

21世纪高等教育计算机规划教材

COMPUTER

Visual C++ 面向对象 程序设计教程实验指导与习题集

The Answer and Practice of the
Visual C++ Programming Language

■ 郭有强 马金金 朱洪浩 编著

- 内容丰富，结构紧凑，选题典型
- 注重操作步骤和细节
- 使读者掌握面向对象思想和编程方法



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

Visual C++ 面向对象 程序设计教程实验指导与习题集

The Answer and Practice of the
Visual C++ Programming Language

■ 郭有强 马金金 朱洪浩 编著



人民邮电出版社

北京

Visual C++面向对象程序设计教程实验指导与习题集/
郭有强, 马金金, 朱洪浩编著. — 北京: 人民邮电出版社, 2019.10

21世纪高等教育计算机规划教材
ISBN 978-7-115-51671-8

I. ①V… II. ①郭… ②马… ③朱… III. ①C++语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第154342号

内 容 提 要

本书是《Visual C++面向对象程序设计教程》的配套实验指导及习题集。其中, 第1章包括11个基础实验, 基本覆盖了面向对象程序设计的主要知识点和 Visual C++程序设计相关内容; 第2章包括3个课程设计样例, 并提供了14个课程设计题; 第3章包括11套模拟试题及解答; 第4章是主教材习题参考答案; 附录包括 Visual C++ 6.0 程序调试方法和技巧、常见错误提示、标准 ASCII 码表、实验报告参考样本和课程设计报告参考样本等内容。

本书适合作为计算机及相关专业的“C++/VC++面向对象程序设计”课程的辅导教材, 也可供读者自学使用和参考。

-
- ◆ 编 著 郭有强 马金金 朱洪浩
责任编辑 李 召
责任印制 陈 鑫
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 14.25 2019年10月第1版
字数: 375千字 2019年10月北京第1次印刷
-

定价: 45.00 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

前言

“C++/VC++面向对象程序设计”是一门实践性很强的课程，上机实验和课程设计是该课程不可缺少的环节，其目的是使学生掌握面向对象思想和编程方法，进行面向对象程序设计的训练，从而巩固所学的知识，真正能够利用 Visual C++这个强有力的编程工具去解决实际问题。

本书是《Visual C++面向对象程序设计教程》的配套实验指导与习题集。本书内容丰富，结构紧凑，选题典型，注重操作步骤及细节。书中所有程序都在 Visual C++ 6.0 开发环境中测试通过。本书教学 PPT、例题源代码及慕课资源可在安徽省教育厅主管国家精品在线开放课程学习平台下载。

本书由郭有强、马金金、朱洪浩编著，郭有强统稿。在编写过程中，编者参考和引用了大量文献资料，在此，向被引用文献的作者表示诚挚的谢意，同时，向本书的读者表示真挚的感谢。

由于编者水平有限，书中难免存在不足，衷心希望得到有关专家和读者的批评和建议。您在使用本书时如遇到问题需要与作者商榷，或想索取其他相关资料，请与作者联系。联系方式：bbutguo@163.com。

编者
2019 年 8 月

目 录

第 1 章 基础实验	1	2.4.2 排课程序	91
实验一 程序流程控制与数组	1	2.4.3 图书馆管理系统	91
实验二 函数与指针	8	2.4.4 有理数运算	92
实验三 类和对象（一）	11	2.4.5 银行账户管理程序	92
实验四 类和对象（二）	16	2.4.6 水电气管理系统	93
实验五 继承与派生	20	2.4.7 小型公司工资管理系统	93
实验六 多态性与虚函数	24	2.4.8 课程设计选题管理系统	94
实验七 运算符重载与类模板	27	2.4.9 班级人事管理系统	94
实验八 I/O 流	31	2.4.10 自定义一个整数集合类	94
实验九 MFC 编程技术	34	2.4.11 商场货品管理	95
实验十 数据库编程	46	2.4.12 列车车次管理系统	95
实验十一 图形绘制	53	2.4.13 电话簿管理	95
第 2 章 课程设计	57	2.4.14 二维绘图系统	96
2.1 概述	57	第 3 章 模拟试题	97
2.2 总体要求	57	3.1 模拟试题（一）	97
2.3 课程设计样例	59	3.2 模拟试题（二）	105
2.3.1 课程设计（一）：复数类的设计		3.3 模拟试题（三）	114
和复数的运算	59	3.4 模拟试题（四）	121
2.3.2 课程设计（二）：用鼠标绘制		3.5 模拟试题（五）	130
曲线	64	3.6 模拟试题（六）	136
2.3.3 课程设计（三）：模拟时钟应用		3.7 模拟试题（七）	142
程序	82	3.8 模拟试题（八）	147
2.4 课程设计题目	90	3.9 模拟试题（九）	153
2.4.1 模拟计算器程序	90	3.10 模拟试题（十）	160
		3.11 模拟试题（十一）	167

