

轻轻松松学电脑



C++ 语言编程 实用教程

北京希望电子出版社 总策划
宋振会 编 著



科学出版社
www.sciencep.com

轻轻松松学电脑

C++ 语言编程 实用教程

北京希望电子出版社 总策划
宋振会 编 著



科学出版社

www.sciencep.com

内 容 简 介

本教程从 C 语言基础开始, 到 C++ 的面向对象的编程思想。

本书共由 16 课和 3 个附录构成。主要内容包括: 编程逻辑和技术; 编辑和编译 C++ 程序; 常量、变量和指针; 运算符、优先级和结合律; 函数、函数指针和指针函数; 面向对象的编程方法: 类; 成员函数、构造符和析构符; 条件判定、循环和跳转; 数组、数组指针和指针数组; 静态多态: 构造符重载和运算符重载; 类的关系: 类的包含和类的继承; 动态多态: 滞后联编和函数重载; 多重继承、歧义性和虚基类; 文件的输出和输入; 类指针、单链表和双链表; 用链表实现栈和队列。

本书参考了大量的国际软件工程师培训教程, 以及作者多年编程经验和教学经验, 采用符合国际性标准的编程方法和惯例。实例典型、通俗易懂、版式活泼。

本书适合想通过自学快速掌握 C++ 的读者, 也可作为培训教材。

本书配有内容为范例程序源代码以及 Quincy 99 编译器和 Editplus 2 编辑器的光盘一张。需要本书或技术支持的读者, 请与北京中关村 083 信箱 (邮编 100080) 发行部联系, 电话: 010-62528991, 62524940, 62521921, 62521724, 82610344, 82675588 (总机)

传真: 010-62520573, E-mail: yanmc@bhp.com.cn

图书在版编目 (CIP) 数据

C++ 语言编程实用教程/宋振会编著. —北京: 科学

出版社, 2003

ISBN 7-03-012268-2

I. C... II. 宋... III. C 语言-程序设计-教材

IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 086053 号

责任编辑: 周艳

/ 责任校对: 赵文博

责任印刷: 双青

/ 封面设计: 梁运丽

科学出版社 出版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

双青印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2003 年 10 月第一版 开本: 787×1092 1/16

2003 年 10 月第一次印刷 印张: 26 1/2

印数: 1—4000 字数: 607620

定价: 39.00 元 (含 1CD)

傲雄软件工作室

主编	宋振会						
编委	李秋芬	张 瑾	孔祥国	张卫东	白先栋	张 青	
	王志刚	顾茂香	高冬梅	梁冰捷	肖国建	张 庆	
	赵 洁	仇晓敏	刘国华	王 斌	魏 伟	顾 蓓	
	陈荣国	李 丽					

前言

关于本教程

本教程的目的

目前，由于软件产业的美好前景和诱人的薪酬，软件培训和软件学习如火如荼，热遍大江南北，光 NIIT、北大青鸟等知名的培训机构在国内就有 1000 多家，培训学员 200 多万人。本人作为 NIIT 特聘教师，深切体会到由于时间或学费等问题不能参加培训而又渴望学习和快速掌握软件编程的人们。需要这样一本详细浅显、通俗易懂的教程。

本教程的主要内容

阅读本教程不需任何编程基础，本教程从最基础的 C 语言讲起，包括变量、数组、函数和指针；运算符、优先级、条件构造和循环迭代，并引入了 C++ 的类、构造符和析构符第 1 课到第 9 课。从第 10 课到第 16 课主要是 C++ 的内容，包括函数重载、构造符重载和运算符重载；类的包含、类的继承、多重继承和虚基类；以及类指针、链表、栈和队列等内容。

谁能参加本教程

本教程适用于想使用 C++ 语言，进入面向对象编程世界的任何人。我在学校读书的时候，经常听老师说的一句话是“好记性，不如懒笔头”。说明动手的重要性。学习计算机语言编程，尤其如此。尽管本教程附带的 CD-ROM 光盘中包含全书所有范例程序的源代码，但要求学员在读懂代码的基础上，合上教程，亲手默写并调试运行，只有这样才能亲自感受编程的感觉。

