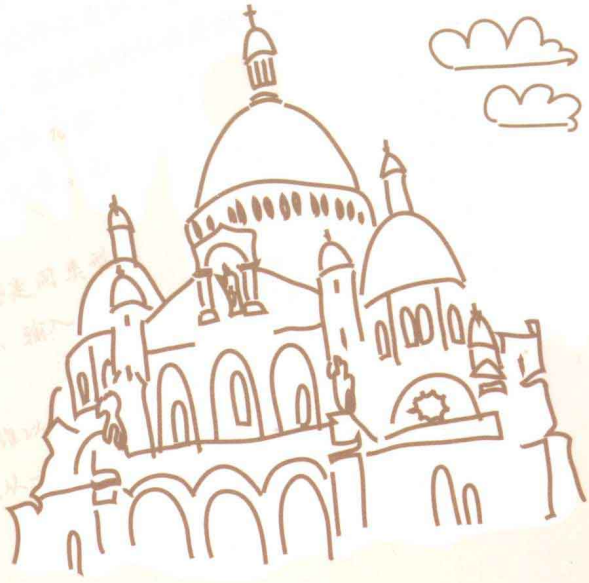
一样的语言，
不一样的学习方法



好学的 C++ 程序设计

张祖浩 沈天晴 编著

深奥的概念**通俗化** 通俗的概念**实例化**，**降低**学习 C++ 的**门槛**
实用的示例，**完整**的代码，为学习者**量身打造**的案例
复杂问题**图表化**，**易学**、**易用**、**易于实践**

人民邮电出版社
POSTS & TELECOM PRESS

好学的C++程序设计

张祖浩 沈天晴 编著

人 民 邮 电 出 版 社
北 京

图书在版编目（CIP）数据

好学的C++程序设计 / 张祖浩，沈天晴编著. -- 北京 : 人民邮电出版社，2012.8
ISBN 978-7-115-28309-2

Ⅰ. ①好… Ⅱ. ①张… ②沈… Ⅲ. ①C语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第098928号

内 容 提 要

本书内容与大学课程“C++程序设计”基本相同，内容包括基本数据和别名指针、运算、流程控制、函数、数组、枚举类型与结构类型、类和对象、继承与派生、多态性、输入/输出流和上机操作等。

“C++程序设计”是高校普遍开设的一门计算机核心基础课程，同时也是一门非常难懂的课。为此，作者采取了多方面的革新措施，作出独到阐述，达到增强系统性、提高可读性的目的，驱除团团迷雾，化解层层难点。使其既进得了中学生书房，使中学生能先修登上“C++殿堂”；又上得了大学生课堂，使大学生对难点迷雾豁然开朗。

本书多方面的革新措施与独到阐述（详见前言）显示了本书的如下特色：

1. 章节内容编排新颖，系统性强；　2. 深奥概念通俗比喻，豁然开朗；
3. 难点问题精心铺垫，化难为易；　4. 复杂问题配图制表，一目了然；
5. 理论问题阐述独到，清晰透彻；　6. 相关概念前后呼应，一脉相承。

本书可用作中学生先修或选修课程“C++程序设计”的教学用书，可用作中学生兴趣小组的学习用书，也可用作中学生自学“C++程序设计”用书或课余读物。

本书还可用作高等院校本、专科相关专业学习“C++程序设计”课程的教学用书，或编程爱好者自学用书及教学参考书。

好学的 C++程序设计

◆ 编　　著　张祖浩　沈天晴
责任编辑　张　涛
◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市潮河印业有限公司印刷
◆ 开本：787×1092　1/16
印张：20.75
字数：513 千字　　2012 年 8 月第 1 版
印数：1- 3 500 册　　2012 年 8 月河北第 1 次印刷

ISBN 978-7-115-28309-2

定价：45.00 元

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

一、编写本书的背景

2011 年伊始，中小学教学改革、高考改革波涛汹涌。在这场改革大潮中，媒体报道，一些地方的高中拟开大学先修课。这反映了中学教学改革在加强素质教育中，要有意着力培养学生的专业兴趣、爱好和专长，展现学生的专业发展潜力，促使其早日成才。

媒体又报道，自主招生的大学复试时增加了学科专业面试。这反映了高考将测试学生在所选专业方面的兴趣、爱好、专长和发展潜力。

教育部办公厅已发出通知，明确指出 2012 年高校自主招生对象主要是具有学科特长，以及全面发展且具有创新潜质的考生。

以上情况必将会激发中学生先修或在课余、兴趣小组中学习与志趣相关的大学课程。

计算机的应用已遍及各行各业，而“C++程序设计”是高校普遍开设的一门计算机核心基础课程。作者就是在上述背景下，编写了这本通俗易懂的 C++先修读物《好学的 C++程序设计》，以满足中学生的知识发展需求。

