

任 光 编

学习 C++ 的最佳途径



大连海事大学出版社

学习 C⁺⁺的最佳途径

任 光 编

大连海事大学出版社

1995 年

(辽)新登字 11 号

图书在版编目(CIP)数据

学习 C++ 的最佳途径/任光编. —大连:大连海事大学出版社, 1995. 2

ISBN 7-5632-0853-4

I. 学… I. 任… III. C 语言-学习参考资料 N. TP312

中国版本图书馆 CIP 数据核字(94)第 00520 号

大连海事大学出版社出版

(大连)

大连海事大学出版社印刷厂印刷 大连海事大学出版社发行

1995 年 2 月第 1 版 1995 年 2 月第 1 次印刷

开本: 787×1092 1/32 印张: 8.5

字数: 184 千 印数: 0001~2000

定价: 9.80 元

内 容 简 介

本书是为已经了解了C语言的读者学习C⁺⁺而编写的。C⁺⁺是C的更高一级的程序语言,来源于C语言,是对C的扩展。它改进了C的语法结构和可应用性,引进了面向对象的程序语言结构。

本书共分十章。第一章基本输入/输出流;第二章对C的扩展;第三章C⁺⁺的自由存储区;第四章重复装载函数;第五章C⁺⁺的结构;第六章引用;第七章类;第八章重复装载运算符;第九章类的继承;第十章C⁺⁺的高级输入/输出流。

本书适用于各专业、各层次的学生、教师、计算机软件工程技术人员,也可供研究生,本科生做为教材使用。

本书初稿曾作为大连海事大学九四级研究生C⁺⁺课程的教材。

前 言

C++代表着计算机语言的发展方向,即面向对象的程序语言,是最有发展前途的计算机语言之一。随着计算机的普及应用,乃至进入家庭,C++将越来越受到人们的重视。虽然国内已有几本关于C++语言的书,但适合于自学的书却找不到,本书编写的目的也正在于填补此项空白。

本书编者在挪威 Trondheim 大学攻读博士学位以及做博士后研究共六年时间。在此期间,发现学习计算机语言的最好方法就是在计算机上通过典型例子小程序。回忆作者以前在国内学习计算机语言的时候,深感学习方法的落后。老师认真备课与讲课,从定义到语法讲的头头是道;学生认真记笔记、背定义。学完后对这一程序语言还是没有总体的概念,不会应用。由于学习方法不当,浪费了大量的时光。为改进国内学习计算机语言的方法,作者决心将此书编写出来,奉献给国内的计算机C和C++语言爱好者。本书采用的学习方法是把C++语言的每一个功能都嵌入到小程序之中,当读者在计算机上编译并执行该程序,就马上了解这一功能的执行结果,然后看一看下面的解释就立刻明白了。这种学习方法看得见、摸得着、用得上。看得见,读者可以看到这一功能的语句是在程序什么地方;摸得着,读者可以摸到(看到)执行的结果;用得上,当你编应用程序时你有了一个小程序参考、模仿,这就节

约了大量的查找例子的时间。按着本书的学习方法,读者决不会浪费时间走弯路,而且一定能学到实用的知识。

书中大约 130 个程序例子全部在微机 386 和 486 使用 Borland C++ 2.0 和 3.0 版编译器调试通过、执行并给出执行结果。读者可以直接将程序送入计算机中执行,不会有任何麻烦。送入程序时应注意在每一章的后一个程序与前一个程序的差别只有三、五个语句。注意到这点,可节省大量键盘输入时间,并且这三、五个语句正是该程序中要学习的重点。如果读者不想自己输入,作者已将全部程序存在计算机中,可为你免费复制,只收软盘的费用。

本书的练习设计的顺序与读者学习过程相适应。每一个小程序都是由先一个程序的修改或增加一些语句而得到的。读者应该遵照本书练习出现的次序而学习。由于书中给出了程序的执行结果,即使没有上机条件的读者通过阅读程序并对照执行结果也可以学到 C++ 的基本内容。

