

· 高等学校计算机基础教材精选 ·

C++程序设计实验、 辅导与习题解答

刘加海 朱云芳 编著

· 高等学校计算机基础教育教材精选 ·

C++程序设计实验、 辅导与习题解答

刘加海 朱云芳 编著

清华大学出版社
北京

出版说明

——高等学校计算机基础教育教材精选——

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,面向各高校开设的计算机必修课、选修课,以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本,出版一本,并保持不断更新),坚持宁缺毋滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是:jiaoh@tup.tsinghua.edu.cn;联系人:焦虹。

清华大学出版社

前言

C++ 程序设计实验、辅导与习题解答

目前,高校理工科各专业都将 C 或 C++ 程序设计作为首选的程序设计课程。C++ 程序设计语言具有丰富的数据结构、灵活的控制语句、清晰的程序结构、良好的可移植性、面向对象的属性,并且易于开发与维护大型程序。

本书是作者根据多年教学与科研经验,配合清华大学出版社出版、由浙江大学刘加海等主编的《C++ 程序设计》编写的配套用书,主要包括 C++ 程序设计实验、C++ 程序设计要点与分析、习题解答及补充习题。

书中给出了一学期的实验内容,覆盖了 C++ 的主要编程思想。实验编排由浅入深,将面向对象的思想贯通于整个过程。这些实验将会有效提高学生 C++ 程序设计的能力,减轻教师实验教学的工作负担。

通过 C++ 实验题的训练,能有效提高学生程序设计的能力;通过 C++ 重点内容的学习,能有效帮助学生加深对 C++ 编程思想的理解,掌握编程方法;通过 C++ 习题的解答,可迅速提高学生的编程能力;通过补充习题启发,可扩大学生 C++ 的应用视野。

本书可作为高等学校理工科专业学生 C++ 程序设计课程的实验教材或教学辅导书,也可供 C++ 程序设计爱好者参考。

在这里,作者对给予大力支持的浙江大学计算机学院表示衷心感谢。本书由浙江大学城市学院刘加海、严冰,浙江工商大学朱云芳,浙江商业职业技术学院孔美云编写。

由于编者水平有限,不足之处在所难免,敬请各位专家和广大读者批评指正。如需教学课件、源程序可发电子邮件到 Ljhqyyq@yahoo.com.cn。

刘加海

目录

C++ 程序设计实验、辅导与习题解答

第 1 部分 C++ 程序设计实验

实验 1 熟悉上机环境及 C++ 基础实验	3
实验 2 C++ 文件组织与 C++ 运算符的应用	8
实验 3 C++ 循环程序设计	12
实验 4 C++ 分支程序设计	16
实验 5 函数设计与应用(1)	19
实验 6 函数设计与应用(2)	23
实验 7 数组的应用(1)	27
实验 8 数组的应用(2)	31
实验 9 指针在数组中的应用	33
实验 10 指针与函数	36
实验 11 类与对象(1)	41
实验 12 类与对象(2)	44
实验 13 运算符重载	47
实验 14 类继承与多态性	53
实验 15 模板和异常处理	58
实验 16 I/O 与文件操作	64
实验 17 可视化程序设计初步	67

第 2 部分 C++ 程序设计要点与分析

第 1 章 C++ 程序设计入门	73
1.1 本章要点	73
1.2 习题参考解答	77

第2章 C++程序的文件组织与基本运算符	80
2.1 本章要点	80
2.2 习题参考解答	82
2.3 补充习题阅读	88
第3章 循环程序设计	89
3.1 本章要点	89
3.2 习题参考解答	93
3.3 补充习题阅读	99
第4章 分支程序设计	102
4.1 本章要点	102
4.2 习题参考解答	106
4.3 补充习题阅读	112
第5章 函数	121
5.1 本章要点	121
5.2 习题参考解答	124
5.3 补充习题阅读	132
第6章 指针与数组	140
6.1 本章要点	140
6.2 习题参考解答	145
6.3 补充习题阅读	150
第7章 指针与函数	161
7.1 本章要点	161
7.2 习题参考解答	163
7.3 补充习题阅读	166
第8章 类及其应用	169
8.1 本章要点	169
8.2 习题参考解答	172
8.3 补充习题阅读	185
第9章 运算符重载	203
9.1 本章要点	203
9.1 习题参考解答	207
9.3 补充习题阅读	214
第10章 继承与多态性	224
10.1 本章要点	224

10.2	习题参考解答.....	226
10.3	补充习题阅读.....	246
第 11 章	模板和异常处理	256
11.1	本章要点.....	256
11.2	习题参考解答.....	259
第 12 章	I/O 流与文件	267
12.1	本章要点.....	267
12.2	习题参考解答.....	273
第 13 章	可视化程序设计初步	278
13.1	本章要点.....	278
13.2	习题参考解答.....	279