第4章 主教材习题参考答案173

附录210

附录 A Visual C++6.0 程序调试方法和技巧210

附录 B Visual C++ 6.0 常见错误提示 213

附录 C 标准 ASCII 码表 218

附录 D 实验报告参考样本 219

附录 E 课程设计报告参考样本 220

参考文献 222

实验一 程序流程控制与数组

【实验目的】

1. 掌握 Visual C++6.0 集成开发环境开发程序的过程，熟悉常用的功能菜单命令，学习使用 Visual C++6.0 环境的帮助功能。
2. 学习并理解简单的 C++程序结构。
3. 掌握关系、逻辑运算符及其表达式的正确使用。
4. 掌握 if 语句和 switch 语句的使用。
5. 掌握 while、do-while、for 循环的语法结构、应用与区别。
6. 掌握数组的定义、赋值和输入输出的方法。
7. 掌握冒泡排序和选择排序算法，比较两者的不同点。

【实验准备】

1. 了解 C++程序的基本结构。
2. 了解简单的程序输入输出流。
3. 掌握分支结构和循环结构的使用。同时掌握 if 语句与 switch 语句的异同点，以及三种循环语句的区别。
4. 复习数组的定义方法、初始化及元素的引用。
5. 掌握冒泡排序和选择排序的算法思想。

【实验内容】

1. 初步熟悉 C++语言的上机开发环境，并编程输出：This is my first practice!。

(1) 启动集成开发环境。

选择菜单【开始 | 程序 | Microsoft Visual Studio 6.0 | Microsoft Visual C++ 6.0】，进入 Visual C++ 6.0 用户界面，如图 1.1 所示。

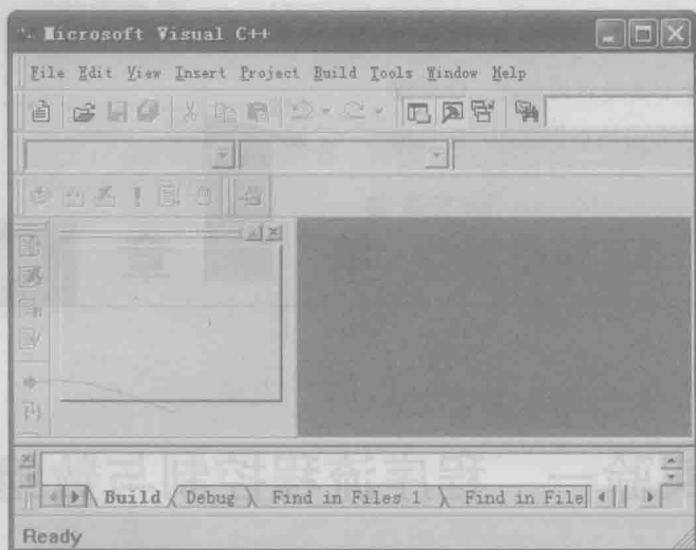


图 1.1 Visual C++ 6.0 用户界面

(2) 创建新工程。

选择【File】下拉菜单中的【New】菜单项，出现“New”对话框，选择“Projects”标签页面，如图 1.2 所示。

在列表栏中，选择“Win32 Console Application”（Win32 控制台应用程序）；在右侧的“Project name:”文本框中输入一个工程名，如“实验一”；在“Location:”文本框中指定该工程的保存路径，如图 1.2 所示，然后单击【OK】按钮。

