c);
};

void Cstudent::Display()
{
    cout << reg_num << " " << name << " " << math << " " << eng << " " << comp <<
endl;
}
```

```
float Cstudent::Sum()
{
    return(math + eng + comp);
}

float Cstudent::Average()
{
    return (*this).Sum() / 3;
}

void Cstudent::setData(unsigned long r, char* n, float m, float e, float c)
{

    reg_num = r;
    strcpy(name, n);
    math = m;
    eng = e;
    comp = c;
}



int main()
{
    int n;
    unsigned long num;
    char name[30];
    float math, eng, comp;
    float max = 0;
    int max_stu = 0;

    cin >> n;
    Cstudent* cp = new Cstudent[n];

    int j;
    for (j = 0; j < n; j++)
    {
        cin >> num >> name >> math >> eng >> comp;
        cp[j].setData(num, name, math, eng, comp);
    }
    for (j = 0; j < n; j++)
    {
        if (max < cp[j].Sum())
        {
            max_stu = j;
```

```
            max = cp[j].Sum();
        }

    }

    cout << "TOTAL=" << n << endl;
    for (int i = 0; i < n; i++)
    {
        cout << "CStudent " << i + 1 << " : ";
        cp[i].Display();
    }
    for (int i = 0; i < n; i++)
    {
        cout << "CStudent" << i + 1 << ".Sum=" << cp[i].Sum();
        cout << ",CStudent" << i + 1 << ".average=" << cp[i].Average() << endl;
    }
    cout << "class_Sum_max=" << max << endl;
    cout << "The infomation of the CStudent with class_Sum_max : ";
    cp[max_stu].Display();

    delete[] cp;
    return 0;
}
```

3.3 运行截图



3.4 调试情况

4、第四题

4.1 题目描述

A. 问题描述

设计一个 Point(点)类，数据信息包含 x 轴和 y 轴的坐标。设计一个 Circle(圆)类，数据信息包含圆心和半径。

要求：（1）数据部分都采用整型；

（2）圆心作为 Circle 类中的子对象；

（3）每个类都包含带有参数的构造函数；

（4）重载运算符“<<”和“>>”，用于输入输出每个类对象的数据信息；

（5）主函数内验证各个功能。

主函数参考代码：

```
int main()
{   Point p(0,0);
    cin>>p;
    cout<<p;
    Circle c(0,0,0);
    cin>>c;
    cout<<c;
    return 0;
}
```

B. 输入

输入有两行，第一行两个整数，分别代表 x 轴和 y 轴的坐标值；第二行三个整数，分别代表 x 轴坐标值、y 轴坐标值和半径值。

C. 输出

输出三行，第一行是点的坐标，形式为（x，y）；第二行是圆心坐标，形式仍为（x，y）；第三行是半径值，最后有换行。注意：输出可以和输入交叉出现。

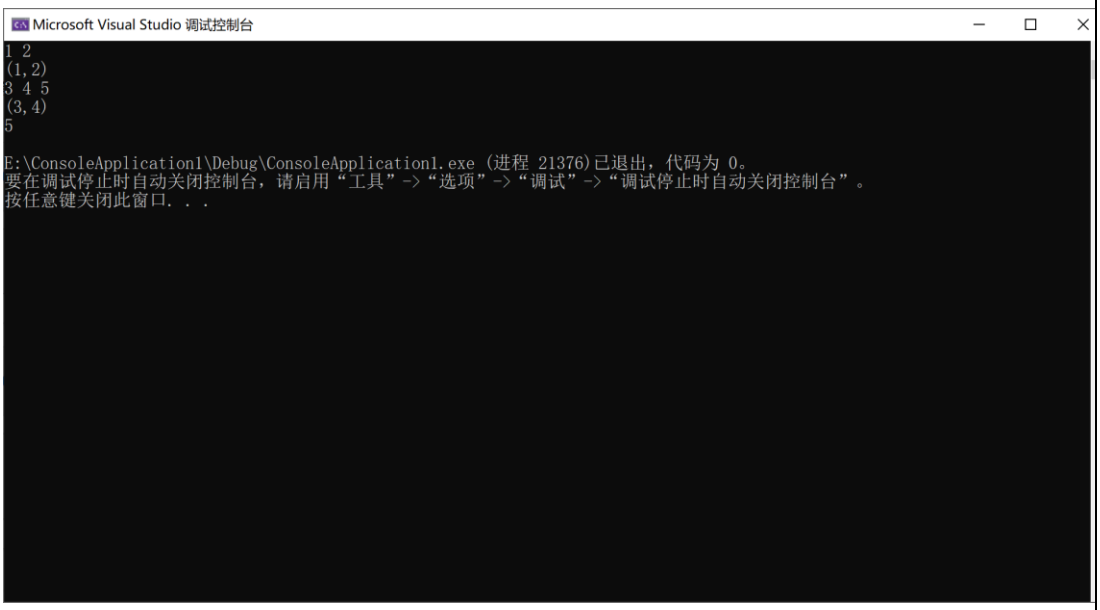
4.2 源代码

```
#include<iostream>

using namespace std;
```

```
class Point
{
    int x;
    int y;
public:
    Point(int x = 0, int y = 0)
    {
        this->x = x;
        this->y = y;
    }
    friend istream& operator>>(istream& is, Point& p);
    friend ostream& operator<<(ostream& os, Point p);
};

istream& operator>>(istream& is, Point& p) {
    is >> p.x >> p.y;
    return is;
}

ostream& operator<<(ostream& os, Point p) {
    os << "(" << p.x << "," << p.y << ")";
    return os;
}

class Circle {
    Point O;
    int r;
public:
    Circle(int a, int b, int r) {
        O = *new Point(a, b);
        this->r = r;
    }
    friend istream& operator>>(istream& is, Circle& c);
    friend ostream& operator<<(ostream& os, Circle c);
};

istream& operator>>(istream& is, Circle& c) {
    cin >> c.O >> c.r;
    return is;
}

ostream& operator<<(ostream& os, Circle c) {
```

```
    cout << c.O << endl << c.r;
    return os;
}

int main()
{
    Point p(0, 0);
    cin >> p;
    cout << p << endl;
    Circle c(0, 0, 0);
    cin >> c;
    cout << c << endl;
    return 0;
}
```

4.3 运行截图



4.4 调试情况

三、实验体会

通过这次实验，我更加明白了 c++理论中的一些编程规范和 c++语言特性，掌握了基本编程知识，以后会更加认真的学习 c++理论知识，并不断实践和练习，在 debug 中不断学习。