二、本书的革新措施和特色

“C++程序设计”是一门难懂的语言。为达到通俗易懂，能为中学生读者所接受，同时也帮助大学生化难为易，作者为本书的编写作了一番努力，以求既进得了中学生书房，使中学生能先修登上“C++殿堂”；又上得了大学生课堂，使大学生对难点迷雾豁然开朗。

要达到这两点，关键在于革新。作者采取了多方面的革新措施并作出独到阐述，达到增强系统性、提高可读性的目的，驱除团团迷雾，化解层层难点，现略举数例革新如下：

（一）对章节之间的总体编排作了调整，增强了内容的目的性和系统性。例如：

（1）将别名和指针提前放在第 2 章，讲完基本数据后就讲，目的是对数据变量作访问。这正当其时，并且为后续各章的使用作好铺垫，使后续各章随时需要都能用上别名和指针。

（2）将指针的整数加减运算放在专讲运算的第 3 章，这也适得其所。免得临到要用时才现讲，撇下主题而“临阵磨刀”，造成“节外生枝”、“喧宾夺主”的尴尬。

（3）讲完函数、模块化程序设计以后，讲数组、枚举、结构、链表时，都明确最终目的，就是要落实到调用函数对这些数据进行操作，以实现模块化程序设计。这样，既增强了前后内容的连贯性，又增强了这些自定义数据大片内容讲授的目的性。有了明确目的，教师就能讲得头头是道，同学就能听得津津有味，并能

为实现目的而自觉动脑。

（4）在第 7 章枚举类型中，专门添加了一小节（7.1.4）枚举元素的组合状态。通俗简练阐述了枚举元素的组合状态字，为理解流类中成批的枚举元素和各式状态字打下基础。

（二）在第 1 章中，回避了传统的“真值”、“原码”、“反码”、“补码”等诸多概念，从通俗事物实质出发，轻松阐明了二进制数据的正负表示及其范围。

（三）在第 2 章对变量的存储空间，明确阐明了存储空间的类型、大小、地址、内容，以及存储空间内数据安排的规则等属性，从而建立起存储空间概念。以后一提到存储空间，所有这些属性概念就会随之在脑中涌现，简单明了，轻轻松松学习各项相关内容和概念。

（四）第 2 章初次接触别名和指针，就用通俗事例轻松化解“别名访问”和“指针访问”的玄奥。可使读者感觉到，“别名访问”和“指针访问”司空见惯，实际就经常发生在我们身边。

（五）第 3 章指针整数增减问题通俗比喻为走步，轻松而正确建立了指针整数增减的概念，免除了采用地址运算来说明问题的繁琐。

（六）3.11.3 小节“小小一例竟使多方受益”中，涉及存储空间、地址、指针所指、指针走步、指针强迫类型转换、内存中数据存储等诸多重要概念，为彻底化解有关指针的难点打下了基础。

（七）提出一目了然的表，轻松、明朗而正确地解决了不同类型两个操作数运算时的隐性类型转换问题，以及如何规避错误的问题。纠正了一般的难懂、含糊而错误的简单介绍。

（八）明确揭示了同名成员函数在直接受访时的“接访”规则。“接访”规则概括了虚函数和非虚函数，构造函数或析构函数的运行期间和非运行期间的情况。当遇到直接访问同名成员函数时，接访规则将成为回答由哪个同名成员函数出面接受访问的理论根据。

（九）本书对复杂问题常采用图形表达。例如右图所示，可以清晰简明表达下列几方面概念（设 s 为 S 类对象）。

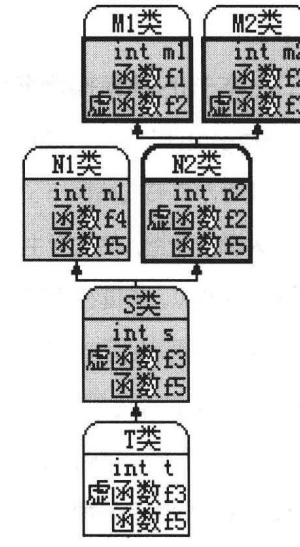


图　多级多脉继承例

（1）继承和派生关系；（2）各类的自设成员；（3）虚函数和非虚函数分布；（4）同名成员和非同名成员分布；（5）带浅灰色的成员是对象 s 所拥有的全部成员；（6）粗黑线框中的成员是对象 s 成员中的原基类 N2 成员组的成员；（7）指向派生类对象 s 的基类 N2 指针只能访问粗黑线框中成员；（8）因这基类指针 N2 所访问的成员是属于派生类对象 s 的，故对象 s 是这些成员的当前对象；（9）从任何类都可沿继承路径，一目了然地寻找其可见同名成员。