本教程附带的 CD-ROM 光盘中所有范例程序都经过 Microsoft Visual C++ 编译器编译运行。

怎样使用本教程

本教程范例程序所涉及的知识点，必在前面的相关课中详细讲过，因此要求学员按顺序学习，注意阅读每课后的小结，完成独立实践，是对本课学习内容的总结和复习。

参加本书编写的人员

主编 宋振会

编委 李秋芬 张 瑾 孔祥国 张卫东 白先栋 张 青
王志刚 顾茂香 高冬梅 梁冰捷 肖国建 张 庆
赵 洁 仇晓敏 刘国华 王 斌 魏 伟 顾 蓓
陈荣国 李 丽

目 录

引言

关于本教程	I
谁能参加本教程	I
怎样使用本教程	II

第 1 课 编程逻辑和技术..... 1

引入框图	3
I-P-O 周期	3
使用框图表示程序流程	3
变量、常量和内存	7
循环迭代和条件判定	9
预检和预检表	9
循环迭代	9
条件判定	12
理解编程的模块化方法	15
养成一个好的习惯	20
小结	24
独立实践	25

第 2 课 编辑和编译 C++ 程序..... 26

C 和 C++ 语言概述	28
计算机语言的代	28
C 和 C++ 语言的发展历史	29
C 和 C++ 语言的特点	30
编写“我的第 1 个 C++ 程序”	32

编写 C++ 程序	32
保存 C++ 程序	33
C++ 程序的成分	33
使用名字空间 <code>std</code>	42
程序的编译、链接和执行	46
Linux 平台下的 GNU 编译器	47
Quincy 99 集成开发环境编译器	47
Microsoft Visual C++ 编译器	48
使用模块化编程方法	55
使用 C 结构化编程方法	57
使用 C++ 面向对象编程方法	59
小结	61
独立实践	62

第 3 课 常量、变量和指针

数据类型概述	64
数据类型的分类	64
常量、变量和内存	65
基本数据类型	70
布尔型 (<code>bool</code>)	70
字符型 (<code>char</code>)	71
ASCII 字符集	73
整型 (<code>int</code>)	75
浮点型 (<code>float</code>)	76
数据大小与 <code>sizeof</code> 运算符	77
数据类型转换	78
指针变量	81
指针基础	81
定义指针变量	83
引用指针变量 <code>&</code> 和 <code>*</code>	84
类型修饰符	87
<code>Const</code> 类型修饰符	87
<code>Volatile</code> 类型修饰符	87

小结	88
独立实践	89
第 4 课 运算符、优先级和结合律	90
基本概念	92
基本运算符	93
算术运算符	93
算术赋值运算符	95
一元增量、减量运算符	96
比较运算符	97
逻辑运算符	98
条件运算符	100
逗号运算符	101
优先级与结合律	102
小结	106
独立实践	107
第 5 课 函数、函数指针和指针函数	108
定义函数和调用函数	110
变量的作用域	113
块作用域	115
函数作用域	115
全局作用域	115
带参数的函数	117
形参和实参	117
调用函数	120
声明函数原型	121
函数调用方式	123
直接调用（也叫传值调用）	123
使用别名的引用调用	125
用指针的引用调用	130
new 和 delete 运算符	133
变量的存储类型	137

静态存储和动态存储	137
函数指针	140
指针函数	142
小结	143
独立实践	144
第 6 课 面向对象的编程方法：类	145
C++中的类	146
声明类	146
作用域分解运算符 (::)	147
访问成员变量和成员函数	150
创建类对象	150
类对象访问符 (.)	151
类指针访问符 (->)	154
类的访问区分符	159
抽象和封装	159
使用访问区分符实现抽象和封装	160
静态变量和静态函数	165
静态变量	165
静态函数	167
类作用域、友元函数和友元类	169
类作用域	169
友元函数	170
友元类	171
小结	173
独立实践	174
第 7 课 成员函数、构造符和析构符	176
构造符函数的需要	177
声明构造符函数	178
成员方式初始化	180
析构符函数	182
对象的作用域和生命周期	184

带参数的构造符.....	186
初始化值来自键盘.....	186
初始化值来自实参.....	187
带参数默认值的构造符.....	188
小结.....	190
独立实践.....	191

第 8 课 条件判定、循环和跳转..... 192

条件构造.....	193
if...else 构造.....	193
switch...case 构造.....	201
循环构造.....	205
while 循环.....	205
do...while 循环.....	207
break 和 continue 语句.....	208
for 循环构造.....	212
跳转构造.....	218
goto 语句的错误用法.....	218
goto 语句错误的改正.....	219
小结.....	221
独立实践.....	222