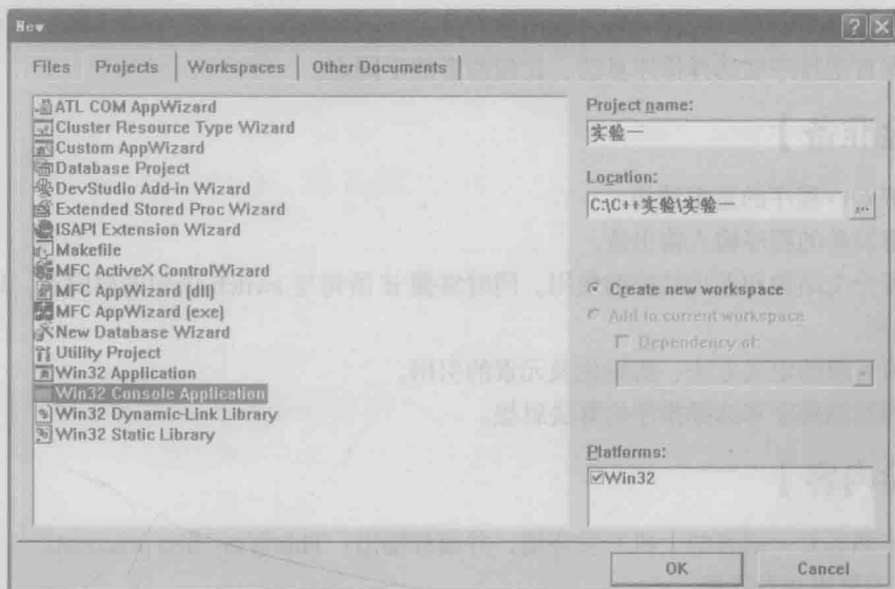


图 1.2 “New”对话框

在“Win32 Console Application-Step 1 of 1”对话框中选择“An empty project.”项。然后单击【Finish】按钮，如图 1.3 所示。

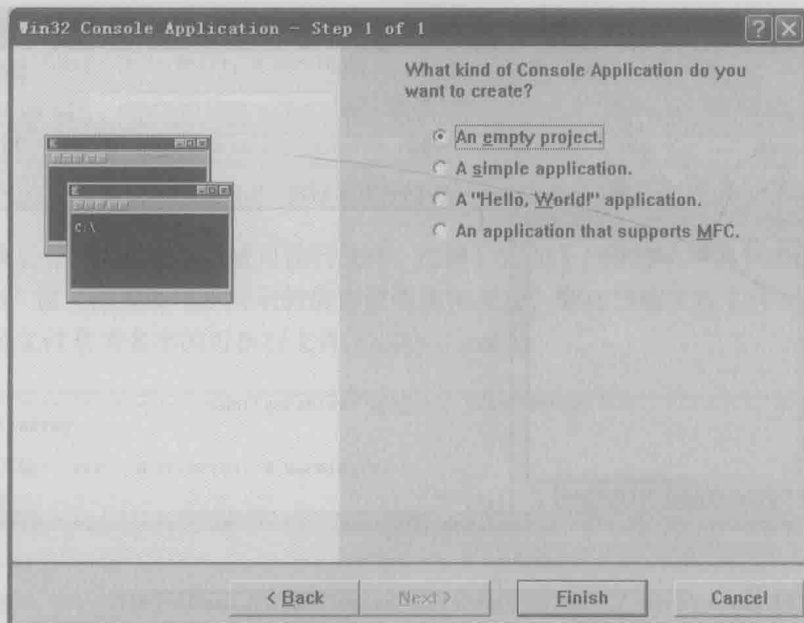


图 1.3 “Win32 Console Application-Step 1 of 1” 对话框

在“New Projects Information”对话框中单击【OK】按钮，完成工程创建。

(3) 创建 C++ 源程序文件。

选择【File】下拉菜单中的【New】菜单项，显示“New”对话框，选择“Files”标签页面，如图 1.4 所示。在列表栏中，选择“C++ Source File”，然后在右边的“File”文本框中输入源程序的文件名。

