

C++

程序设计简明教程

○ 刘甲耀 严桂兰 编著

⌚ 类成员与对象的构建

⌚ 运算符重载

⌚ 核心技术

⌚ 输入/输出流的使用

⌚ 派生类的构建



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

C++

程序设计简明教程

—— 第2版 ——



清华大学出版社
Tsinghua University Press
http://www.tup.tsinghua.edu.cn

C++程序设计简明教程

刘甲耀 严桂兰 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是为已学过 C 语言的读者进一步学习 C++ 语言编程而编写的。全书高度浓缩了 C++ 面向对象程序设计的方法与技巧, 内容涉及: C++ 的新特征; C++ 编程的核心技术; 类成员与对象的构建; 派生类的构建; 运算符重载; 输入/输出流的使用; 模板; 异常处理。书中各章均有小结与习题, 书末附录给出了参考答案、上机步骤及 C++ 与 SQL 数据库连接技术。书中示例与习题约为 170 题, 均在 Microsoft Visual C++ 环境中通过, 其示例、习题与运行结果可通过华信教育资源网 (<http://www.hxedu.com.cn>) 免费下载使用。

本书可作为大专院校计算机与其他类专业及培训班教材, 也可供各行各业从事计算机工作的人员使用。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

C++ 程序设计简明教程 / 刘甲耀, 严桂兰编著. — 北京: 电子工业出版社, 2004.8

ISBN 7-121-00143-8

I. C… II. ①刘… ②严… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 071147 号

责任编辑: 龚立革

印 刷: 北京京科印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 19 字数: 486 千字

印 次: 2004 年 8 月第 1 次印刷

印 数: 4 000 册 定价: 26.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系。联系电话: (010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

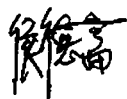
序

刘甲耀教授原任教于国立华侨大学，长期从事计算机教学与科研工作，取得了优异的成绩，有不少著述面世。因此，他享受国务院政府特殊津贴。2002 年受聘来我院工作，继续从事计算机教学与科研，同样，也取得了可喜的成果。现由电子工业出版社出版的《C++程序设计简明教程》、《Core Java 应用程序设计教程》，就是刘教授在私立华联学院任教的教学与科研总结，具有很大的实用价值。该书既适合我院学生使用，也将满足社会多种类型的学校、培训班及读者自学的需求。

我院始成立于 1990 年，1994 年经广东省人民政府批准，教育部备案，正式成立私立华联学院，是具有高等学历教育招生资格的省属普通高等学校。现学院有校本部和龙洞、清远两个校区。校本部占地 400 多亩，校舍建筑面积 13 万多平方米。设有计算机科学技术系，计算机软工工程系，电子信息工程系，工商管理系，会计系，英语系，日语系，美术设计系，音乐系，美容与造型艺术系，法律系，服装设计与工艺系，电视编导系，体育服务与管理系等 14 个系，31 个专业。在校生 5800 多名，在未来两年内将达到万人的规模。各类实验室 30 多个，教学设备固定资产 3000 多万元。

学院十分重视师资队伍的建设，并已形成一支以离退休教授、副教授为主体的高层次及有志于教育事业的年轻人的教师队伍。汇集于我校的这些高级人才，大部分原执教于华南理工大学、暨南大学、华南师范大学、中山大学、外语外贸大学等省内外名校。我校现有任课教师 285 人（其中外籍教师 12 人），教授、副教授占 63%，70% 以上的专业课由教授、副教授主讲。他们教学认真，对学生热情负责，边教学，边从事教材建设，取得了教学、科研双丰收。刘甲耀教授出版此书，正是我院教学、科研双丰收的成果之一，特此祝贺。

私立华联院校长：



2004 年 5 月

前 言

C++是一门面向对象程序设计语言，能编写各式各样的应用软件。为使读者在学习 C 语言程序设计的基础上，能进一步进入 C++编程领域，特编写《C++程序设计简明教程》一书。它高度浓缩了 C++编程的核心技术，讲述了 C++编程的方法与技巧，以满足读者的需要。

