

精品丛书“24小时轻松掌握系列”
全新改版，重装上市

C++ 程序设计

匡松 主编

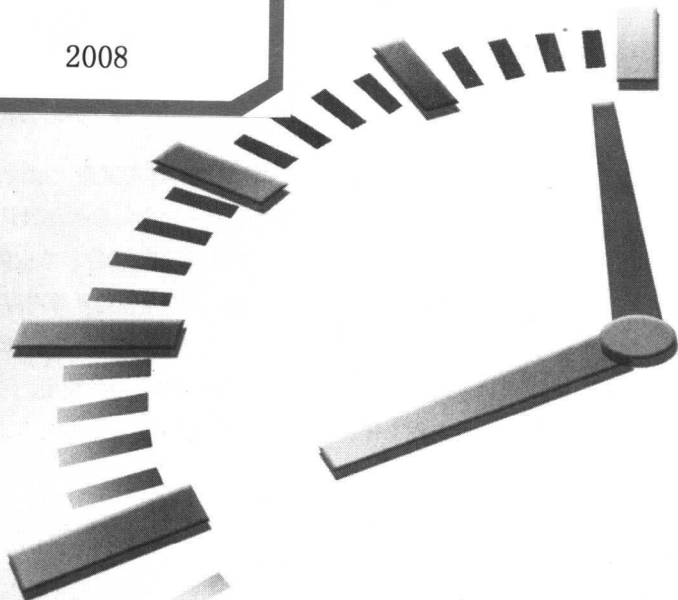
- 打破常规，24学时合理安排，焕然一新。
- 满足等考，点透程序设计理念，轻松掌握。
- 边学边练，理论习题相得益彰，效果最棒。



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

TP312/2843

2008



C++程序设计

24 学时轻松掌握

匡 松 主 编

缪春池 吕峻闽 张承虎 谢志龙 陈松明 副主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书根据实际应用和全国计算机等级考试新大纲规定的二级 C++ 考试内容的要求编写而成。全书共 24 学时, 系统介绍了 C++ 面向对象程序设计的基本概念和方法。具体包括 C++ 语法基础、三种程序控制结构、函数、数组、指针、结构体、共用体和枚举类型、类与对象、运算符重载、输入/输出流、文件流以及编译预处理等内容。

本书内容详实、通俗易懂、例题丰富、图文并茂, 每学时提供了多种形式的习题(附有答案)供读者练习。本书尤其便于编程爱好者自学使用。可作为高等学校学生学习 C++ 程序设计的教材, 也可作为计算机培训班的教材, 还可作为全国计算机等级考试二级考试人员或编程初学者的自学用书。

图书在版编目(CIP)数据

C++程序设计 24 学时轻松掌握/匡松主编. —北京: 中国铁道出版社, 2008. 4

(24 小时轻松掌握)

ISBN 978-7-113-08733-3

I. C… II. 匡… III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2008) 第 043786 号

书 名: C++程序设计 24 学时轻松掌握

作 者: 匡 松 等

策划编辑: 严晓舟 荆 波

责任编辑: 贾 星 刘彦会

封面设计: 付 巍

编辑部电话: (010) 63583215

封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 三河市华丰印刷厂

版 次: 2008 年 6 月第 1 版 2008 年 6 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 18.75 字数: 438 千

印 数: 1~5 000 册

书 号: ISBN 978-7-113-08733-3/TP·2765

定 价: 29.00 元

版权所有 侵权必究

凡购买铁道版的图书, 如有缺页、倒页、脱页者, 请与本社计算机图书批销部调换。

丛书序

只需 24 小时， 轻松具备一种电脑技能

进入 21 世纪的你，如果还不能熟练地使用电脑，不能不说是一种遗憾。

电脑的世界是十分美妙的世界，我们通过 Internet 了解世界，通过 E-mail 和朋友们沟通，上网购买所需要的图书……电脑，越来越成为生活的必需品，给我们的工作、学习和生活带来了巨大帮助。

只要会中文，就可以享受高科技带来的便利

可是，在今天，还是有不少读者朋友，不会使用电脑，或者说不能熟练地驾驭电脑，让电脑帮我们完成各种工作，体验电脑文化带给我们的神奇感觉，享受高科技的产品带给我们的便利。

