**海南大学理学院**

**C++语言程序设计**

**实验报告**

学 号：20233002662

姓 名：崔志豪

专业班级：数学与应用数学

指导老师：黄萍

成 绩：

评 语：

2024年 10 月 24 日

# 实验一 Visual Studio2019开发环境应用入门

## 一、实验目的

1．了解Visual Studio2019的特点。

2．熟悉Visual Studio2019的开发环境。

3．学习用Visual Studio2019编写标准的C+ +控制台程序。

## 二、实验内容

1．使用Visual Studio2019建立一个非图形化的标准C+ +程序，编译、运行以下程序：

#include<iostream>

using namespace std;

int main( )

{

cout<<“Hello!\ n”;

cout<<“Welcome to C+ +! \ n”;

}

2．编写程序，输出如下图案：

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

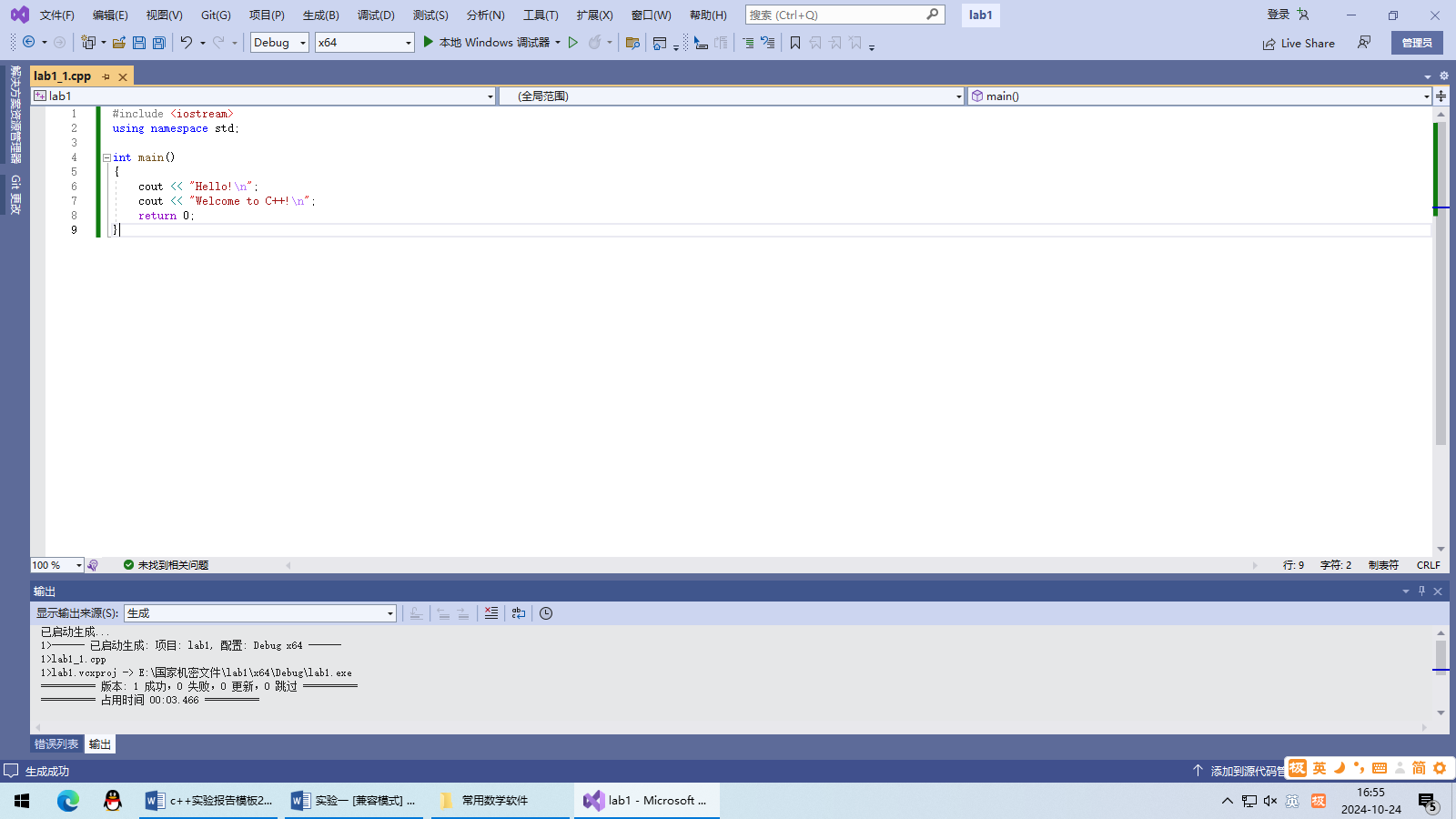
\*\*\*

\*

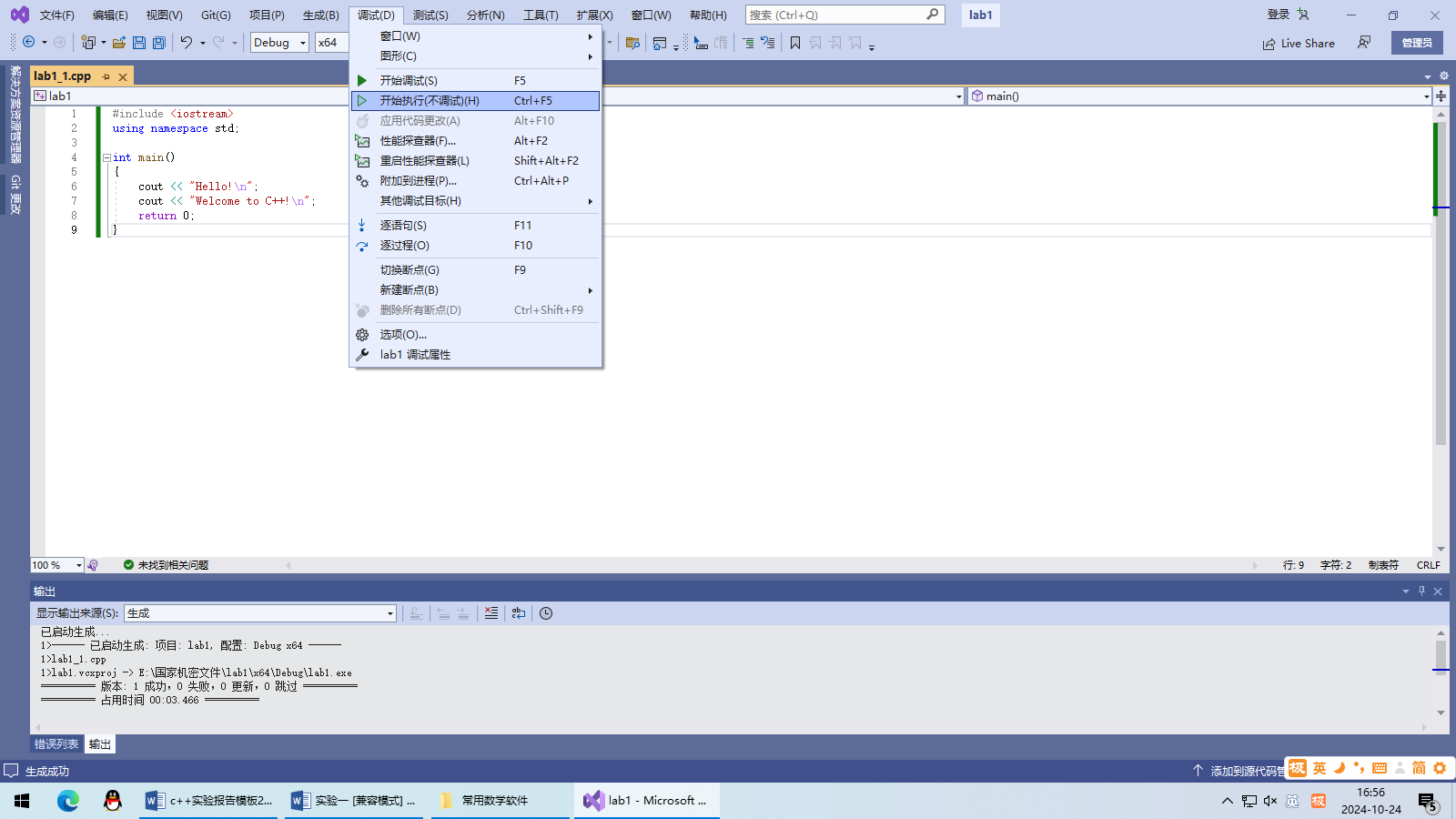
1. 编写一个简单的C++ 程序

要求：输入圆的半径，计算圆的周长和面积。

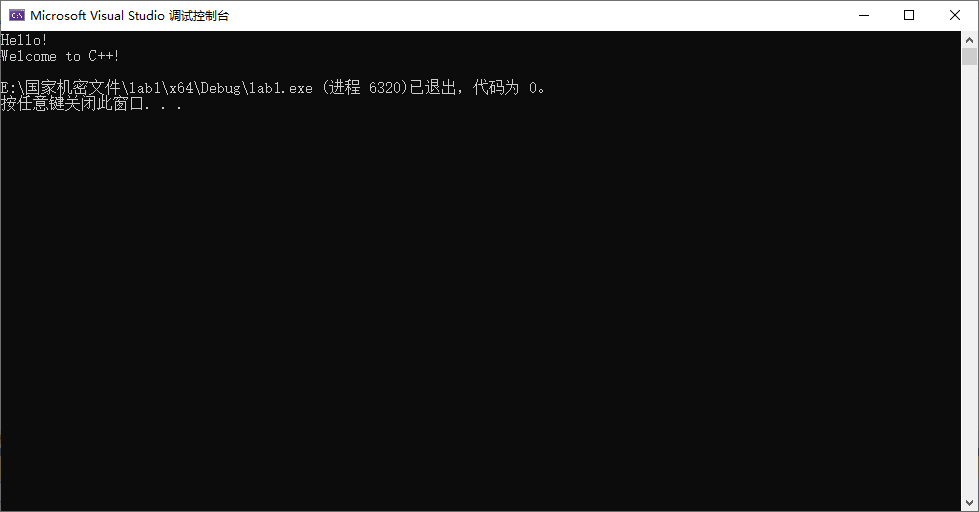
1. **实验步骤与结果：**
2. （1）编写源程序：



1. 编译、运行程序：



1. 运行结果：



1. （1）源程序：

#include <iostream>

using namespace std;

int main()