本书经过作者多年教学经验的积累，并在实践中不断完善与创新，形成了自己特有的体系。因而，它是一本取材广泛，内容由浅入深，重点、难点分明，易学、易掌握且方法与示例并举的简明书籍。内容涉及：C++的新特征；C++编程的核心技术；类成员与对象的构建；派生类的构建；运算符重载；输入/输出流的使用；模板；异常处理。书中各章均有小结与习题，书末附录给出了参考答案、上机步骤及 C++与 SQL 数据库连接技术。书中所有示例均在 Microsoft Visual C++ 6.0 环境中运行通过，许多例子采用多种解决方案，充分体现了 C++编程的灵活性、多样性、实用性与趣味性。为使读者阅读方便，特在重点、难点处加以注释。书中示例、习题与运行结果可通过华信教育资源网（<http://www.hxedu.com.cn>）下载使用。

本书编写过程中，得到荣获“广东省首届十佳民办院校”称号的私立华联学院校长侯德富教授等有关领导的大力支持。谨此，我们将此书献给私立华联学院，衷心祝愿私立华联学院越办越好，尽快地跨入国内乃至国际知名大学的行列！同时也感谢林汶彬、王少莉、梁少兴等同志热心参加了本书部分录入工作。

本书不足之处，敬请读者指正，作者 E-mail 地址：ygl0501@sina.com.cn。

作 者
2004 年 6 月

目 录

第 1 章 C++的新特征	1
1.1 C++的输入/输出	1
1.1.1 引例	1
1.1.2 说明	2
1.2 C++的注释方式	2
1.2.1 单行注释	2
1.2.2 多行注释	2
1.3 C++的符号常量定义	3
1.3.1 定义符号常量的形式	3
1.3.2 注意事项	3
1.3.3 C 与 C++的比较	3
1.3.4 应用示例	4
1.4 C++的函数原型	9
1.5 C++函数的默认参数	9
1.6 C++的 new 与 delete 运算符	10
1.6.1 使用形式	10
1.6.2 应用示例	10
1.7 C++的内联函数	11
1.7.1 内联函数的定义	11
1.7.2 应用示例	11
1.8 C++的引用	12
1.8.1 引用的含义与用法	12
1.8.2 应用示例	13
1.9 C++面向对象编程基础	16
1.9.1 封装	16
1.9.2 继承	17
1.9.3 多态性	17
小结	18
习题 1	18
第 2 章 C++编程的核心技术	21
2.1 类的定义与使用	21
2.1.1 类的定义	21

2.1.2	类的使用	22
2.1.3	应用示例	22
2.2	数据的封装	37
2.2.1	数据封装的内涵	37
2.2.2	应用示例	38
2.3	函数的重载	40
2.3.1	函数重载的内涵	40
2.3.2	应用示例	40
2.4	对象的初始化	42
2.4.1	初始化的方式	42
2.4.2	应用示例	43
2.5	默认构造函数、拷贝构造函数与析构函数	56
2.5.1	默认构造函数的形式与作用	56
2.5.2	拷贝构造函数的形式与作用	56
2.5.3	析构函数的形式与作用	56
2.5.4	应用示例	56
小结		62
习题 2		63
第 3 章	类成员与对象的构建	66
3.1	使用 <code>this</code> 指针	66
3.1.1	<code>this</code> 指针的用法	66
3.1.2	应用示例	66
3.2	使用静态成员	70
3.2.1	静态成员的作用与含义	70
3.2.2	应用示例	71
3.3	使用友元	74
3.3.1	友元的作用与形式	74
3.3.2	应用示例	75
3.4	使用对象成员与使用对象作为函数的参数	81
3.4.1	使用对象成员	81
3.4.2	使用对象作为函数的参数	83
3.5	使用对象数组	84
3.5.1	对象数组的含义	84
3.5.2	应用示例	84
3.6	使用指向对象的指针	89
3.6.1	对象指针的用法	89
3.6.2	应用示例	89
3.7	类型的转换	99