很多读者向我们抱怨，电脑学习这么难，而且，没有足够的时间去学习……根据我们多年的教学经验，只要会中文，可以阅读中文书籍，就能够看懂电脑的中文应用界面，培养基本的电脑技能，并逐步地熟练。只要你能定期抽出一个小时的完整时间，认真地实践我们提供的技能培养计划，就一定可以成功地驾驭电脑，并可以体验学习新知识的快乐。

科学安排，学会不难

我们把常用的电脑技能，分解成一个一个的学习单元。只要能定期抽出一个小时的空余时间，按照本书的安排，学习其中一个单元，一个小时一点进步，一个小时一点提高。由慢到快，电脑技能很快就可以上一个新的台阶。

按照我们的学习安排，只要 24 小时，一定可以掌握一种电脑应用技能。这个时候，学习的流程安排和内容就相当重要。

根据作者多年的经验，我们在这 24 个小时里面的每一个小时，或者安排读者学习某种技能；或者让读者跟我们学做某个实例；或者让读者强化训练某项技能。这 24 个小时的安排串联起来，就是一张电脑技能的学习地图，它伴随读者探索电脑奥秘的全过程。加上一定时间的训练，一定能教会读者应用电脑，并熟练起来。

按图索骥，提高更快

针对任何一项电脑技能的学习，24 小时培养计划，犹如学习中的 24 级台阶，由作者精心设计。读者可按这个学习顺序，由浅入深，由易到难，逐步掌握好有用的电脑技能。

学习是一个由慢到快的过程。每个人的情况不一样，一般来说，前面的基础打好了，后面的学习速度就会越来越快。所以，在一些内容的安排上，

我们遵循了这个特点。在最后的几个小时的学习计划中，学习内容具有并列特性，读者可根据自己的需要选择学习的顺序。

另外，作为正文的补充，有的图书我们还提供了附录，供读者查询某些资料。

边学边练，事半功倍

学习电脑技能，还要讲究一定的技巧。有了完美的学习方案，还得有足够的练习。

根据我们的经验，电脑技能的学习，上机练习非常重要。所以，建议读者在学习的过程中，同时找一台电脑练习所学内容。

一本图书，一台电脑，一边学习，同时按书中所讲进行练习，可加深印象，更能巩固技能，越用越熟练，越用越体会到使用电脑的乐趣。希望我们的每一本书，加上读者的 24 小时自我训练，能使读者的电脑水平在某一个方面得到飞快地提升。

联系作者，答疑解惑

每一个读者，都有不同的基础和学习经验。我们虽然设计了大多数读者的学习地图，但由于每位读者电脑配置不一定相同，学习碰到的问题也可能各不相同。所以，除了本书之外，我们特地开辟了读者答疑邮箱：jb18803242@yahoo.com.cn。

如果读者在应用电脑的过程中碰到疑难问题，可以发邮件给我们，我们很乐意为您解答，并将典型问题放在下一版的图书中。

编 者
2008 年 3 月

前言

C++是由 C 语言进化而来的，保留了 C 语言的很多优点，比如，更接近机器的思考逻辑，方便编写计算机底层程序。但更重要的是提供了面向对象这一更符合人们思考逻辑的形式。这一形式也是所有现代编程语言的特色。所以，从某种程度上说，学好了 C++，基本上所有的编程语言都能举一反三，轻松掌握。但某种程度上 C++ 又是如此难以掌握，因为 C++ 提供了面向对象和面向过程的混合编程模式，个别从 C 语言继承来的语法又是如此晦涩难懂。结合多年教学实践经验，我们尝试把众多的重点难点分散在各个学时中，以一步一个台阶的阶梯式教学方法逐步推进，让读者逐步掌握 C++ 最基本的知识，为后续的学习和开发工作打下良好基础。我们把全书分为 24 学时，整体内容上分为三部分：