用这样的图，阐述派生类成员组成、赋值兼容、同名覆盖以及同名成员函数们在被直接访问时的“接访”规则等概念问题，配以灰色和粗黑线框，可将概念讲解得一清二楚，突显其优越性。

（十）在第 11 章输入/输出流的开始，作者由分析各种流的共有属性，明确提出“端口”和“模式”两者作为流类的数据成员，讲清楚这两者的含义以及如何表示。然后，以端口、模式、成员函数作为主线，展开 I/O 流全部内容。这样，流类的概念就和前面所讲的类的概念取得了一致，都具有数据成员和成员函数。从而使流类对象的声明和初始化，以及经由对象调用基类或派生类成员函数等情况完全和前面所学的内容接轨了。用前面的理论解决流类的具体问题，眉目清楚，概念准确。相关问题前后呼应，一脉相承。这就很容易说清楚：各种各样流的处理方法实质都一样，只是端口、模式、成员函数有具体不同而已。这样就达到了前后内容共促共赢，驱除了团团迷雾，化解了层层难点。

以上只略举 10 例，实际革新措施多多，无法全面一一列出。但窥 10 例可知全貌，全书的革新措施显现出本书所具有的特色。

三、本书的使用

本书可用作中学生先修“C++程序设计”的教学用书，可用作中学兴趣小组的学习用书，也可用作中学生自学“C++程序设计”用书。

本书还可用作高等院校本、专科相关专业学习“C++程序设计”课程的教学用书，或自学用书及教学参考书。

希望在学习中要多做题，多上机操作。关于上机操作的内容集中放在第 12 章。可根据内容的需要，阅读此章相关内容，或随时上机操作，随时翻阅此章。

本书在附录中有习题参考答案，但不包括概念题答案。因为书中已对概念作了阐述，应在书中阅读求得解答。概念要搞得很清楚，这很重要，不可忽视。

在做题时，先别看答案，自己做，自己学编程序，自己上机操作。当编成一个程序并上机操作通过时，那时心情是最愉快的，一种成就感能使你高兴得手舞足蹈，激动万分！

本书第 1～11 章由张祖浩编写，第 12 章由沈天晴编写。因限于作者水平，书中难免有错误和不妥之处，敬请读者批评指正，作者电子邮箱：zhangzh28@126.com，编辑联系邮箱：zhangtao@ptpress.com.cn。源程序下载网址为：www.ptpress.com.cn。

编者

目　录

目录 Contents

目录 Contents

目录 Contents

第1章 概论

1.1 计算机怎样计数

1.1.1 乒乓球场上的计分牌

我们知道，乒乓球球场上的计分牌有0、1、2、3、4、5、6、7、8和9，共10种牌。计分时，逢十进一，叫做十进制。平常我们用的就是。

我们设想，如果计分牌不是10种，而是0和1两种牌。那该怎么计分呢？那只能逢二进一了，这叫作“二进制”。计算机内部用的是二进制。

我们再设想，如果计分牌有0、1、2、3、4、5、6、7、8、9、A、B、C、D、E和F，共16种牌，其中A、B、C、D、E依次代表10、11、12、13、14和15。计分时，逢十六进一，这叫作十六进制。

3种方法的计分情况，见表1-1。从表中可看出，4位二进制数相当于1位十六进制数，它俩一个个相互对应。计分至15时，二进制1111与十六进制F相对应。此时，如果各再增1分，则二者都同时向高位进位，分别进位成为10000和10。这进位的1分是顶十进制16分的。

表1-1 计分表示

十 进 制	二 进 制	十六进制
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	**10000**	**10**

如果用二进制计分牌计分和进位，最后的得分为11101101101101，则表示十进制是多少分呢？我们可从右至左，每4位用一空格隔开，得：0011 1011 0110 1101，分别写出它们所对应的十六进制数得：3 B 6 D。

3B6D是4位十六进制数。此数的每一位满16都要向高位进位。最低位的1分顶1分；高1位则是1分顶16分；高两位则1分顶16×16分；高3位则1分顶16×16×16分……可按此规律，计算得分如下：