3.7.1 类型转换的规则	99
3.7.2 应用示例	99
小结	100
习题 3	100
第 4 章 派生类的构建	104
4.1 派生类的定义	104
4.1.1 定义形式	104
4.1.2 应用示例	104
4.2 类的保护成员	106
4.2.1 类的保护成员的定义	106
4.2.2 应用示例	106
4.3 访问权限的设置	107
4.3.1 访问权限的含义与使用	107
4.3.2 应用示例	109
4.4 派生类的构造函数与析构函数	111
4.4.1 用法	111
4.4.2 应用示例	112
4.5 多重继承	113
4.5.1 多重继承的定义与用法	113
4.5.2 多重继承的构造函数	113
4.5.3 应用示例	114
4.6 在派生类中改写超类的函数成员	118
4.6.1 函数成员的改写	118
4.6.2 应用示例	118
4.7 虚函数	121
4.7.1 用法	121
4.7.2 应用示例	121
4.8 纯虚函数与抽象类	124
4.8.1 纯虚函数与抽象类的定义	124
4.8.2 应用示例	124
小结	134
习题 4	135
第 5 章 运算符重载	139
5.1 运算符重载的作用与形式	139
5.1.1 作用与形式	139
5.1.2 应用示例	139
5.2 类运算符与友元运算符重载	140
5.2.1 作用与形式	140

5.2.2 应用示例	140
5.3 ++与--运算符的重载	142
5.3.1 ++与--运算符重载的形式	142
5.3.2 能被重载的运算符	143
5.3.3 应用示例	143
5.4 new 和 delete 运算符的重载	148
5.4.1 new 和 delete 重载的含义	148
5.4.2 内存分配错误	149
5.4.3 应用示例	149
小结	150
习题 5	151
第 6 章 输入/输出流的使用	152
6.1 标准的屏幕输出	152
6.1.1 使用 cout 流和插入操作符<<输出	152
6.1.2 调用流函数成员	153
6.1.3 使用格式化输出	154
6.2 标准的键盘输入	160
6.2.1 使用 cin 流和提取操作符>>输入数据	160
6.2.2 使用函数成员输入文本与字符	161
6.3 用户自定义类的输入/输出	163
6.3.1 用法	163
6.3.2 应用示例	164
6.4 格式化字符串流类的使用	165
6.4.1 用法	165
6.4.2 应用示例	166
6.5 磁盘文件的输入/输出	168
6.5.1 向文件写文本输出	168
6.5.2 从文件中读文本输入	169
6.5.3 get()和 put()函数成员的使用	170
6.5.4 使用 get()和 put()函数成员拷贝整个文件	172
6.5.5 使用二进制数据文件	173
6.5.6 随机访问数据文件	175
6.6 打印机的使用	177
6.6.1 用法	177
6.6.2 应用示例	177
小结	181
习题 6	181
第 7 章 模板	182
7.1 模板的基本概念	182

7.2 函数模板的定义与使用	183
7.2.1 函数模板的定义	183
7.2.2 函数模板处理方法	184
7.2.3 应用示例	184
7.3 类模板的定义与使用	192
7.3.1 类模板的定义	192
7.3.2 类模板的对象	193
7.3.3 应用示例	194
小结	197
习题 7	197
第 8 章 异常处理	198
8.1 使用 try/catch 和 throw 捕捉和抛出异常	198
8.1.1 形式	198
8.1.2 应用示例	198
8.2 捕捉多重异常	199
8.2.1 形式	199
8.2.2 应用示例	200
小结	202
习题 8	202
附录 A C++上机步骤	204
附录 B 习题参考答案	205
习题 1	205
习题 2	232
习题 3	243
习题 4	251
习题 5	277
习题 6	280
习题 7	284
习题 8	286
附录 C 本书所使用的符号	288
附录 D C++与 SQL 数据库连接技术	289
参考文献	291

第1章 C++的新特征

Visual C++是美国微软公司开发的一种面向对象编程语言，适用于编制各式各样的软件。尤其适用于开发中、大型程序项目。使用 Visual C++的集成化开发环境，能大大缩短开发时间，减少开发费用，使软件具有可靠性、实用性、可扩充性与可维护性。

本章介绍 C++的新特征，内容涉及：C++的输入/输出；C++的注解方式；C++的符号常量定义；C++的函数原型；C++函数的默认参数；C++的 new 与 delete 运算符；C++的内联函数；C++的引用；C++面向对象编程基础。