{

cout << "\*" << endl;

cout << "\*\*\*" << endl;

cout << "\*\*\*\*\*" << endl;

cout << "\*\*\*\*\*\*\*" << endl;

cout << " \*\*\*\*\*" << endl;

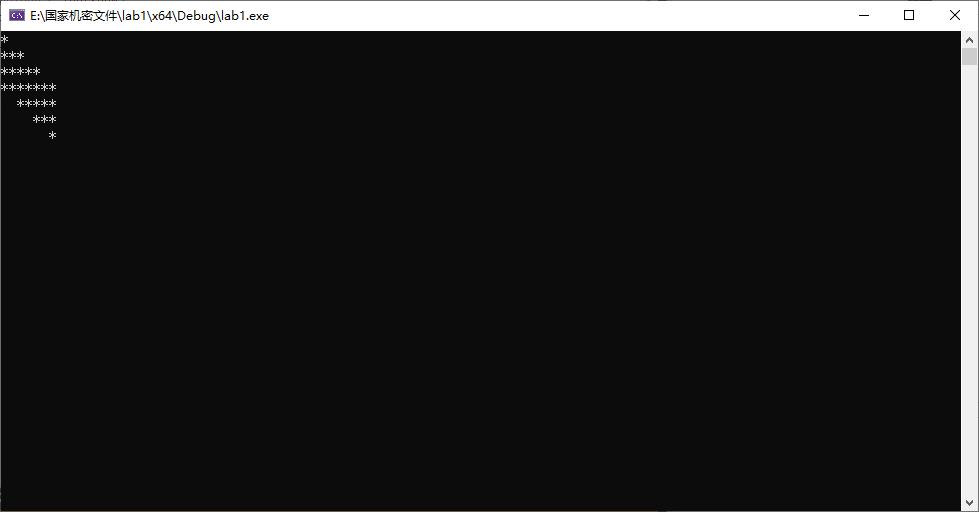
cout << " \*\*\*" << endl;

cout << " \*" << endl;

cin.get();

{

1. 运行结果：



1. （1）源程序：

#include <iostream>

using namespace std;

const double PI = 3.14;

//length

double Circle(double radius)

{

double C = 2 \* PI \* radius;

return C;

}

//Area

double Area(double radius)

{

double S = PI \* radius \* radius;

return S;

}

int main()

{

double r;

cout << "please enter the radius" << endl;

cin >> r;

double C = Circle(r);

double S = Area(r);

cout << "The length of this circle is " << C << endl;

cout << "The area of this circle is " << S << endl;

cin.get();

}

（2）运行结果：

**四、分析与总结**

1.在lab1\_2中用暴力输出方法，得到结果，也可以考虑用循环的方式；

2.在lab1\_3中，采用了模块化处理的方式，定义double Circle和double Area函数来分别对用户输入的radius进行程序操作；

# 实验二 C+ +简单程序设计

**一、实验目的**

1．学会编写简单C+ +程序。

2．复习基本数据类型变量和常量的应用。

3．复习运行符与表达式的应用。

4．复习结构化程序设计基本控制结构的运用。

5．复习自定义数据类型的应用。

6．练习使用简单的输入输出。

7．观察头文件的作用。

**二、实验内容**

1．输入并运行例2-7，即用do-while语句编程，求自然数1～10之和。程序正确运用之后，去掉源程序中#include语句，重新编译，观察会有什么问题。

2．将do-while语句用for语句代替，完成相同的功能。

3．编程计算图形的面积。程序可计算圆形、长方形、正方形的面积，运行时先提示用户选择图形的类型，然后，要求用户对圆形输入半径值，要求用户对长方形输入长和宽的值，要求用户对正方形输入边长的值，计算出面积的值后将其显示出来。

4．实现一个简单的菜单程序，运行时显示“Menu: A(dd) D(elete) S(ort) Q(uit), Select one:” 提示用户输入。A表示增加，D表示删除，S表示排序，Q表示退出。输入为A、D、S时分别提示“数据已经增加、删除、排序。”，输入为Q时程序结束。要求使用switch语句。

**三、实验步骤和运行结果**

1.（1）源程序

#include <iostream>

using namespace std;

int main()

{

int sum = 0 , i = 1;

do {

sum += i;

i++;