十六进制数3 B 6 D = 3×16×16×16 + B×16×16 + 6×16 + D

=12288+11×16×16+96+13=15213（此即十进制得分）

1.1.2 二进制怎样表示全正数和正负数

将表1-1中4位二进制数的全部16个数围成一圈，如图1.1所示。我们设想，在图中A点处把圆周剪开，然后把圆周拉直，竖立起来如图1.2所示。在此图中表示了与十进制数0～15相对应的全正

数的 4 位二进制数。那么，要表明负数怎么办？

如果我们在图 1.1 中的 B 点处把圆周剪开，然后把圆周拉直，竖立起来如图 1.3 所示。此图的上半部 0000～0111 表示了十进制的正数 0～7，下半部 1000～1111 则表示了十进制的负数−8～−1。这样看来，对于有正负的二进制数，其最高位为 0 者为正数，最高位为 1 者为负数。从最高位就能判别数的正负，因此，对于有正负的二进制数，其最高位就被称之为符号位。符号位如图 1.3 中所示。

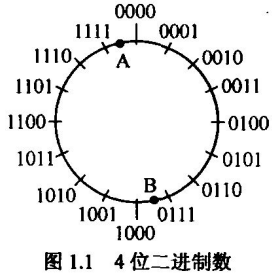


图 1.1　4 位二进制数

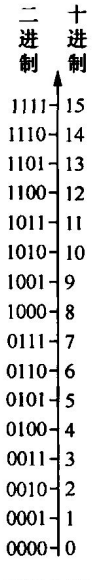


图 1.2　全正数 4 位二进制数

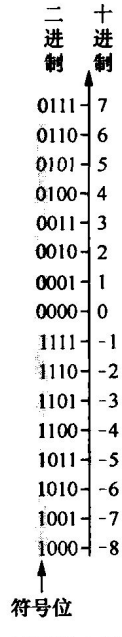


图 1.3　有正负 4 位二进制数

读者会问，图 1.3 中，0000 以下的数分明比上半部的数更大，为何反说是负数呢？我们可认为，这些数个个都背了一笔巨债 10000（二进制）。因此它们不是“富翁”，而是“负翁”。比如，拿负翁 1011 来说，负多少？观察可知，若支援它 0101，则与 1011 相加就是 10000，就可还清债务啦。可见负翁 1011 实际是负 0101，即负 5（十进制）。可见，要问负翁负多少，就看它要还清巨债还差多少。差多少就说明它负多少。

事实上，负数应该是比 0000 小的数，比如负 5，应该由 0000-0101 计算而得。

$$\begin{array}{r} 10000 \\ -\ 0101 \\ \hline 1011 \end{array}$$

但不好减而向高位借位，算式变为 10000-0101，算得结果为 1011。因借位而成为“负翁”。

综上可见，4 位二进制数可表示全正数范围为 0000～1111，即 0～15；正负数范围为 1000～0111，即−8～7。对更多位二进制数可作类似分析。比如，8 位二进制数全正数范围为 00000000～11111111，即 0～255；正负数范围为 10000000～01111111，即−128～127。

1.1.3　计算机中二进制数怎样存储

计算机内部用二进制表示各种数据，每 8 位二进制数称为一个字节。计算机数据存储在内存中，内存由一个个内存单元组成。一个内存单元可存储一个字节，即 8 位二进制数。

1.2 计算机怎样对数据进行处理

1.2.1 把算法形成的程序输入计算机

要让计算机为我们作数据处理，我们必须把数据处理操作的步骤考虑好，交给计算机。目的是要让计算机按我们所考虑的步骤一步步进行操作，最后达到我们的目的。

我们所考虑的，对数据处理操作的步骤叫做算法。

算法必须用某种语言写出来形成一个程序。程序输入计算机，使计算机按程序指令一步步地对数据进行操作，最终实现按算法对数据进行的处理。

那么，当我们对数据处理的算法考虑好了，用什么语言写出程序来输入计算机呢？

1.2.2 用什么语言输入计算机能懂

计算机硬件只能听从由二进制码（0 和 1）组成的指令。由二进制码组成的指令序列叫做机器语言。计算机对机器语言能识别，并能据此直接对内存中的数据进行相应操作。机器语言是低级语言。在计算机诞生初期，专家们是用机器语言写出程序的。

由于机器语言全是二进制码，人们难懂难记，编程费时又费力。后经专家们的不懈努力，设计出与人们习惯语言相近的程序设计语言，这种语言叫做高级语言。

20 世纪 50 年代至 70 年代，FORTRAN、BASIC、ALGOL、PASCAL、COBOL、ADA 和 C 等高级语言相继问世。它们凭借各自的优点，在程序设计中都曾占有一席之地。高级语言的出现大大促进了计算机软件的开发。C++语言是在 C 语言的基础上发展起来的，并且与 C 语言兼容。