第 1 部分 C++ 程序设计实验

C++语言程序设计是一门实践性很强的程序设计基础课,该课程对提高学生的逻辑分析、抽象思维、程序设计能力具有重要意义。上机实践是学好本课程十分重要的环节。通过上机实践,将有利于进一步巩固和加深对本课程基本概念、基本知识的理解和掌握,同时,也可后续相关课程的学习打下必要的基础。

上机操作时,主要完成源程序的编辑、编译、链接和调试运行。上机过程中检查程序的算法、语法、链接与运行是否正确过程称为调试过程。上机的各个环节都有可能碰到不少问题,也许不会一次成功,针对问题应细心查找原因,逐个解决。这也是锻炼上机程序调试能力的好机会,碰到困难时不要轻易放弃。学习 C++程序设计的最好方法就是多上机、多实践,程序是调试出来的,不是“编”出来的。此部分共安排了 17 个实验。

说明:每个实验分为实验目的和要求、实验内容两部分。

实验 1 熟悉上机环境及C++ 基础实验

1. 实验目的和要求

(1) 熟悉上机环境,了解 Visual C++ 6.0 集成开发环境、掌握源程序编辑、程序调试、查看变量、程序运行的方法。

(2) 熟悉 C++ 的程序结构,掌握 main 函数、保留字、变量及变量定义、输入与输出流等概念。

(3) 熟悉类与类对象的定义、类成员的调用方法。

2. 实验内容

任务 1: 程序调试

(1) 熟悉 Visual C++ 开发环境

① 从 Windows 的“开始”菜单,选择“程序”→Microsoft Visual Studio→Microsoft Visual C++ 6.0,用鼠标左键单击 Microsoft Visual C++ 6.0,进入 Developer Studio 的应用程序集成开发环境。

② 用鼠标左键单击 File→“新建”命令,打开“新建”对话框。在“新建”对话框中单击“文件”,选择 C++ Source File,并且在右边“文件”文本框中输入文件名 file1,单击命令按钮“确定”。

③ 在编辑区光标处开始输入源程序。

```
/* 程序功能:输入输出练习 */
#include<iostream.h>
int main()
{
    cout<<"您好!浙江大学欢迎您,愿您喜欢 C++ 的输入输出。"; //表示输出一个字符串
    cout<<2008; //打印一个整数
    cout<<"\n"; //换行
    cout<<20.1; //打印一个实数
    cout<<endl; //换行
    cout<<"I am "<<20<<" years old student."; //连续打印
    char name[30]; //变量的声明位置与 C 有什么区别吗?
    int age;
    cout<<"please give your name :";
    cin>>name; //表示键盘输入字符串到变量 name 中
    cout<<"please tell me how old are you?";
```

```

cin>>age; //表示键盘输入整型数到变量 age 中
cout<<"Your name is "<<name<<endl;
cout<<"you are "<<age<<"years old.";
}

```

④ 单击菜单“编译”→“编译 file1.cpp”，建立一个默认的工作区。编译成功，产生一个目标文件 file1.obj。如果程序有错误，在输出窗口中会指出每一项的错误及产生错误的原因、所在行。请思考：在程序中任何一行少一个“;”符号，有什么提示？删除“int age;”这一行，编译后有什么错误提示？

⑤ 单击菜单“编译”→“执行 file1.exe Ctrl+F5”，单击命令按钮“是(Y)”，将产生一个可执行文件 file1.exe。

⑥ 编辑第二个C++源程序。单击“文件”→“新建”，在“新建”对话框中选择C++ Source File，在文件框中输入文件名 file2。

⑦ 单击菜单“文件”→“关闭工作区”，关闭前一题的工作空间，重新编译即可。

⑧ 打开已存在的文件。在 Developer Studio 中一次只能打开一个工作空间。工作空间的打开过程为“文件”→“打开工作区”，在对话框中选择要打开的工程(.dsw)文件。

问题：• C++中的注解有哪些？

- 应用输入输出流应包含什么库？
- 定义变量有什么含义？
- 常用变量有哪些基本类型？

(2) C++程序的调试

编辑程序后存在一些错误是不可避免的，这些错误的发生有可能是：编译错误，如语法、输入错误等；链接错误，如函数名写错或所调用的函数没有定义；运行错误等。

与位置有关的断点设置：

① 最简单的是设置一般位置断点。只要把光标移到要设断点的位置，当然这一行必须包含一条有效语句；然后按 F9 键(F9 键为开关键)，会看到在这一行的左边出现一个红色的圆点。