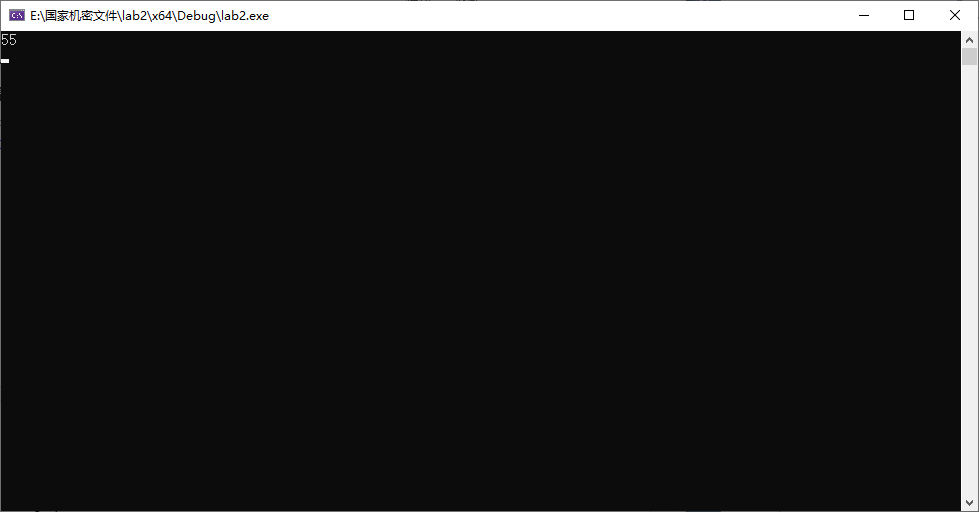
} while (i <= 10);

cout << sum << endl;

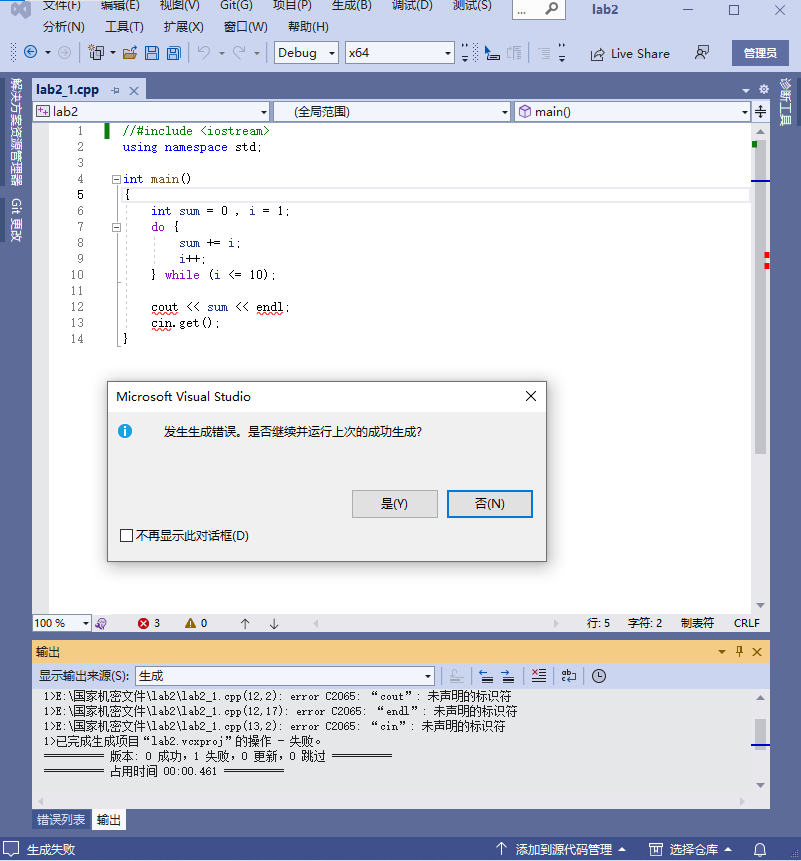
cin.get();

}

1（2）.原程序运行截图



1（3）.删除#include语句后截图



2（1）.源程序、

#include <iostream>

using namespace std;

int main()

{

int sum = 0, i = 0;

for ( i = 0; i <= 10; i++)

{

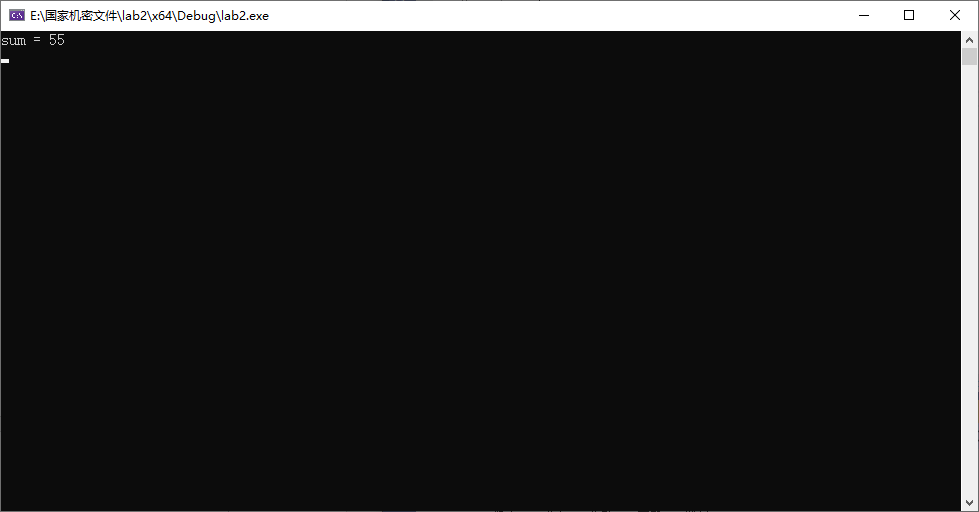
sum += i;