本书不是 C++ 语言的参考手册,它是以教学为目的,帮助读者学习 C++ 语言程序。作者认为,学习计算机语言需要两本书:一本是简单易学的书,不求全面;另一本是参考手册,内容要详尽。本书属于前者。它首先引进 C++ 简单的概念,并渐渐引导你进入更复杂的题目。

本书中的练习是很小的程序。每一个练习的目的是演示某一 C++ 语言的特点和功能;某些练习可以变成有用的软件开发工具。读者应按本书的学习方法做全部练习,这样你会学到 C++ 的基本内容。真正学到 C++ 的时候是当你使用 C++ 时,这一点对所有的其它计算机语言都是同样的。

C++ 语言像 C 语言一样,不是很容易学的,但是 C++ 也像

C 一样将变成软件工作者所选择的语言, C++ 将有一天取代 C。我相信你学了这本书的新技术后将变成 C++ 的爱好者。

记住, C++ 编译器将把 C++ 程序在计算机内部翻译成 C 程序, 整个 C 语言的结构特点和 C 的标准函数库都将自动成为 C++ 的一部份。读者可从 C 开始, 一次学一点 C++ 的特点。本书采取这一方法。

本书确实提供了学习 C++ 的最佳方法。阅读本书不但学到了 C++ , 而且还掌握了学习与教授计算机语言的新方法。一书在手学会 C++ , 本书确实做到了无师自通 C++ 。编者希望通过本书献给读者一份爱。

本书在编写过程中曾得到大连海事大学校长司玉琢教授, 科研处长殷佩海教授, 轮机工程学院院长吴恒教授以及船舶自动控制研究室主任贾欣乐教授等同志的支持和帮助。对此, 作者深表谢意。我的夫人佟雅君誊写了书稿, 并做了文字修饰工作。

任 光

1994 年国庆

于大连海事大学

目 录

前 言

第一章 基本输入/输出流	(1)
1.1 标准的输入/输出流	(2)
1.1.1 格式化输出	(4)
1.1.2 标准输入流	(5)
1.2 C++ 的注解	(7)
1.3 原型	(7)
1.4 C++ 的关键字	(9)
1.5 小结	(10)
第二章 对 C 的扩展	(11)
2.1 函数的缺补参数变量	(11)
2.2 变量说明的位置	(13)
2.3 全局作用域分解操作符	(15)
2.4 在线函数 (inline)	(16)
2.5 常值变量 (const)	(17)
2.6 枚举数据类型 (enum)	(18)
2.7 与 C 语言连接说明	(21)
2.8 匿名联合	(23)
2.9 小结	(25)

第三章 C++的自由存储区	(26)
3.1 分配和释放内存操作符 new 和 delete	(26)
3.1.1 分配固定维的数组	(27)
3.1.2 分配动态数组	(28)
3.2 自由存储区用完	(32)
3.2.1 分配内存出错处理指针 _new_handler ...	(32)
3.2.2 设置出错处理指针 set_new_handler	(32)
3.2.3 一个释放内存的机会	(34)
3.3 建立用户分配和释放内存操作符 new 和 delete	(34)
3.4 小结	(39)
第四章 重复装载函数	(40)
4.1 不同运算的重复装载函数	(40)
4.2 不同格式的重复装载函数	(42)
4.3 小结	(43)
第五章 C++的结构	(44)
5.1 作为数据类型的结构	(44)
5.2 带有函数的结构	(46)
5.2.1 面向对象程序语言一瞥	(47)
5.2.2 加入函数到结构中	(47)
5.3 同一结构的多个实例	(48)
5.4 重复装载的结构函数	(50)
5.5 小结	(52)
第六章 引用	(53)
6.1 引用是替代名	(54)
6.2 必须将引用初始化	(56)