② 表达式满足一定的条件。要设置这种断点需单击“编辑”→“断点...”命令，这时 Breakpoint 对话框将会出现(或按组合键 Alt+F9)。选中 Breakpoint 对话框中的 Location 标签，使 Location 页面弹出。输入行号或单击 condition 按钮，弹出“断点条件”对话框。在 Expression 编辑框中写出逻辑表达式，如 age==18 或 age>20，最后单击 OK 返回。

(3) 控制程序的运行

① 单击“编译”→“开始调试”→“去”命令，程序开始运行在 Debug 状态下(此时菜单“编译”改变为 Debug)。程序会由于断点而停顿下来，这时会看到有一个小箭头，它指向即将执行的代码。

② 四条命令：Step Over, Step Into, Step Out, Run to Cursor。

Step Over：运行当前箭头指向的代码(只运行一条代码)。

Step Into: 如果当前箭头所指的代码是一个函数的调用,则用 Step Into 进入该函数进行单步执行。

Step Out: 如当前箭头所指向的代码在某一函数内,则用它使程序运行至函数返回处。

Run to Cursor: 使程序运行至光标所指的代码处。

任务 2: 程序调试

在实验 1 的源程序基础上,调试以下程序,并回答相关问题。

(1) 用单行注释符“//”注解预处理命令,编译程序有什么提示? 如果程序中使用函数 fabs,应该增加什么样的预处理?

(2) 把 int main() 写成 int Main(),编译程序有什么提示? C++ 程序设计对大小写有什么要求?

(3) 如果把语句 cout<<2008; 改写为: cout<<2008 编译程序有什么提示? C++ 程序设计对语句结束有什么要求?

(4) 程序中的变量定义: int age; 如改写为: int age_1; 编译程序有什么提示? 如改写为: int l_age; 编译程序有什么提示? 如改写为: int for; 编译程序有什么提示? 关于变量的取名能总结出什么经验?

任务 3: 程序设计

仿照任务 1,应用 cin、cout 进行输入、输出的程序设计。给出提示“请输入您的学院名”,输入所在的学院;按照输出提示“请输入您的姓名”,输入姓名;然后输出提示“请输入您的年龄”。输入年龄后,输出“欢迎来自???学院的???同学,您的年龄是??岁”,其中“?”为输入的值。

任务 4: 程序调试

进入 Visual C++ 集成环境,编辑下列源程序:

```
#include<iostream.h>
class s
{
private:
    int x,y;
public:
    void s(int a,int b){x=a;y=b;}
    int area(){return x*y;}
    void show()
    {
        cout<<"面积:"<<area()<<endl;
    }
};

int main()
{
```

```

int a,b;
cin>>a>>b;
s x(a,b);
x.show();
}

```

编译运行程序后,写出程序的运行结果并回答下列问题:

- (1) 省略预处理行 `#include<iostream.h>`, 会出现什么样的错误提示?
- (2) 程序中出现了哪些关键字?
- (3) 函数 `init`、`area`、`show` 的功能分别是什么?
- (4) 在类中, 如果不写 `private`、`public`, 会出现什么样的错误提示? 为什么?
- (5) 重写程序, 把成员函数定义在类的外部。提示: 请参照下列类定义。

```

class s
{
private:
    int x,y;
public:
    void init(int a,int b);
    int area();
    void show();
};

void s::init(int a,int b)
{
    x=a;
    y=b;
}

int s::area()
{
    return x*y;
}

void s::show()
{
    cout<<"面积:"<<area()<<endl;
}

```

任务 5: 程序设计

模仿任务 4, 设计一个立方体类, 它有三个 `int` 型私有成员 `a`、`b`、`c`。求出这个立方体的体积, 然后输出。

类定义为:

```

class cub
{

```

```

public:
    cub(int x,int y,int z){a=x;b=y;c=z;}
    int Volume()
    {
        int z;
        z=a*b*c;
        return z;
    }
    void show()
    {
        cout<<"立方体体积为："<<Volume()<<endl;
    }
private:
    int a,b,c;
};

```

请参照任务 4 的 main 函数。

任务 6：程序设计

首先在屏幕上输出以下菜单：

1. 求函数值
2. 计算方程的根
3. 画简单图形
0. 退出

请输入您的选择：2

您的选择是：2

其中第一个 2 是键盘输入的，第二个 2 是输出所输入的变量值。

实 2

验 C++ 文件组织与C++ 运算符的应用

1. 实验目的和要求

- (1) 熟悉C++的多文件组织(一个C++程序由多个文件构成)。
- (2) 掌握C++中常量与变量的应用,并掌握①读入不同进位制常量,用 dec、oct、hex 等控制输出;②各种算术、关系运算符的应用,包括数学函数的应用。
- (3) 熟悉类与类对象的定义、类成员的调用。