}

cout << "sum = " << sum << endl;

cin.get();

}2（2）程序运行截图



3(1).源程序

#include <iostream>

using namespace std;

double CircleArea(double radius)

{

double Area;

Area = 3.14 \* radius \* radius;

return Area;

}

double ChangArea(double l, double s)

{

double Area;

Area = l \* s;

return Area;

}

double ZhengArea(double a)

{

double Area;

Area = a \* a;

return Area;

}

int main()

{

int i;

cout << "请输入你要求的面积类型：1-圆，2-长方形，3-正方形：";

cin >> i;

switch (i)

{

case 1:

{

double radius;

cout << "请输入圆的半径：";

cin >> radius;

double AreaC = CircleArea(radius);

cout << AreaC << endl; break;

}

case 2:

{

double l, s;

cout << "请依次输入长方形的长和宽" << endl;

cin >> l >> s;

double Area = ChangArea( l, s);

cout << Area << endl;break;

}

case 3:

{

double a;

cout << "请输入正方形的边长:";

cin >> a;

double Area = ZhengArea(a);

cout << Area << endl; break;

}

default:

break;

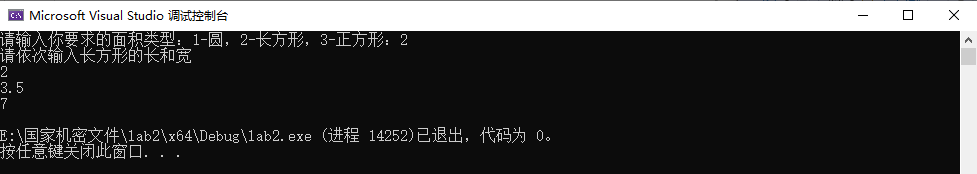
}

cin.get();

}

3（2）.运行截图







4(1).源程序

#include <iostream>

using namespace std;

int main()

{

char Key;

cout << "Menu: A(dd) D(elete) S(ort) Q(uit), Select one:" << endl;

cin >> Key;

while (Key != 'Q') {

switch (Key)

{

case 'A':

cout << "数据已增加" << endl;

cout << "Menu: A(dd) D(elete) S(ort) Q(uit), Select one:" << endl;

cin >> Key;

break;

case 'D':

cout << "数据已删除" << endl;

cout << "Menu: A(dd) D(elete) S(ort) Q(uit), Select one:" << endl;

cin >> Key;

break;

case 'S':

cout << "数据已排序" << endl;

cout << "Menu: A(dd) D(elete) S(ort) Q(uit), Select one:" << endl;

cin >> Key;

break;

default:

break;

}

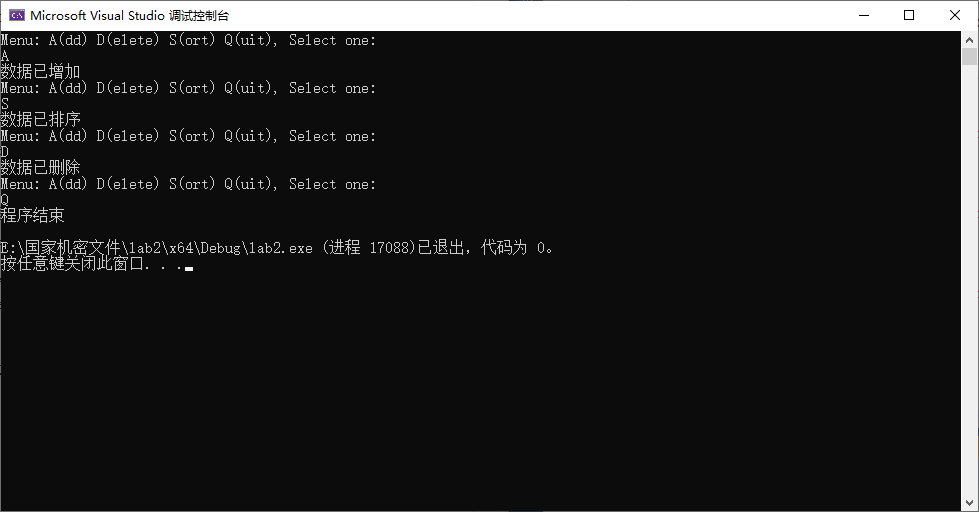
}

cout << "程序结束" << endl;

cin.get();

}

4（2）.运行截图



**四、分析与总结**

1.在第三个实验中，定义了CircleArea,ChangArea,ZhengArea三个函数，首先对用户的期望进行识别，然后使用switch语句来调用三个函数实现功能；