C++语法基础篇

基本数据类型，变量的定义与使用，基本运算符和程序代码表达式，顺序、选择和循环三种基本程序逻辑结构，函数的定义和使用。这些基本上任何一种程序设计语言的语法基础，C++ 也不例外，我们集中在前 8 学时中讲述：

第 1 学时：介绍 C++ 的特色，更重要的是提供 C++ 学习经验。

第 2 学时：介绍 C++ 程序的组成方式，从一个最简单的程序开始，介绍 C++ 程序的基本结构。

第 3 学时：程序设计，说到底还是对各种数据的操作，本学时介绍 C++ 中各种数据常量和数据变量的定义和使用方法。

第 4 学时：程序代码由一条条语句构成，一条条语句由各种变量及运算符构成表达式，本学时介绍 C++ 中的运算符和表达式。

第 5 学时：主要讲述 3 种程序结构中的顺序和选择结构。

第 6 学时：主要讲述 while、do...while、for 这 3 种循环结构的使用方法。

第 7 学时：几行到几十行程序代码完成某个特定功能，并可以被任何程序调用，这种机制叫作代码复用。C/C++ 中的函数机制就是满足这种功能的语法。

第 8 学时：C++ 在 C 语言的基础之上，提供的一些更有特色的函数机制和内联函数、带缺省参数值的函数等，都在本学时的函数的高级话题中讲述。

基于过程的程序设计机制

在基本数据类型的基础上，C++ 提供了数组、结构体等构造数据类型，加上指针这个特殊机制，一起构成了基于过程的程序设计机制。

第 9 学时：介绍了数组的基础知识。

第 10 学时：介绍了字符数组和多维数组的概念。

第 11 学时：开始进入 C++ 精华之一的指针的学习。

第 12 学时：进入指针技术的高级话题。

第 13 学时：引入了 C++ 中的重要概念“引用”，并与指针进行了对比，有助于大家掌握这两个重要的 C++ 工具。

第 14 学时和第 15 学时：介绍结构体、共用体等重要构造数据类型。

第 16 学时：综合前面的知识，介绍数据结构和算法的重要结构——链表。

面向对象的程序设计方法

面向对象的程序设计才是 C++ 的精髓。封装、继承和多态是三个基本概念，我们按照逐步掌握这 3 个概念的原则逐步推进。

第 17 学时：引入 **class** 关键字，介绍类和对象，建立基本的概念。

第 18 学时：通过构造函数、**this** 指针的介绍，强化类的封装概念。

第 19 学时：继承特性的介绍。

第 20 学时：C++ 的多态机制。

第 21 学时：C++ 的运算符重载机制，让我们学会更自然地使用类及其方法。

第 22 学时和第 23 学时：介绍 C++ 的各种输入/输出文件机制。

第 24 学时：一个专门的小型项目，教读者学习使用 C++ 开发的基本方法。

本书特色：适合教学，满足等级考试需要，便于自学，快速掌握开发方法

本书根据 C++ 教学大纲和全国计算机等级考试新大纲规定的二级 C++ 考试内容的要求编写而成。全书共 24 学时，系统介绍了 C++ 面向对象程序设计的基本概念和方法。

在编写本书过程中，作者结合多年从事 C++ 教学的经验，理论联系实际，力求通俗易懂。本书在体系结构安排上尽可能将概念、知识点与例题结合起来，每学时的开始都给出了导读，最后都有小结。在例题的选择上具有针对性强的特点，通过一些典型程序将前后的一些知识点联系起来，使读者在对比中理解各种实现方式的特点和异同，能够融会贯通、举一反三。每学时附有习题，让学生加深和巩固所学知识，提高编程能力，培养良好的编程风格和习惯。

本书系统全面、例题丰富、图文并茂。通过对本书的学习，读者能快速掌握 C++ 的编程方法和技巧，得心应手地解决实际问题，不仅可全面掌握 C++ 程序设计的方法与开发技术，同时能充分满足参加全国计算机等级考试二级 C++ 考试的需要。