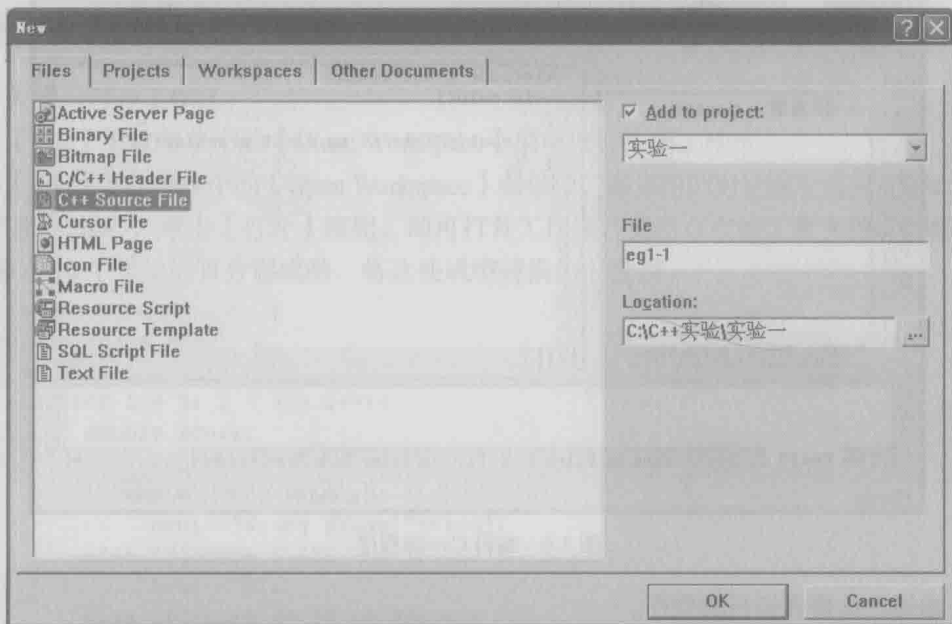


图 1.4 “New”对话框

单击【OK】按钮，源程序文件创建完成，并出现代码编辑窗口，如图 1.5 所示。

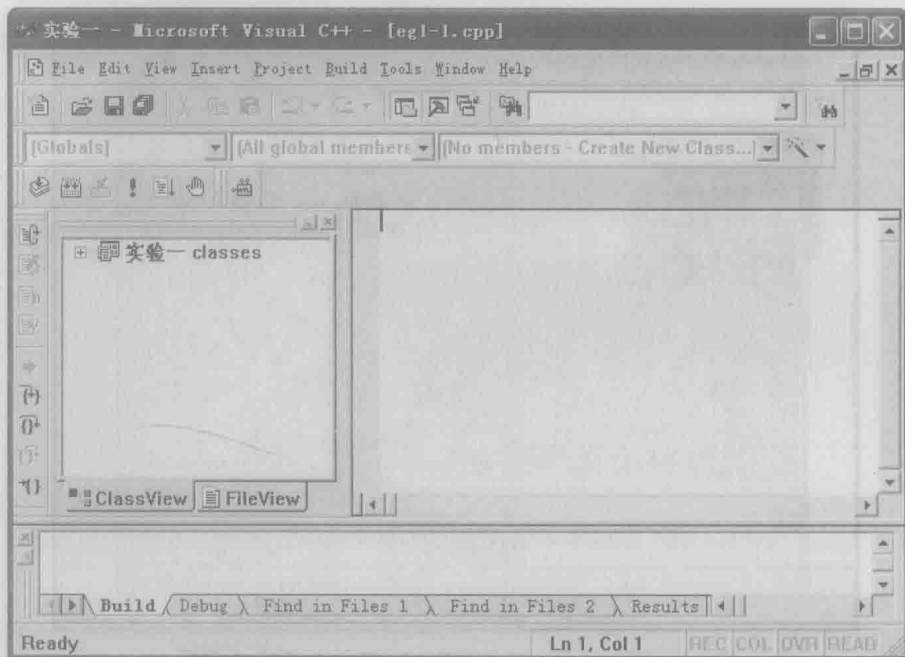


图 1.5 C++源程序代码编辑窗口

在代码编辑窗口中编辑 C++源程序，如图 1.6 所示。

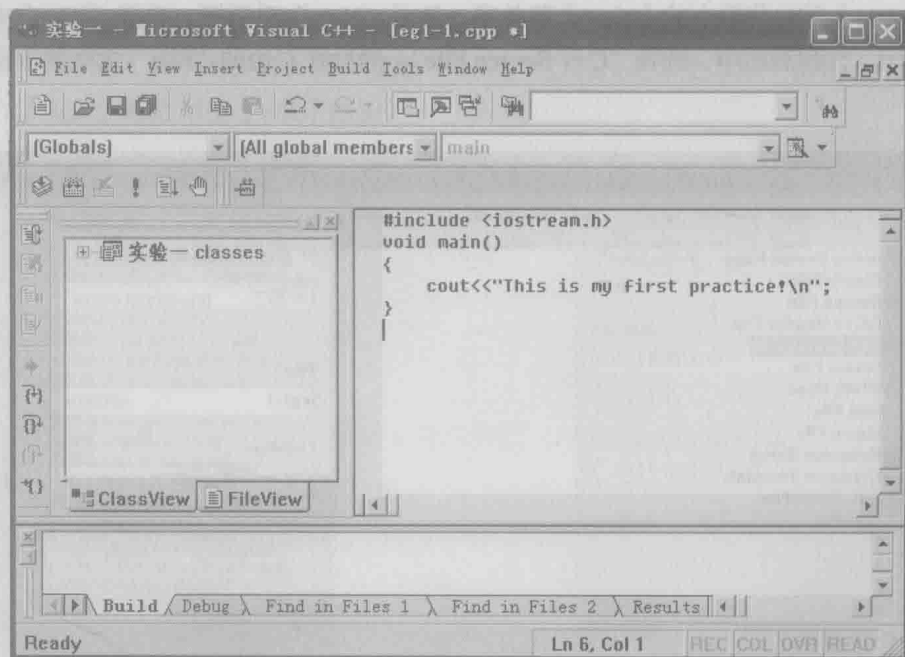


图 1.6 编辑 C++源程序

(4) 编译、连接和运行源程序。

选择【Build】下拉菜单中的【Compile egl-1.cpp】菜单项，系统开始对当前的源程序进行编译。编译过程中发现的错误信息显示在屏幕下方的“Build”窗口中。错误信息包含该错误所在行号和该错误的性质，用户可根据信息进行修改。编译上述源程序时的“Build”窗口如图 1.7 所示。

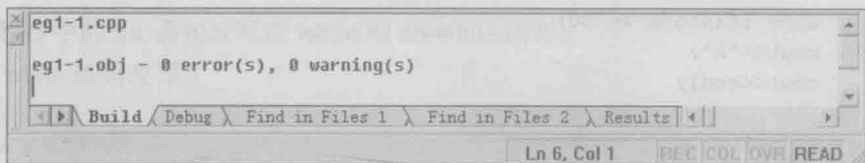


图 1.7 编译源程序时的“Build”窗口