2.在第四个实验中，利用while循环来对程序进行优化，实现多次对数据的操作，减轻了操作员的工作压力；

3.实验表明,do while和for循环在功能上是一致的。

# 实验三 函数的应用

**一、实验目的**

1．掌握函数的定义和调用方法。

2．练习重载函数的使用。

3．练习使用系统函数。

**二、实验内容**

1．编写一个函数把华氏温度转换为摄氏温度，转换公式为：C=(F-32)\*5/9。

2．编写重载函数Maxl可分别求取2个整数、3个整数、2个双精度数、3个双精度数的最大值。

3．使用系统函数pow(x , y)计算的值，注意包含头文件cmath。

4．用递归的方法编写函数求Fibonacci级数，观察递归调用的过程。

**三、实验步骤**

1.（源代码）

//lab3\_1 摄氏度华氏度转换

#include <iostream>

using namespace std;

double F\_H(double f);

int main()

{

double c , f;

cout << "请输入要转换的华氏温度:";

cin >> f;

c = F\_H(f);

cout << c << endl;

cin.get();

}

double F\_H(double f) {

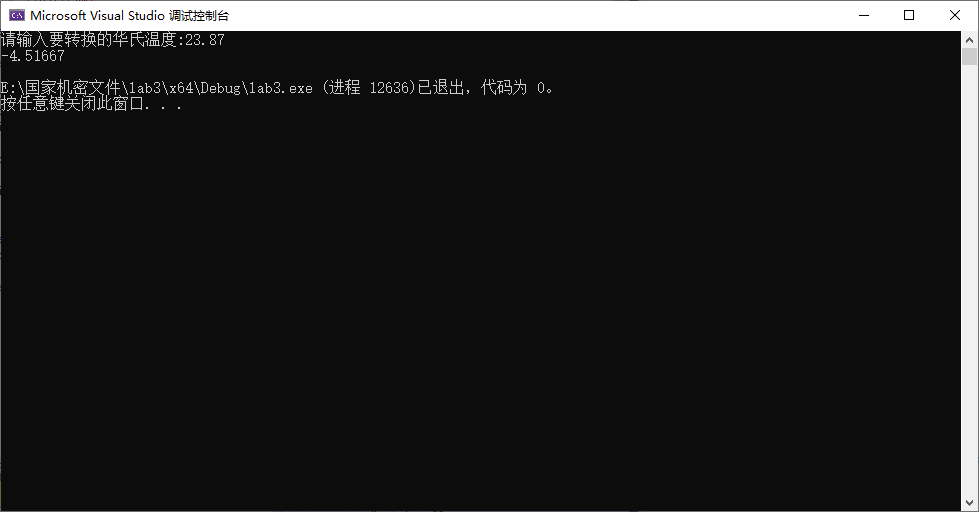
double c;

c = (f - 32) \* 5 / 9;

return c;

}

（运行结果）



2.（源代码）

#include <iostream>

using namespace std;

int Max1(int a, int b) {

return max(a, b);

}

int Max1(int a, int b , int c) {

return max(max(a, b), c);

}

double Max1(double a, double b) {

return max(a, b);

}

double Max1(double a, double b , double c) {

return max(max(a, b), c);

}

int main()

{

int a, b, c;

double m, n, q;

cout << "请输入三个整数：" << endl;;

cin >> a >> b >> c;

cout << "请输入三个浮点数:" << endl;;

cin >> m >> n >> q;

int m1 = Max1(a, b);

int m2 = Max1(a, b, c);

double m3 = Max1(m, n);

double m4 = Max1(m, n, q);

cout << "a与b之间较大的是" << m1 << endl;

cout << "a,b,c之间较大的是" << m2 << endl;

cout << "m,n之间较大的是" << m3 << endl;

cout << "m,n,q之间较大的是" << m4 << endl;

cin.get();

}

（运行结果）



3.（源代码）

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double x, y;

cout << "请输入两个数字" << endl;

cin >> x >> y;

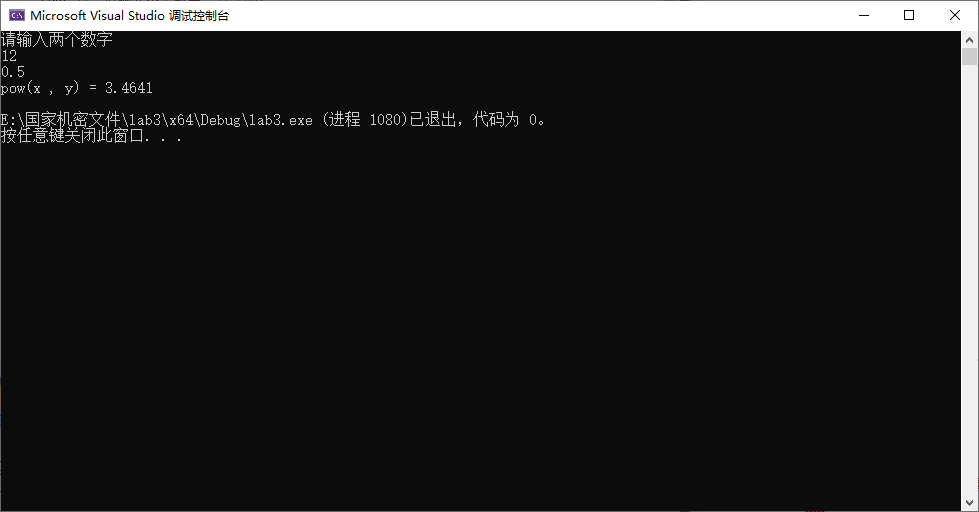
double result = pow(x, y);

cout << "pow(x , y) = " << result << endl;

cin.get();