编写团队：一线专业教师和开发人员组成

本书由匡松主编，缪春池、吕峻闽、张承虎、谢志龙、陈松明执笔编写，查克玮、谢帮伟、胡长清、张小刚、赵华生、张巍、刘小麟、刘莹、徐畅畅等也参加了本书的编写工作。

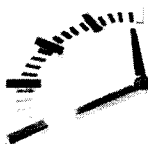
本书可作为高等学校学生学习 C++ 程序设计的教材，也可作为计算机培训班的教材及参加全国计算机等级考试二级 C++ 的考试人员或编程初学者的自学用书。

编 者

2008 年 4 月

目 录

第 1 学时 C++学习概述	1
1-1 C 语言的发展	1
1-2 C 语言的特点	2
1-3 C++的诞生	3
1-4 C++的特点	3
1-5 面向对象程序设计概述	3
1-6 安装 Visual C++ 6.0	5
1-7 启动 Visual C++ 6.0	8
1-8 退出 Visual C++ 6.0	8
1-9 Visual C++ 6.0 程序开发环境介绍	9
1-10 C++新手常见问题	9
1-11 C++学习方法指导	10
1-12 本学时总结	10
1-13 实战练习	10
第 2 学时 C++程序的组成和运行	12
2-1 学习第一个 C++程序	12
2-2 C++程序的运行过程	14
2-3 C++程序的构成	15
2-4 函数的组成	17
2-5 源程序的书写格式	17
2-6 数据的输入/输出	17
2-7 头文件	24
2-8 本学时总结	25
2-9 实战练习	25
第 3 学时 基本数据类型	27
3-1 C++语言的数据类型	27
3-2 常量	28
3-3 整型常量	30
3-4 浮点型常量	31
3-5 字符型常量	31
3-6 字符串常量	32
3-7 变量	33
3-8 整型变量	34
3-9 浮点型变量	36



3-10	字符型变量.....	37
3-11	布尔型数据.....	39
3-12	本学时总结.....	39
3-13	实战练习	40
第 4 学时	运算符与表达式.....	41
4-1	表达式的组成.....	41
4-2	运算符的分类.....	41
4-3	算术运算符与表达式.....	42
4-4	关系运算符与表达式.....	44
4-5	逻辑运算符与表达式.....	45
4-6	位运算符与表达式.....	46
4-7	赋值运算符与表达式.....	49
4-8	条件运算符与表达式.....	50
4-9	逗号运算符与表达式.....	51
4-10	强制类型转换运算符与表达式.....	51
4-11	运算符的优先级.....	52
4-12	本学时总结.....	53
4-13	实战练习	54
第 5 学时	程序的控制结构.....	56
5-1	顺序结构	56
5-2	选择结构	57
5-3	循环结构	58
5-4	条件选择语句——if 语句.....	58
5-5	多路分支语句.....	62
5-6	本学时总结.....	64
5-7	实战练习	64
第 6 学时	循环语句.....	67
6-1	循环语句简述.....	67
6-2	while 型循环语句	67
6-3	do...while 型循环语句	68
6-4	for 型循环语句	70
6-5	continue 语句和 break 语句.....	72
6-6	循环的嵌套.....	74
6-7	本学时总结.....	76
6-8	实战练习	76
第 7 学时	函数.....	79
7-1	函数的定义.....	79
7-2	函数的调用.....	80
7-3	函数调用中的数据传递.....	83

7-4 变量的作用域	87
7-5 变量的生存期	90
7-6 内部函数	92
7-7 外部函数	93
7-8 本学时总结	94
7-9 实战练习	94
第 8 学时 函数的高级应用	97
8-1 内联函数	97
8-2 带缺省参数值的函数	98
8-3 带命令行参数的 main 函数	99
8-4 函数的嵌套调用	100
8-5 函数的递归调用	101
8-6 本学时总结	103
8-7 实战练习	103
第 9 学时 一维数组	106
9-1 数组概述	106
9-2 一维数组的定义	106
9-3 一维数组的初始化	107
9-4 一维数组元素的使用	107
9-5 数组和函数结合应用举例	111
9-6 本学时总结	113
9-7 实战练习	113
第 10 学时 字符数组与二维数组	117
10-1 字符数组的定义	117
10-2 字符数组的初始化	117
10-3 字符数组的使用	118
10-4 字符串处理函数	120
10-5 字符串类 string	123
10-6 二维数组的定义	125
10-7 二维数组的初始化	126
10-8 二维数组的使用	127
10-9 本学时总结	130
10-10 实战练习	130
第 11 学时 指针	133
11-1 地址、指针与指针变量	133
11-2 指针变量的使用	135
11-3 指向一维数组的指针	137
11-4 指向二维数组的指针	141
11-5 指向字符串的指针	145