编译无错误后,可进行连接生成可执行文件。选择【Build】下拉菜单中的【Build 实验一.exe】菜单项,“Build”窗口出现图 1.8 所示的信息说明连接成功,即在当前工程文件夹下的 Debug 文件夹中生成以源文件名为名字的可执行文件(实验一.exe)。

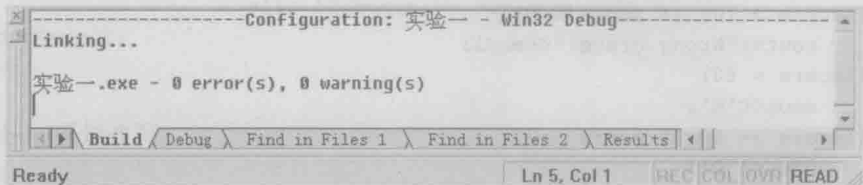


图 1.8 连接生成可执行文件

运行可执行文件:选择【Build】下拉菜单中的【Execute 实验一.exe】菜单项运行文件,结果显示在一个 DOS 窗口中,如图 1.9 所示。

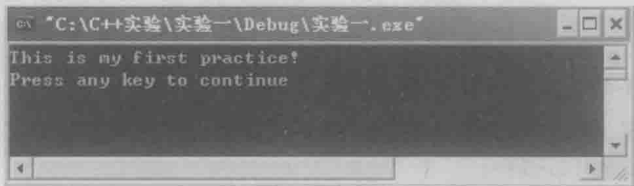


图 1.9 运行实验一.exe 的结果

(5) 关闭与打开工作区。

选择【File】下拉菜单中的【Close Workspace】菜单项,关闭工作区。

选择【File】下拉菜单中的【Open Workspace】菜单项,在弹出的对话框中选择“C:\C++实验\实验一\实验一.dsw”,单击【打开】按钮,即可打开工作区,对已存在的工程文件进行修改。