6.3	隐藏的匿名目标.....	(57)
6.4	引用最适合与用户定义的数据类型一起工作.....	(59)
6.5	作为函数参数的引用.....	(60)
6.5.1	引用改进了参数传递中的额外开销.....	(60)
6.5.2	引用删除了指针标记.....	(62)
6.5.3	引用调用.....	(63)
6.5.4	何时用指针而不用引用.....	(65)
6.6	返回引用.....	(66)
6.7	不能用引用做的事.....	(68)
6.8	小结.....	(69)
第七章	类.....	(70)
7.1	设计一个类.....	(70)
7.1.1	定义一个类.....	(70)
7.1.2	类的说明.....	(72)
7.1.3	类的成员.....	(73)
7.1.4	类目标的作用域.....	(76)
7.1.5	内联函数 inline.....	(76)
7.1.6	带有缺补变量的构造函数.....	(78)
7.1.7	重复装载构造函数.....	(79)
7.2	类的转换.....	(80)
7.2.1	转换函数.....	(81)
7.2.2	转换类.....	(85)
7.2.3	调用转换函数.....	(87)
7.2.4	哪里发生转换.....	(89)
7.3	处理私有数据成员.....	(94)

7.4	友元.....	(97)
7.4.1	友元类.....	(98)
7.4.2	友元函数	(101)
7.5	类和引用	(105)
7.6	析构函数	(105)
7.7	指针 this	(113)
7.7.1	返回指针 * this	(113)
7.7.2	用指针 this 做表的连接	(118)
7.7.3	如何不用指针 this	(120)
7.8	类目标的数组	(121)
7.8.1	类数组的构造函数	(122)
7.8.2	类数组的析构函数	(123)
7.9	静态成员	(125)
7.9.1	静态数据成员	(125)
7.9.2	静态函数成员	(128)
7.9.3	静态公有成员	(131)
7.10	类和自由存储区.....	(137)
7.10.1	使用操作符 new 和 delete 管理类 目标的作用域.....	(137)
7.10.2	构造函数与 new、析构函数与 delete ...	(137)
7.10.3	自由存储区和类数组.....	(139)
7.10.4	重复装载类的 new 和 delete	(141)
7.11	类的源程序和目标文件管理.....	(146)
7.11.1	类的定义在头文件中.....	(146)
7.11.2	类的函数成员在函数库中.....	(147)
7.12	小结.....	(147)

第八章 重复装载运算符	(148)
8.1 二元算术运算符	(150)
8.1.1 类的成员运算符函数	(150)
8.1.2 非成员运算符函数	(152)
8.2 关系运算符	(155)
8.3 赋值运算符	(157)
8.4 自动增量和自动减量	(158)
8.5 地址和引用运算符	(160)
8.6 一元加和减运算符	(162)
8.7 下标运算符	(163)
8.8 函数调用运算符	(167)
8.9 指向成员的运算符	(168)
8.10 小结	(172)
第九章 类的继承	(173)
9.1 基本类和派生类	(173)
9.2 简单继承	(175)
9.2.1 设计派生类	(176)
9.2.2 保护成员	(177)
9.2.3 公有类和私有类	(178)
9.2.4 在基本类和派生类中的构造函数	(179)
9.2.5 派生类中的取代成员	(180)
9.3 从派生类导出的派生类	(182)
9.4 指向基本类和派生类的指针	(185)
9.5 伴随基本类和派生类的全局作用域 分解运算符	(186)
9.6 基本类和派生类的引用	(188)

9.7	虚函数	(189)
9.7.1	带有虚函数的类 Time	(189)
9.7.2	代替虚函数的取代	(193)
9.7.3	没有派生类取代的虚函数	(194)
9.7.4	纯虚函数	(196)
9.7.5	虚函数和多路派生类	(198)
9.7.6	虚函数的虚函数	(200)
9.8	基本类和派生类中的析构函数	(204)
9.9	多路继承	(208)
9.9.1	多路继承的二义性	(212)
9.9.2	多路继承构造函数的执行	(212)
9.9.3	多路继承类析构函数的执行	(213)
9.10	虚拟的基本类	(213)
9.11	小结	(217)
第十章	C++的高级输入/输出流	(219)
10.1	流	(220)
10.1.1	类 ios	(220)
10.1.2	类 ostream	(220)
10.1.3	类 istream	(221)
10.1.4	类 istream	(221)
10.2	缓冲输出	(221)
10.3	格式化的输出	(223)
10.3.1	输出宽度	(224)
10.3.2	输出的调整	(227)
10.3.3	精度	(228)
10.3.4	操作符、标记符和成员函数	(231)