1.1 C++的输入/输出

1.1.1 引例

首先，我们从一个简单的 C 语言与 C++语言编程例子入手，来看看 C++与 C 语言的差别及 C++的编程风格。

【例 1.1】 输入你的名字，并在屏幕上显示所输入的名字。

(1) 用 C 语言编程

```
/* Name.c */
#define N 8
#include <stdio.h>
#include <stdlib.h>
void main()
{
    char *name;
    name=(char *)malloc(N*sizeof(char));
    printf("Please input your name: ");
    scanf("%s",name);
    printf("Your name is: %s\n",name);
}
```

运行结果：

```
Please input your name: Annie
Your name is: Annie
```

(2) 用 C++语言编程

```
//Name.cpp
const int N=8;
#include <iostream.h>
void main()
```

```

{
    char *name;
    name=new char[N];
    cout<<"Please input your name:";
    cin>>name;
    cout<<"Your name is "<<name<<endl;
}

```

运行结果:

```

Please input your name: Annie
Your name is: Annie

```

1.1.2 说明

比较上述两个程序，可以看出，C++在输入、输出，以及对指针动态分配空间方面做了较大的改进。

(1) C++的输入/输出是通过流实现的（将在第 6 章详细介绍），读者目前只需知道，其输入/输出是 C++本身自带的，并根据数据类型能自动使用合适的输入/输出方式。

(2) 在第一行的头文件中，C 语言使用的是标准输入输出头文件 `stdio.h`，而 C++使用的是输入输出流头文件 `iostream.h`，它体现了 C++的输入/输出风格。

(3) 在常量定义方面，C 语言用“`#define N 8`”，而 C++则用“`const int N=8;`”，后者是语句，前者不是语句，而是宏命令。

(4) 在定义字符指针 `name` 后，C 语言要用内存动态分配函数 `malloc()`给指针分配 N 个 `char` 字节的内存地址，即“`name =(char*)malloc(N*sizeof(char));`”，而 C++则用 `new` 操作符进行动态内存分配，即“`name = new char[N];`”，显然，C++比 C 语言简单得多。

(5) 在输出方面，C 语言使用 `printf()`函数，且要在参数表中指定输出格式，而 C++则只要使用“`cout<<`”就能把后面的内容送到标准输出设备（显示器），并能自动地进行格式转换。

(6) 在输入方面，C 语言中使用 `scanf()`函数进行输入，并要指定输入格式，而在 C++中，则只要使用“`cin>>`”就能从输入设备（键盘）接收的数据存入后面的变量。显然可见，C++的输入/输出比 C 语言简单，方便得多，但一定要加上包含头文件 `iostream.h`，而不是 `stdio.h`。然而，在 C++中也允许使用 C 风格的输入/输出，但使用 `scanf()`和 `printf()`语句时，一定要使用包含头文件 `stdio.h`。

1.2 C++的注释方式

1.2.1 单行注释

C++提供单行注释方式。以“`//`”开头直到行结束均作为注解，例如：

```
int i=7; // 整型变量说明
```

1.2.2 多行注释

在 C++中，可使用 C 语言的多行注释方式，即“`/*`”注解内容“`*/`”。例如：

```
/*This is C++ language*/
/*Visual C++ is Microsoft's Integrated Development
Environment (IDE) for developing*/
```

1.3 C++的符号常量定义

1.3.1 定义符号常量的形式

在C语言中，使用#define 宏定义来定义符号常量，例如：

```
#define PI 3.14159
```

而在C++中，则使用 const 类型修饰符来定义符号常量，其形式为

```
const 类型 符号常数名=常量;
```

例如: const double PI=3.14159;

1.3.2 注意事项

(1) 对基本数据类型加上 const 修饰符，编译器就把它作为一个常量，不再对其分配内存，而且，每当在程序中遇到它时，就被说明时所给定的初值所取代。

(2) 使用 const 定义符号常量，可允许编译器对其进行类型检查，同时避免对常量分配内存，还可提高程序的可读性。

(3) const 可以修饰指针变量，以 char 指针为例，有下面三种情况：

- ① const char *cp; /*说明 cp 是一个字符指针，它只能指向一个被 const 修饰的 char 类型的变量，即只能指向一个字符常量，例如：
const char *cp;
const char var='c';
cp=&var; */
- ② char *const cp; /*说明指针 cp 本身是常量，即 cp 指向 char 类型的一个固定地址，该地址中的内容是可修改的，例如：
char const *cp="C++";
cp[0]=C; */
- ③ const char *const cp; /*说明指针 cp 及其所指的内容均为常量，对任何一个进行修改均是错误的。 */