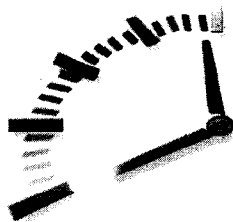


11-6	本学时总结	146
11-7	实战练习	147
第 12 学时	指针的高级应用	149
12-1	指针数组	149
12-2	指向指针的指针	150
12-3	指针变量作为函数参数	151
12-4	返回指针的函数	153
12-5	指向函数的指针	154
12-6	本学时总结	157
12-7	实战练习	158
第 13 学时	引用与内存管理	160
13-1	引用的定义及简单使用	160
13-2	引用作为参数	161
13-3	分配内存 new 运算符	163
13-4	释放内存 delete 运算符	164
13-5	C++ 中指针和引用的区别	165
13-6	本学时总结	166
13-7	实战练习	166
第 14 学时	结构体	169
14-1	结构体的定义	169
14-2	结构体变量的使用	171
14-3	结构体数组	172
14-4	指向结构体的指针	173
14-5	结构体变量作为参数在函数之间进行传递	175
14-6	本学时总结	176
14-7	实战练习	176
第 15 学时	共用体与枚举类型	178
15-1	共用体的定义	178
15-2	共用体变量的使用	179
15-3	枚举类型的定义	181
15-4	枚举类型变量的使用	181
15-5	自定义类型	182
15-6	本学时总结	184
15-7	实战练习	184
第 16 学时	指针与链表	186
16-1	链表的定义	186
16-2	链表的操作	187
16-3	应用举例	192
16-4	本学时总结	193
16-5	实战练习	193

第 17 学时 类与对象.....	195
17-1 对象 (Object) 的概念.....	195
17-2 抽象 (Abstraction) 的概念.....	196
17-3 封装 (Encapsulation) 的概念.....	197
17-4 继承 (Inheritance) 的概念.....	198
17-5 多态性 (Polymorphism) 的概念.....	198
17-6 类的定义.....	199
17-7 类的数据成员.....	200
17-8 类的成员函数.....	201
17-9 对象的定义.....	205
17-10 对象成员的引用.....	206
17-11 本学时总结.....	207
17-12 实战练习.....	207
第 18 学时 类与对象的高级应用.....	209
18-1 构造函数.....	209
18-2 析构函数.....	213
18-3 静态数据成员.....	214
18-4 静态成员函数.....	216
18-5 对象指针.....	218
18-6 this 指针.....	219
18-7 友元.....	220
18-8 对象数组.....	221
18-9 本学时总结.....	222
18-10 实战练习.....	223
第 19 学时 继承与派生.....	224
19-1 基本概念.....	224
19-2 派生类的定义.....	224
19-3 派生类成员的访问属性.....	226
19-4 派生类的构造函数.....	232
19-5 虚基类的定义.....	234
19-6 虚基类的构造函数.....	235
19-7 虚基类的应用实例.....	235
19-8 本学时总结.....	237
19-9 实战练习.....	238
第 20 学时 多态与虚函数.....	239
20-1 多态性的概念.....	239
20-2 简单的多态性实例.....	239
20-3 虚函数的定义.....	240
20-4 纯虚函数.....	243