第 9 课 数组、数组指针和指针数组..... 224

一维 int 数组.....	225
int 数组和数组指针.....	229
串_一维 char 数组.....	230
字符串和数组指针.....	231
string 字符串对象.....	232
构造字符串.....	232
给字符串对象赋值.....	232
字符串的连接.....	233
字符串的下标.....	233
字符串的子串.....	233

字符串的比较	234
指针数组和指针的指针	235
指针数组	235
指针的指针	235
二维数组	237
三维数组	242
小结	244
独立实践	245

第 10 课 静态多态：构造符重载和运算符重载..... 246

静态多态性	247
函数重载	248
成员函数重载	248
构造符重载	251
运算符重载	254
运算符重载的需要	254
一元运算符重载	255
简单的前缀一元运算符	255
事前和事后的增量和减量运算符	259
重载二元运算符	262
简单运算符	262
加号运算符重载	263
大于 (>) 运算符重载	272
this 指针	275
小结	276
独立实践	277

第 11 课 类的关系：类的包含和类的继承..... 278

识别类之间的关系	279
继承关系	280
组合关系	281
利用关系	282
实例化关系	282

类和继承性	283
继承性	283
调用构造符和析构符的顺序	285
基类初始化	287
派生的访问区分符	289
小结	292
独立实践	293
第 12 课 动态多态：滞后联编和函数重载	295
实现滞后联编	296
联编的概念	296
用虚函数实现滞后联编	297
静态多态性与动态多态性的比较	302
小结	312
独立实践	313
第 13 课 多重继承、歧义性和虚基类	314
多重继承性	315
多重继承中访问区分符	317
多重继承中的歧义性	318
虚基类	322
调用构造符和析构符	324
小结	328
独立实践	329
第 14 课 文件的输出和输入	331
流类层次结构	332
流的插入和抽取	334
流的插入	334
流的抽取	335
文件输出和输入	339
使用内部数据类型的文件输出和输入	339
使用对象的文件输出和输入	342
二进制输出和输入	344

文件的打开和关闭.....	349
open()函数.....	349
close()函数.....	349
打开方式位	353
文件指针	354
查询文件	355
修改文件的内容	355
小结	356
独立实践	357

第 15 课 类指针、单链表和双链表..... 358

链表	359
链表的类型.....	359
链表的操作.....	361
链表的应用.....	361
单链表	362
INFO 在节点中的表示.....	362
Node 类的表示.....	365
List 类的表示.....	366
链表中插入节点.....	369
作为新链表的第 1 个节点.....	369
在表的开始处插入节点.....	370
在表的中间插入节点	371
在表的尾部插入节点	372
修改链表	374
遍历链表	375
查询信息	376
删除节点	377
链表排序	379
链表操作	380
小结	384
独立实践	385

第 16 课 用链表实现栈和队列	386
栈	387
栈操作	388
用链表实现栈	389
队列	394
队列类型	394
队列操作	396
用链表实现队列	397
小结	402
独立实践	403

附录

附录 A ASCII 字符集	404
附录 B 运算符优先级	408
附录 C C++ 的关键字	410

第 1 课

编程逻辑和技术

目标:

本课中,你将学习:

- ☒ 引入框图
 - ☒ I-P-O 周期
 - ☒ 使用框图表示程序流程
- ☒ 变量、常量和内存
- ☒ 循环迭代和条件判定
 - ☒ 预检和预检表
 - ☒ 循环迭代
 - ☒ 条件判定
- ☒ 理解编程的模块化方法
- ☒ 养成一个好的习惯

我们在开始任何一门编程语言之前，首先需要知道和了解编程所使用的技术。也许，你学过 **BASIC**、**FORTRAN** 和 **COBOL** 等编程语言，在编程之前，首先要使用框图绘制程序流程图，绘制程序流程图的过程，就是程序设计的过程，它包含着编程逻辑和编程技术，就好像，我在写这本书之前，首先要写好大纲和目录，确定要编写的内容和深度，以及所面向的读者群体，构起本书的框架。

本课提供你对编程技术的理解，使用这种技术来表示使用程序求解问题所需要的编程逻辑。主要使用框图技术开发基本的编程逻辑。

本课你将学习如下技术：

- 使用框图绘制程序流程图
- 用条件表示程序逻辑
- 理解循环和迭代
- 实现模块化编程
- 学会一题多解，法向思维和超常规思维