1.3.3 C与C++的比较

C与C++的比较见表1.1。

表 1.1 C 与 C++的基本比较

语言 类别	C		C++
输入/输出	文件	头文件	流头文件
	包含	stdio.h	iostream.h
	输入	scanf 函数	cin>>
	输出	printf 函数	cout<<
动态分配	malloc(类型) / calloc(n,类型)		new 类型/new 类型(size)
换行	\n		endl
字符常量	#define 字符常量名 常量		const 类型 字符常量名=常量;
注释	形式	/* 注释内容 */	// 注释内容
	性质	多行	单行
程序扩展名	.C		.CPP
组成之一	函数		方法/函数

1.3.4 应用示例

【例 1.2】 输入一个圆的半径，计算圆的面积。

(1) 用 C 语言编程

```
/*CircleArea.c*/
#include <stdio.h>
#define PI 3.14159f
void main()
{
    float Radius, Area;
    printf("Enter radius of a circle: ");
    scanf("%f",&Radius);
    Area=PI*Radius*Radius;
    printf("Area=%f\n",Area);
}
```

运行结果:

```
Enter radius of a circle: 2
Area=12.5664
```

(2) 用 C++语言编程

```
// CircleArea.cpp
#include <iostream.h>
const float PI=3.14159f;
void main()
{
    float Radius, Area;
    cout<<"Enter radius of a circle: ";
```

```

    cin>>Radius;
    Area=PI*Radius*Radius;
    cout<<"Area="<<Area<<endl;
}

```

运行结果:

```

Enter radius of a circle: 2
Area=12.5664

```

【例 1.3】 输入你的名字，并在屏幕上显示所输入的名字。

(1) 用 C 语言编程：使用字符串

```

/* LJY2B.c */
#include <stdio.h>
typedef char string[80]; //定义字符型数组
void main()
{
    string SourceName,LastName;
    printf("Please your source and last name: ");
    scanf("%s %s",SourceName,LastName);
    printf("Your name is %s %s\n",SourceName,LastName);
}

```

运行结果:

```

Please your source and last name: Annie Liu
Your name is Annie Liu

```

(2) 用 C++语言编程：使用字符串

```

//LJY1B.cpp
#include <iostream.h>
typedef char string[80]; //定义字符型数组
void main()
{
    string SourceName,LastName;
    cout<<"Please your source and last name: ";
    cin>>SourceName>>LastName;
    cout<<"Your name is "<<SourceName<<" "<<LastName<<endl;
}

```

运行结果:

```

Please your source and last name: Annie Liu
Your name is Annie Liu

```

【例 1.4】 输入两个数，求其和。

(1) 用 C 语言编程

① 主函数型

```

/* Sum.c */

```



```
#include <stdio.h>
void main()
{
    float a,b,sum;
    printf("Enter two numbers: ");
    scanf("%f%f",&a,&b);
    sum=a+b;
    printf("sum=%f\n",sum);
}
```

运行结果:

```
Enter two numbers: 10.25 11.19
sum=21.440000
```

② 用户自定义函数型

方案一:

```
/* SumFunA.c */
#include <stdio.h>
void SumFun(float a,float b) //定义有参函数
{
    printf("Sum=%f\n",a+b);
}
void main()
{
    float a,b;
    printf("Enter two numbers: ");
    scanf("%f%f",&a,&b);
    SumFun(a,b); //调用有参的函数
}
```

运行结果:

```
Enter two numbers: 10.25 11.19
sum=21.440000
```

方案二:

```
/*SumFun.c*/
#include <stdio.h>
void SumFun() //定义无参无返回值的函数
{
    float a,b;
    printf("Enter two numbers: ");
    scanf("%f%f",&a,&b);
    printf("Sum=%f\n",a+b);
}
```