20-5	抽象类.....	244
20-6	本学时总结.....	247
20-7	实战练习.....	248
第 21 学时	运算符重载.....	249
21-1	基本概念.....	249
21-2	运算符重载的定义.....	249
21-3	重载运算符的规则.....	255
21-4	本学时总结.....	256
21-5	实战练习.....	256
第 22 学时	输入/输出流.....	257
22-1	输入/输出流简介.....	257
22-2	标准输出流对象 cout.....	258
22-3	标准错误输出流对象 cerr.....	259
22-4	自定义格式输出.....	260
22-5	标准输入流对象 cin.....	263
22-6	cin 的成员函数 get().....	264
22-7	cin 的成员函数 getline().....	265
22-8	本学时总结.....	266
22-9	实战练习.....	266
第 23 学时	文件流.....	267
23-1	文件流的概念.....	267
23-2	文件的打开与关闭.....	267
23-3	ASCII 码文件的读写.....	269
23-4	二进制文件的读写.....	271
23-5	应用程序举例.....	273
23-6	本学时总结.....	274
23-7	实战练习.....	274
第 24 学时	编译预处理和项目开发.....	276
24-1	编译预处理.....	276
24-2	类的定义.....	277
24-3	类的实现.....	278
24-4	主程序调用.....	279
24-5	本学时总结.....	281
24-6	实战练习.....	281
附录 A	实战练习参考答案.....	283



第 1 学时 C++学习概述

主要介绍 C 语言和 C++的发展历史、主要特点以及 C++语言的产生，同时介绍面向对象程序设计的基本概念。最后通过图解介绍 Visual C++ 6.0 的安装、启动与退出。

本学时导读

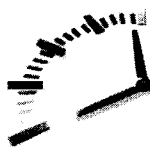
- C 语言的发展
- C 语言的特点
- C++的诞生
- C++的特点
- 面向对象程序设计概述
- 安装 Visual C++ 6.0
- 启动 Visual C++ 6.0
- 退出 Visual C++ 6.0
- Visual C++ 6.0 程序开发环境介绍
- C++新手常见问题
- C++学习方法指导
- 本学时总结
- 实战练习

1-1 C 语言的发展

C 语言是 1972 年由美国的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL (Basic Combined Programming Language) 发展演变而来。

在 1970 年，AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 设计出较先进的取名为 B 的语言，并用 B 语言写了第一个 UNIX 操作系统。1972 年，贝尔实验室的 Ritchie 在 B 语言的基础上设计出了 C 语言，取的是 BCPL 的第二字母。到了 1973 年，两人又用 C 语言一起改写了 UNIX 操作系统 90%以上的代码，也就是 UNIX 第 5 版本。1978 年，Ritchie 和 Brain Kernighan 编写了《The C Programming Language》，并于 1988 年作了修订，该书就是 C 语言版本的基础，被称为标准 C。

随着计算机的日益普及，出现了许多 C 语言版本，正是 C 语言的高可移植性的特点，使得 UNIX 系统在 AT&T、VAX 等计算机上能够迅速实现。但是由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，美国国家标准化协会 (ANSI) 为 C 语言制定了一套 ANSI 标准成为现行的 C 语言标准，包括了 1983 年和 1987 年两个版本。到了



1990 年, 国际标准化组织 (International Standard Organization, ISO) 接受了 1987 年公布的新标准 87 ANSI C 作为 ISO C 的标准, 即 ISO 1989—1990。目前比较流行的 C 语言的编译系统都是以此作为基础, 如 Turbo C、Borland C、Microsoft C 等。

1-2 C 语言的特点

C 语言发展如此迅速, 而且成为最受欢迎的语言之一, 主要因为它具有强大的功能。C 语言的特点如下:

(1) C 语言提供的语句简洁, 使用方便, 格式紧凑, 语法灵活。C 语言共有 32 个关键字, 9 种控制语句, 语句简练, 书写自由。以下关键字是由系统定义的, 不能用作其他定义。

与数据类型相关的关键字有:

char	int	short	long	signed	unsigned	float
double	enum	struct	union	typedef	void	

与存储类型相关的关键字有:

auto	register	static	extern
------	----------	--------	--------

与控制语句相关的关键字有:

if	else	switch	case	default	do
while	for	break	continue	goto	return

其他的关键字有:

const	sizeof	volatile
-------	--------	----------

下面是 C 语言中的 9 种控制语句:

if(...)	...else ...	switch(...)	... case	for(...)	... while(...)	...
do ...	while(...)	continue		break	goto	return