}

（运行结果）



4.（源代码）

#include <iostream>

using namespace std;

int fib(int n) {

if (n == 1 || n == 2)

return 1;

else

return fib(n - 1) + fib(n - 2);

}

int main()

{

int n;

cout << "请输入要计算到的位数:";

cin >> n;

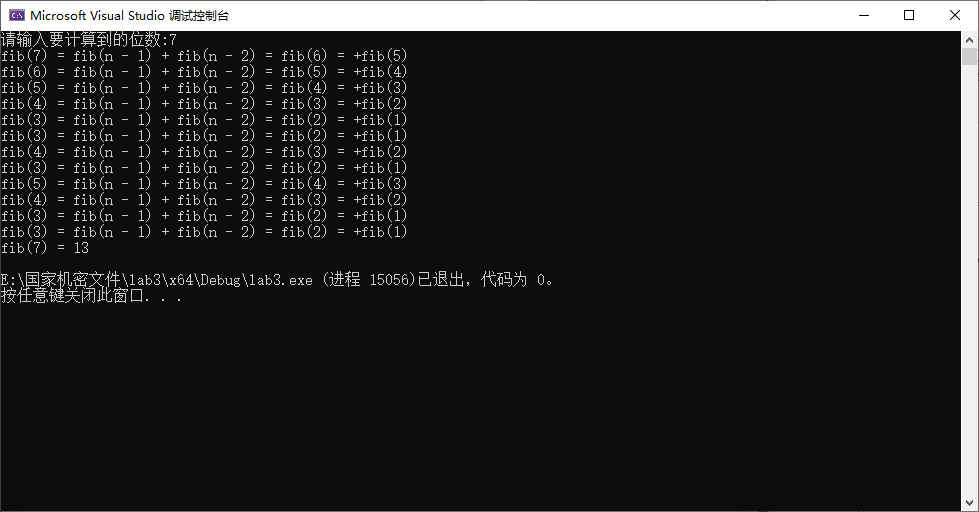
int result = fib(n);

cout << "fib(" << n << ") = " << result << endl;

cin.get();

}

（运行结果）



**四、分析与总结**

1.在实验一中，定义了函数f，实现了华氏度向摄氏度的转化；

2.在实验二中，定义了四个同名函数Max1，分别的实现了四种不同的运算，这是函数的重载；

3.在实验四中，采用函数递归的方法求出斐波那契数列的第n项的值，并对其进行了调试输出，可以清晰直观的看出程序的运行过程。

# 实验四 类与对象（两课时）

1. **实验目的**

1．掌握类的声明和使用。

2．掌握类的声明和对象的声明。

3．复习具有不同访问属性的成员的访问方式。

4．观察构造函数和析构函数的执行过程。

5．学习类的组合使用方法。

6. 复习自定义数据类型结构体的应用。

**二、实验内容**

1．声明一个CPU类，包含等级（rank）、频率（frequency）、电压（voltage）等属性，有两个公有成员函数run 、stop。其中，rank为枚举类型CPU-Rank，声明为enum CPU-Rank{P1=1,P2,P3,P4,P5,P6,P7}；frequency为单位是MHz的整型数，voltage为浮点型的电压值。观察构造函数和析构函数的调用顺序。

2．声明一个简单的Computer类，有数据成员芯片（cpu）、内存（ram）、光驱（cdrom）等，有两个公有成员函数run和stop。Cpu为CPU类的一个对象，ram为RAM类的一个对象，cdrom为CDROM类的一个对象，声明并实现这个类。

3．涉及一个用于人事管理的people（人员）类。考虑通用性，这里只抽象出所有类型人员都具有的属性：number（编号）、sex（性别）、birthdy（出生日期）、id（身份证号）等。其中“出生日期”声明为一个“日期”类内嵌子对象。用成员函数实现对人员信息的录入和显示。要求包括：构造函数和析构函数、拷贝构造函数、内联成员函数、组合。

4.声明一个表示时间的结构体，可以精确表示年、月、日、小时、分、秒；提示用户输入年、月、日、小时、分、秒的值，然后完整地显示出来。

**三、实验步骤**

1.（1）源代码

#include <iostream>

using namespace std;

enum CPU\_Rank { P1 = 1, P2, P3, P4, P5, P6, P7 };

class CPU {

private:

CPU\_Rank rank;

int frequency;

float voltage;

public:

CPU(CPU\_Rank r, int f, float v) {

rank = r;

frequency = f;

voltage = v;

cout << "构造了一个CPU函数" << endl;