2. 输入 10 个学生的百分制成绩,将这些成绩转换为五分制。

```
#include<iostream>
using namespace std;
void main(){
    for(int i = 0; i < 10; i++){
        double score;
        cin>>score;
        if(score > 100 && score < 0)
            cout<<"Wrong grade!"<<endl;
        else if(score < 60)
            cout<<'E';
        else if(score >= 60 && score < 70)
            cout<<'D';
        else if(score >= 70 && score < 80)
            cout<<'C';
        else if(score >= 80 && score < 90)
            cout<<'B';
```

```

        else if(score >= 90)
            cout<<'A';
            cout<<endl;
    }
}

```

思考

(1) 将程序中的 if-else-if 语句改写成 if 语句。

答：程序可改写为

```

for(int i = 0; i < 10; i++){
    double score;
    cin>>score;
    if(score > 100 && score < 0)
        cout<<"Wrong grade!"<<endl;
    if(score < 60)
        cout<<'E';
    if(score >= 60 && score < 70)
        cout<<'D';
    if(score >= 70 && score < 80)
        cout<<'C';
    if(score >= 80 && score < 90)
        cout<<'B';
    if(score >= 90)
        cout<<'A';
    cout<<endl;
}

```

(2) 将程序中的 if-else-if 语句改写成 switch 语句。

答：程序可改写为

```

for(int i = 0; i < 10; i++) {
    double score;
    cin>>score;
    int c=(int)score/10;
    switch(c){
        case 10:
        case 9:
            cout<<'A';break;
        case 8:
            cout<<'B';break;
        case 7:
            cout<<'C';break;
        case 6:
            cout<<'D';break;
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0:
            cout<<'E';break;
        default:
            cout<<"Wrong grade!"<<endl;
    }
    cout<<endl;
}

```

(3) 将程序中的 for 语句改写成 while 和 do-while 语句。

答: 用 while 语句改写

```
int i=0;
while(i < 10){
    double score;
    cin>>score;
    if(score > 100 && score < 0) cout<<"Wrong grade!"<<endl;
    else if(score < 60) cout<<'E';
        else if(score >= 60 && score < 70) cout<<'D';
            else if(score >= 70 && score < 80) cout<<'C';
                else if(score >= 80 && score < 90) cout<<'B';
                    else if(score >= 90) cout<<'A';

    cout<<endl;
    i++;
}
```

用 do-while 语句改写

```
int i=0;
do{
    double score;
    cin>>score;
    if(score > 100 && score < 0) cout<<"Wrong grade!"<<endl;
    else if(score < 60) cout<<'E';
        else if(score >= 60 && score < 70) cout<<'D';
            else if(score >= 70 && score < 80) cout<<'C';
                else if(score >= 80 && score < 90) cout<<'B';
                    else if(score >= 90) cout<<'A';

    cout<<endl;
    i++;
} while(i < 10);
```

(4) 是否可以将程序中的 (score>=60 && score<70) 改写为 (score<70)? 为什么?

答: 可以。由于上一条语句已经判断了 score<60, 执行 else 语句时, 已经默认为 score<60 不成立即 score>=60, 因此可以改写。

(5) 是否可以将 for 语句中的 i++ 改写为 ++i? 为什么?

答: 可以。当 i++ 和 ++i 不与其他运算时, 二者等价。

3. 输入 10 个数, 按降序对其排序并统计其中大于平均值的数的个数。

```
#include<iostream>
using namespace std;
#define N 10
void main(){
    int i, j, t, a[N], count=0;
    float ave=0;
    cout<<"请输入 10 个数: ";
    for(i=0; i<N; i++)                //输入 10 个数
        cin>>a[i];
    for(i=0; i<N-1; i++)                //对数据排序
        for(j=i+1; j<N; j++)
            if(a[i]<a[j]) {
                t=a[i]; a[i]=a[j]; a[j]=t;
            }
    for(i=0; i<N; i++)                //求和
```

```

    ave=ave+a[i];
ave/=10;                                //求平均值
for(i=0; i<N; i++)                      //统计大于平均值的数的个数
    if(a[i]>ave) count++;
for(i=0; i<N; i++)
    cout<<a[i]<<" ";
cout<<endl<<" 大于平均值的数的个数: "<<count;
}

```

思考

(1) 将输入 10 个数的操作用数组初始化形式进行改写。

答: `int a[10]={88,55,22,77,44,66,11,1,99,33};`。

(2) 若改为按升序排序, 如何修改源程序?