10.4	输入/输出成员函数	(231)
10.4.1	输出成员函数 put 和 write	(231)
10.4.2	输入成员函数 get 和 getline	(232)
10.5	重复装载插入<<和提取>>操作符.....	(237)
10.5.1	重复装载 <<	(237)
10.5.2	重复装载 >>.....	(239)
10.6	文件的输入/输出	(240)
10.6.1	输出类 ofstream	(241)
10.6.2	输入类 ifstream	(243)
10.6.3	查寻.....	(245)
10.6.4	类 fstream	(247)
10.6.5	打开和关闭文件流.....	(248)
10.6.6	如何打开文件.....	(250)
10.6.7	测试错误.....	(252)
10.7	小结.....	(253)
结束语.....		(254)

第一章 基本输入/输出流

C++是C的扩展和改进。扩展是引进了新的语言结构特点;改进是以更好的方法做C语言也能做的事情。学了C++以后,我们可以应用新的知识建立自己的面向对象的C++语言系统。本章是通向这一目标的第一步。练习1.1是最小的C++程序,它是我们走向C++的开始。

练习 1.1 最小的 C++ 程序

```
main()  
{  
}
```

我们应该很熟悉练习1.1,它同C语言一样需一个主程序名字为main。这个程序没有做任何事情。这里应注意C++源程序的文件名要求有CPP的后缀,即“文件名.CPP”或者“*.CPP”,这是Borland C++编译器的要求。还有的编译要求有CC的后缀,即“*.CC”,这是一个简单的问题,但必须注意到。练习1.2的程序确实做了一点事情。

练习 1.2 打印信息

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

```
main()  
{  
    cout << "Hello, world";  
}
```

执行结果:Hello,World

这个程序包含了题头文件 `iostream.h`, 此文件下面要讲到, 特别是在第十章要详细讲解有关这个文件的内容。接着, 我们可以看到一个陌生的语句或语法, 有一个变量名 `cout` 和一个操作符 `<<`。现在, 我们还不需详细讲解, 只知道它在这里的功能是打印信息就可以了。

1.1 标准的输入/输出流

练习 1.2 中的程序为我们引进了 C++ 功能非常强的工具, 名字为“类”, 一个新的特点。它允许程序员定义新的数据类型和操作符或运算符。在这里, 我们还不能详细的讲解 C++ 的类, 但我们用到输入/输出流, 这就是一个类的特点。因为我们用到 C++ 的输入、输出流来作为人与计算机交换信息的手段, 所以, 我们首先介绍这一功能。

在练习 1.2 中, 字符串“Hello, World”被显示在屏幕上; 操作符 `<<` 是输出符号, 它指出要送什么东西到哪里去。在此, 字符串被送到 `cout` 设备。

假设我们想要在屏幕上显示一个整型变量。下面练习 1.3 给出这样的例子, C++ 不需要描述数据类型输出格式, 它本身能判别输出的数据类型。这一点与 C 不一样。

练习 1.3 标准输出流

```
#include <iostream.h>
main()
{
    int amount = 123;
    cout << amount;
}
```

执行结果:123

在练习 1.3 中,我们在屏幕上显示了一个整型变量。假定我们想显示这一变量是一句话的一部分,我们可以将在练习 1.2 和 1.3 学过的知识组合在一起,cout 流可以识别出想要显示的数据类型,因为 C++ 编译器有辨识数据类型的功能。在后边第四、八章中还要讲到 C++ 编译器将如何完成这样的功能,现在我们暂时接受 C++ 语言的这个特点。练习 1.4 演示如何送一字符串、一个整型变量和一个字母到输出流。

练习 1.4 送多种数据类型到输出流

```
#include <iostream.h>
main()
{
    int amount = 123;
    cout << "The value of amount is ";
    cout << amount;
    cout << ' ';
}
```