}

void run() { cout << "Start running!" << endl; }

void stop() { cout << "Stop running!" << endl; }

~CPU() { cout << "析构了一个CPU!" << endl; }

CPU\_Rank GetRank() { return rank; }

int GetFerquency() { return frequency; }

float GetVoltage() { return voltage; }

void setRank(CPU\_Rank r) { rank = r; }

void setFrequency(int f) { frequency = f; }

void setVoltage(float v) { voltage = v; }

};

int main()

{

CPU a(P2, 300, 2.7);

a.run();

cout << "rank of this CPU is " << a.GetRank() << "\tFrequency of this CPU is "

<< a.GetRank() << "\tVoltage of this CPU is " << a.GetVoltage() << endl;

a.setRank(P1);

a.setFrequency(33);

a.setVoltage(8.908);

cout << "Exchange rank,frequency and voltage." << endl;

cout << "rank of this CPU is " << a.GetRank() << "\tFrequency of this CPU is "

<< a.GetRank() << "\tVoltage of this CPU is " << a.GetVoltage() << endl;

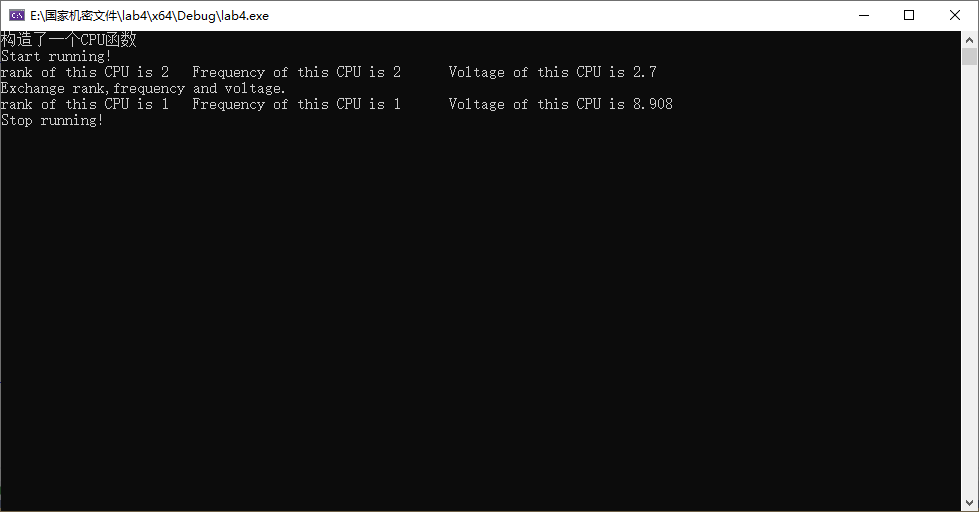
a.stop();

cin.get();

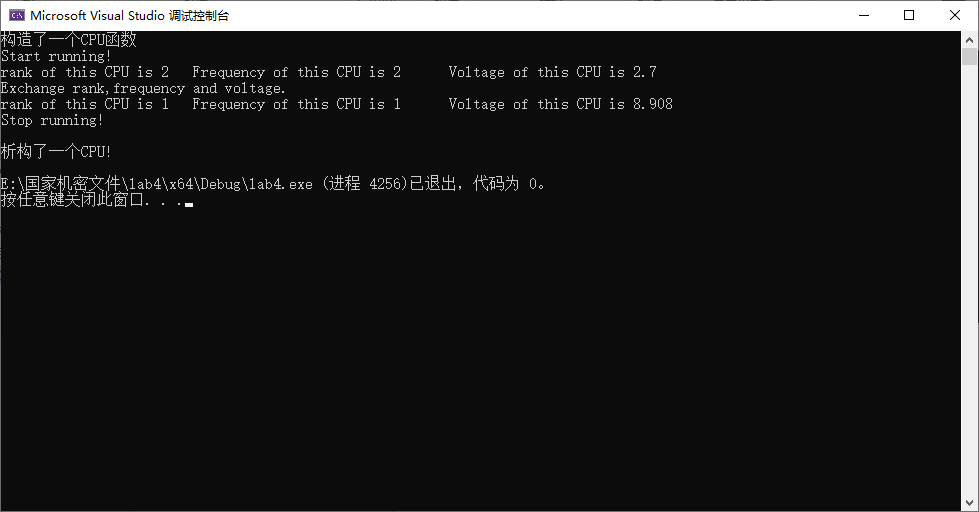
}

（2）运行结果

析构函数之前：



析构函数：



4.

（1）源代码

#include <iostream>

using namespace std;

struct Time {

int year;

int month;

int day;

int hour;

int minute;

int second;

};

int main()

{

int year\_in, month\_in, day\_in, hour\_in, minute\_in, second\_in;

cout << "依次输入年、月、日、时、分、秒" << endl;

cin >> year\_in >> month\_in >> day\_in >> hour\_in >> minute\_in >> second\_in;

struct Time clock = { year\_in, month\_in, day\_in , hour\_in , minute\_in , second\_in };

cout << clock.year << "年" << clock.month << "月" << clock.day << "日" << clock.hour << ":"

<< clock.minute << ";" << clock.second << endl;

cin.get();

}

（2）运行结果