答: 只需将 `if(a[i]<a[j])` 改写为 `if(a[i]>a[j])` 即可。

(3) 该程序采用的是哪种排序方法? 请使用另一种方法实现排序。

答: 该程序采用的是冒泡排序, 还可改写为选择排序。

```

for(i=0; i<N-1; i++){
    int max=i;
    for(j=i+1; j<N; j++)
        if(a[max]<a[j]) max=j;
    if (max!=i){
        t=a[i]; a[i]=a[max]; a[max]=t;
    }
}

```

(4) 将 `#define N 10` 用 `const` 进行改写。

答: `const N=10;`。

(5) 是否可以将 `#define N 10` 改为 `int N=10;`?

答: 不可以。C/C++ 在定义数组时不允许使用变量表示数组长度。

实验二 函数与指针

【实验目的】

1. 掌握函数定义的方法、函数间的参数传递。
2. 掌握重载函数的概念及函数参数在重载函数中的作用。
3. 理解函数模板和模板函数的概念, 并会编写简单的函数模板。
4. 掌握指针和多级指针的概念。
5. 理解并掌握指针作为函数参数的用法。

【实验准备】

1. 复习函数的概念、变量的作用域及函数调用过程中数据的传递方法等。
2. 理解为什么要引入重载函数机制, 了解函数参数在重载函数中的作用。
3. 熟悉函数模板的定义语法。

4. 复习和掌握指针的基本概念和用法。
5. 复习和掌握字符串指针、函数指针及指针函数的基本用法。

【实验内容】

1. 调试主教材【例 2-12】和【例 2-13】程序，思考引用和指针的区别。
2. 有 3 个学生各学 4 门课程，要求：
 - (1) 计算所有学生课程成绩的总平均分；
 - (2) 查找一门以上课程不及格的学生，并输出其各门课成绩。

```
#include<iostream>
using namespace std;
void main(){
    void average(float *p, int n);
    void search(float (*p)[4], int n);
    float score[3][4]={ {65,67,79,60},{80,27,90,81},{90,99,100,98}}; //初始化
    average(*score, 12);
    search(score, 3);
}
void average(float *p,int n){
    float *p_end, sum=0, aver;
    p_end=p+n-1; //p_end 指向当前行最后一个元素
    for(; p<=p_end; p++) //求所有课程成绩之和
        sum=sum+(*p);
    aver=sum/n;
    cout<<"average="<<aver<<endl;
}
void search(float (*p)[4], int n){
    int i, j, flag;
    for(i=0; i<n; i++){
        flag=0; //设标志变量
        for(j=0; j<4; j++) //查找不及格课程
            if(*(p+i)+j)<60) flag=1;
        if(flag==1) { //输出不及格学生的成绩信息
            printf("No.%d is fail,his scores are:\n",i+1);
            for(j=0; j<4; j++)
                printf("%6.1f ", *(p+i)+j));
            printf("\n");
        }
    }
}
```

思考

(1) 主函数中的语句

```
void average(float *p, int n);
void search(float (*p)[4], int n);
```

可否省略？为什么？

答：不可以省略。由于被调函数在主调函数之后，函数在调用时须做声明。

(2) 是否可以将 `average(*score, 12);` 中的 * 去掉?

答: 不可以。*score 表示传递给形参的是列指针, 使得函数 `average()` 中的形参 `p` 可以逐一访问二维数组中的每个元素。而如果去掉 *, 二者的指针类型将不匹配。

(3) 是否可以将 `void search(float (*p)[4], int n)` 中的 `float (*p)[4]` 改写为 `float *p`?