2. 实验内容

任务 1: C++多文件程序调试

掌握C++多文件组织、类对象与类成员的使用。理解一个C++程序可以由一个或多个文件构成,一个文件可以由一个或多个函数构成。下列程序通过简单的函数调用,包括一些基本运算,主要目的是使读者掌握多文件系统的编辑、编译、运行过程。此C++程序由五个文件构成。

(1) 单击“开始”→“程序”→Microsoft Visual Studio→Microsoft Visual C++ 6.0→“新建”→“文件”→C++ Source File,在文件对话框中输入文件名 2-1-1,编辑 2-1-1.cpp 内容:

```
int add(int x,int y)
{
    return x+y;
}
```

保存并关闭此文件。2-1-1.cpp 中只有一个求和的函数。

(2) 单击“文件”→“新建”→“文件”→C/C++ Header File,在文件对话框中输入文件名 2-1,编辑 2-1.h 内容:

```
#include<iostream.h>
#include "2-1-1.cpp"
```

保存并关闭此文件。2-1.h 为头文件。

(3) 单击“文件”→“新建”→“文件”→C++ Source File,在文件对话框中输入文件名 2-1,编辑 2-1.cpp 内容:

```
int main()
{
    int a,b;
    cin>>a>>b;
```

```
int c=add(a,b);
cout<<a<<'+'<<b<<'='<<c<<endl;
}
```

(4) 编译程序,有什么提示? 请思考是什么原因。

(5) 在文件 2-1.cpp 的第一行加入预处理命令:

```
#include "2-1.h"
```

(6) 再次编译,思考一下程序为什么成功了。

注意: 编译时当前文件应是 2-1.cpp。

任务 2: 多文件程序设计

(1) 仿照任务 1,其中主调函数的文件名为 2-2.cpp,内容为:

```
#include "2-2.h"
int main()
{
    int a,b;
    cin>>a>>b;
    int c1=f1(a,b);
    cout<<a<<'+'<<b<<'='<<c1<<endl;
    int c2=f2(a,b);
    cout<<a<<'-'<<b<<'='<<c2<<endl;
    int c3=f3(a,b);
    cout<<a<<'*'<<b<<'='<<c3<<endl;
    int c4=f4(a,b);
    cout<<a<<'/'<<b<<'='<<c4<<endl;
}
```

(2) 分别在文件 2-2-1.cpp、2-2-2.cpp、2-2-3.cpp、2-2-4.cpp 中编写函数 f1、f2、f3、f4,完成算术运算+、-、*、/的功能。

(3) 编辑头文件 2-2.h:

```
#include <iostream.h>
#include "2-2-1.cpp"
#include "2-2-2.cpp"
#include "2-2-3.cpp"
#include "2-2-4.cpp"
```

(4) 编译并运行程序。

注意: 编译时当前窗口文件是 2-2.cpp。

任务 3: 输入、输出操纵符控制程序设计

(1) 编辑并调试下列程序,程序的文件名为 2-3.cpp。

```
#include<iostream.h>
```



```
#include<iomanip.h>
int main()
{
    int x,y;
    cin>>x>>y;
    cout<<"x="<<x<<" "<<"y="<<y<<endl;
    cout<<"x="<<oct<<x<<" "<<"y="<<hex<<y<<endl;
    cout<<"x="<<setw(5)<<x<<" "<<"y="<<y<<endl;
}
```

(2) 常用进位制操纵算子有 dec、oct、hex,它们在程序中分别起什么作用?在上述程序中 oct 的位置分别用 dec、hex 去代替,输出结果分别是什么?

(3) setw(5)起什么作用?请改变其中的参数 5,观察程序的运行结果有什么变化。

(4) 设计一个程序,输入 20 给 x,输出结果为:

x=20 x=14

x=20 x=24

注意: 程序设计中不能对 x 作运算,只能使用 dec、oct、hex、setw 进行控制。

任务 4: 程序设计,运算符的应用

阅读程序,写出下列程序的运行结果,设程序的文件名为 2-4.cpp。

```
#include<iostream.h>
class data
{
private:
    int x,y;
public:
    data(int a,int b)           //data 为构造函数
    {
        x=a;y=b;
    }
    void gt()
    {
        cout<<x<<'>'<<y<<'='<<(x>y)<<endl;
    }
    void print()
    {
        gt();
    }
};

int main()
{
    int a,b;
```