(2) C 语言的运算符十分丰富, 共有 34 种运算符, 有算术、关系、逻辑、位、赋值、指针、条件、逗号、下标、类型转换运算符等多种类型。

(3) 数据结构多样, 有整型、浮点型、字符型、枚举类型等基本类型, 有数组、结构体、共用体等构造类型以及指针类型, 还为用户提供了自定义数据类型, 特别是引入了指针概念, 能够实现复杂的数据结构。

(4) C 语言的控制语句形式多样、使用方便。有两路分支、多路分支和循环结构几种控制语句, 便于结构化模块的实现和控制, 便于程序的编制和维护。

(5) C 语言是一种模块化的程序设计语言, 采用自顶向下、逐步求精的结构化程序设计方法, 各模块功能独立, 以函数形式编制, 通过函数之间的相互调用和数据传递, 实现系统整体的功能要求。这样, 把大型系统的实现化整为零, 便于分工合作以及共享。

(6) C 语言可以直接访问地址、进行位运算, 从而能对硬件进行操作, 因此 C 语言既具有高级语言编写简单方便、便于理解的优点, 又具有低级语言与硬件结合紧密的优点。因此 C 语言被称为介于高级语言和低级语言之间的中级语言。

(7) C 语言具有很强的移植性, 由 C 语言编写的程序基本不用太多的修改就可以用于不同型号的计算机上, 程序和硬件的匹配由 C 语言的编译程序完成, 同时也可以在各种操作系统下使用。

(8) C 语言具有很好的通用性, 既可以用于编写应用软件, 也适合编写系统软件。例如 UNIX 操作系统的源代码就是用 C 语言编制的。

1-3 C++的诞生

C 语言、Pascal 语言等面向过程的结构化程序设计语言适合处理比较小规模的程序，随着计算机技术的发展、用户软件需求的不断增长以及软件规模的不断扩大，对于软件规模较大、用户需求较复杂的问题就力不从心了。上个世纪八十年代，计算机科学家提出了面向对象的程序设计方法（Object Oriented Programming, OOP），在这种背景下，仍然是来自 AT&T 贝尔实验室的 Dr. Bjarne Stroustrup 在 C 语言的基础上开发出了 C++ 语言。

和 C 语言相比，C++ 保留了原来 C 语言所有的特点和优点，即向下兼容所有 C 语言的程序，是在 C 语言的基础上增加了面向对象程序设计的方法，增加了“类”和“对象”的概念。

结构化程序设计的精髓就是“自顶向下、逐步求精”，具体来说是将一个程序划分成若干个能够完成独立功能的模块，在模块中采用顺序、选择和循环这三种基本结构实现。在面向对象程序设计中，首先分析具有相同类型的事物，将其中共同的属性和特点抽象成一种新的数据结构——“类”，然后定义类的实例——“对象”，在对象中封装了属性和方法，通过接口与程序进行数据的交换等操作。

C++ 新增加了以下关键字：

new	delete	public	private	protected
class	template	this	inline	virtual
overload	friend	catch		

1-4 C++的特点

C++ 的主要特点表现在两个方面，一个是全面兼容 C 语言，另一个是支持面向对象的方法。

首先，C++ 的确是一个更好的 C 语言。它保持了 C 语言的简洁、高效和接近硬件等特点，对于 C 语言的类型系统进行了改革和扩充，因此 C++ 比 C 语言更安全，C++ 的编译系统能检查出更多的类型错误。

由于 C 语言已经被广泛使用，因而 C++ 与 C 语言的全面兼容使得 C++ 的普及变得异常容易，同时极大地促进了面向对象技术的广泛应用。同时，也正是由于对 C 语言的全面兼容，使得 C++ 不是一个完全的面向对象的语言。它是既支持面向过程又支持面向对象的程序设计。

C++ 最有意义的方面是支持面向对象的特点，我们应该注意按照面向对象的思维方式去编写面向对象的程序。掌握 C++ 的语法并不代表掌握面向对象的技术，这个世界上使用 C++ 却不懂面向对象的人很多；C++ 仅仅是支持面向对象，也就是说他的语法包含面向对象的特点，但面向对象不是语言特质，而是一种思想；C 语言不是面向对象的语言，但用 C 语言一样能写出具有面向对象思想的程序。