答: 不可以。`float (*p)[4]` 是指向一维数组的指针, 因此 `p+i` 是指向二维数组的第 `i` 行的行指针, `*(p+i)` 是指向二维数组的第 `i` 行的列指针, `*(p+i)+j` 是指向二维数组的第 `i` 行第 `j` 列元素的指针, 故后面可以采用 `*(*(p+i)+j)` 取得第 `i` 行第 `j` 列元素的值。如果改写为 `float *p`, 则指针 `p` 直接指向二维数组的第一个元素, `p+1` 指向第二个元素, 以此类推, 后面的程序将出现访问错误。

(4) 是否可以将 `void search(float (*p)[4], int n)` 中的 `float (*p)[4]` 改写为 `float *p[4]`?

答: 不可以。`float *p[4]` 表示 `p` 是一个指针数组。函数调用时会出现参数类型不匹配错误。

(5) `*(*(p+i)+j)` 是否可以改写为 `p[i][j]`?

答: 可以。二者是等价的, C/C++ 中对 `p[i][j]` 的访问会自动转换为对 `*(*(p+i)+j)` 的访问。

3. 求给定数组中的最大值。

程序一

```
#include<iostream>
using namespace std;
int getmax(int intArr[],int len);
double getmax(double douArr[],int len);
int main(){
    int arr1[]={25,1,86,45,365,55,8};           //数组初始化
    int len1=sizeof(arr1) / sizeof(int);         //求数组长度
    double arr2[]={253.55,12.2,8895.12,4.9};     //数组初始化
    int len2=sizeof(arr2) / sizeof(double);      //求数组长度

    cout<<"Int Array MaxValue is: "<<getmax(arr1,len1)<<endl;
    cout<<"Double Array MaxValue is:"<<getmax(arr2,len2)<<endl;
    return 0;
}

int getmax(int intArr[], int len) {              //求最大值
    int result=intArr[0];
    for(int i=1; i<len; i++)
        if (result<intArr[i])
            result=intArr[i];
    return result;
}

double getmax(double douArr[],int len) {         //求最大值
    double result=douArr[0];
    for(int i=1; i<len; i++)
        if (result<douArr[i])
            result=douArr[i];
    return result;
}
```

程序二

```
#include<iostream>
using namespace std;
template<typename T>
```

```

T getMax(T myarr[],int len) { //求最大值
    T result=myarr[0];
    for(int i=0; i<len; i++)
        if (result<myarr[i])
            result=myarr[i];
    return result;
};

int main(){
    int arr1[]={25,1,86,45,365,55,8}; //数组初始化
    int len1=sizeof (arr1) / sizeof(int); //求数组长度
    double arr2[]={253.55,12.2,8895.12,4.9}; //数组初始化
    int len2=sizeof (arr2) / sizeof(double); //求数组长度
    cout<<"Int Array MaxValue is: "<<getmax(arr1,len1)
        <<endl;
    cout<<"Double Array MaxValue is:"<<getmax(arr2,len2)
        <<endl;
    return 0;
}

```

思考

(1) 请说出程序一和程序二的区别。

答：程序一采用的是重载函数，程序二采用的是模板函数。程序一只能处理 int 和 double 类型数组求最大值的操作，而程序二通过模板函数可以处理任意类型数组求最大值的操作。

(2) 在程序一中，是否可以任意修改主函数中数组 arr1 和 arr2 中的初始化元素个数？

答：可以。因为程序中已经通过 sizeof (数组名) / sizeof (类型) 求出了数组的元素个数，并传递给求最大值函数，所以函数的执行不受影响。

(3) 在程序一中，如果将主函数的数组 arr2 的类型改为 float，程序是否能够运行？为什么？

答：将数组 arr2 的类型改为 float，程序不能运行，会出现类型不匹配的错误。

(4) 在程序二中，是否可以将 template<typename T> 中的 typename 替换为 class？

答：可以。在模板函数中二者作用相同。

(5) 在程序二中，如果将主函数的数组 arr2 的类型改为 float，程序是否能够运行？结果是否正确？为什么？

答：将数组 arr2 的类型改为 float，程序能够运行，但结果错误。由于模板函数会根据传递的实参自动替换类型，因而程序可以运行。但是要得到正确结果需要将程序语句 int len2=sizeof (arr2) / sizeof(double); 中的 double 改为 float。

实验三 类和对象（一）

【实验目的】

1. 掌握类的定义方法。
2. 掌握类成员函数的定义方法。
3. 掌握类对象的定义及其访问方式。