如果读者已经有其他面向过程高级语言的编程经验，那么学习 C++ 语言时应该着重学习它的面向对象的特征，对于与 C 语言兼容的部分只需要了解一下就可以了。如果读者是初次学习编程，那么，依然不能越过与 C 语言兼容的部分。像数据类型、算法的控制结构、函数等，它们不仅仅是面向过程程序设计的基础，同时也是面向对象编程的基础。

1-5 面向对象程序设计概述

尽管类与对象的概念相当简单，但是功能却很强大，简单来讲，面向对象程序设计语言的特点就是抽象、封装、继承和多态。



“抽象”是从具体事物中提取共同的性质，包括对数据的抽象和行为的抽象。数据抽象就是对事物属性的描述，比如圆所共有的性质有圆心、半径等；行为抽象就是对事物所进行操作的抽象，比如圆共有的操作有计算周长、计算面积。

“封装”是把对象相关的数据和操作结合起来，将对象的属性以及操作实现的代码隐藏起来，通过对外提供的接口进行控制。用户不需要知道如何计算圆周长，只要将半径作为参数就可以得到结果。

“继承”是在原有类（父类或者基类）的基础上建立一个新的类（子类或者派生类），通过继承，派生类得到了基类的属性和操作，因此在派生类中只需要定义基类没有的属性和操作。

“多态”是指当不同对象在收到同一消息时会产生不同的操作。C++支持两种多态性，一种是编译时通过函数重载和操作符重载来实现的，称为“静态多态性”；另一种是在运行时通过虚函数来实现的，称为“动态多态性”。

1. 对象（Object）的概念

面向对象的程序设计，顾名思义，程序设计的中心是对象，对象也就是程序可以控制的运行实体，程序依靠对象的特定动作来完成特定的任务。利用面向对象的程序设计方法，把程序和数据封装起来作为一个对象，并为每个对象赋予属性。

现实生活中各种不同的物体都可以看作是不同的对象，选择其中的哪些进行处理，根据不同人的需要来定。例如学校管理者的管理对象主要是学生，建筑师描述的对象主要是建筑体，工厂质检员的工作对象主要是生产出来的产品等。

而选择对象的目的是为了对所选对象进行控制和处理，那么这个控制和处理过程又是如何进行的呢？比如学生对象具有各种不同方面的特征，在此称为属性，包括姓名、学号、所学专业、学习成绩、兴趣爱好、基本生活需求等。那么学校要完成对学生的有序管理，就要从学生的一些普遍共性和不同个性出发，从学习、生活中的不同方面入手实施管理，针对学生的学习，主要由教务处来安排其学习课程的内容、进度和时间；针对学生的基本生活需求，主要由后勤部门安排其住宿、就餐、洗衣等事项；针对学生的阅读需要，由图书馆进行图书的管理并制定借阅规则供学生阅读图书。

由此可见，针对对象的处理是根据对象的特征制定相应的控制过程来达到目的的，这样的原理同样适用于编程过程中的对象设计。在面向对象的程序设计中，对象的特征被称为属性，针对对象的控制过程和相关操作被称为事件和方法，这三者称为描述对象的三个基本特征。建立一个对象后，针对对象的操作可以通过与该对象有关的属性、事件和方法来描述，再把对象和描述对象的属性值以及描述对象行为方式的代码结合成为一个整体，如此动静结合，完成对象功能的描述，进而使多个对象相互配合，从而实现程序的功能。

2. 类（Class）的概念

对象可以是某个甚至某些具体的实体，比如某一个学生或者某个学校的所有学生，但如果把学生这一大类群体抽象出来，就被称为“类”。可以说“类”是对同一种对象的抽象，是对既具有共性又具有个性的同种对象中共性的提取。例如不同大小、不同使用范围的球（足球、排球、网球、乒乓球等）可以被列为“球体”这一类，其共性是球表面每一点到球心的距离均相等；不同区域、不同国家和地区、不同规模和名称的城市，也可以列入城市这一类，城市的基本特征、