用高级语言编写出程序，经编译系统翻译成机器语言，让计算机能识别，并能遵照执行。

1.3 C++程序怎样写出和运行

1.3.1 举个 C++程序简例看看

【例 1-1】 试用 C++语言设计一个程序，输入两个任意整数，输出二者之差的绝对值。

算法：先输入两个整数，然后两个数相减得差 dif。若 dif 为负，则输出-dif；若 dif 为正，则输出 dif。输出的结果就是绝对值。此过程就是算法。根据算法，设计程序如下：

```
#include<iostream>                    //包含头文件 iostream。
using namespace std;                  //使用命名空间 std。
int  main()                           // main 是主函数。程序从主函数开始执行。
{                                     //花括弧之间的内容就是主函数 main 要做的事。
  int a,b,dif;                        //声明 a,b 和 dif 是 3 个整数类型(int)变量。
  cout<<"请输入被减数"<<'\n';          //屏幕显示 "请输入被减数"。'\n' 是换行的意思。
  cin>>a;                             //从键盘对变量 a 输入整数。
  cout<<"请输入减数"<<'\n';            //屏幕显示 "请输入减数"。'\n' 是换行的意思。
  cin>>b;                             //从键盘对变量 b 输入整数。
  dif=a-b;                            //求出 a-b，通过赋值运算符"="赋给变量 dif。
  cout<<"二数之差的绝对值是: ";        //输出 "二数之差的绝对值是:" 至屏幕显示。
  if(dif<0) cout<<-dif<<'\n';         //若表达式 dif<0 为真，则输出 -dif 值。换行。
```

```
    else      cout<<dif<<'\n';                    //否则，输出 dif。换行（'\n'是换行的意思）。
    return 0;                                     //程序正常执行，返回 0。
}
```

运行情况如下：

```
请输入被减数:                                     //屏幕上显示出"请输入被减数:"。
24↙                                               //键盘输入 24，↙表示打一个 Enter 键。
请输入减数:                                       //屏幕上显示"请输入减数:"。
68↙                                               //键盘输入 68，↙表示打一个 Enter 键。
二数之差的绝对值是: 44                            //屏幕上显示所计算的绝对值结果。
```

通过这个简单的例子，我们说明以下几点：

1．注释

程序的右边，以“//”开头写出的一行字是程序的注释。这种注释的写法只限在一行之内有效。如果一个注释超过了一行，则注释必须改为以“/*”开头，以“*/”结尾。

注释的作用是对程序作些解释，以便让人们能看懂程序，即提高程序的可读性。例如，我们把上例程序结合注释来看，基本能看懂。但是，注释是不参与程序运行的。

2．包含头文件和使用命名空间 std

按标准 C++的新要求，程序的头两行用了“#include<iostream>”和“using namespace std”，意思是包含头文件 iostream 和使用命名空间 std。这两行是为程序中所要用的输入和输出提供必要支持的，其道理暂且不去深究。只是要记住，每逢程序设计，开头总少不了要写上这两行。

3．主函数

int main()表示主函数，main 是函数名。int 表示最后返回一个整数。程序总是从主函数 main 开始执行。下面花括弧所括的内容是函数所要做的事，叫函数体。若程序正常执行完毕，最后就向操作系统返回数值 0（即 return 0），否则返回数值−1（即 return −1）。

4．标识符

我们给变量等所取的名字，如程序中变量名 a、b 和 dif，统称为标识符。标识符限定只能由大写字母、小写字母、数字或下划线组成，并且只能以字母或下划线开头。大小写字母有区别。例如，对于_B2d 和_b2d，虽然都是合法的标识符，但因有大小写字母的不同，二者不可视为同一个标识符。

5．关键字

在 C++语言中，已经具有特定含义的词叫做关键字。例如 main、int、if、else，等等都是关键字。我们取名字所用的标识符不可与关键字相同。

6．语句

程序中语句要用分号“;”结尾，分号是语句的组成部分。语句不一定一行只写一句。在一行中写好几句，或一句写成好几行都行。分行写和缩进写只是为求醒目，便于看懂而已。

对于程序中其他标点符号，以后随讲随看随记，逐渐积累是不难的。

7．流程控制

if 和 else 两条语句显然是对程序进行流程控制的，它把程序流程分为两个分支。括弧中的表达式 dif<0 就是作逻辑判别用的，叫判别式。若判别式“dif<0”为真，就执行 cout<<-dif<<'\n';；否则，就执行 cout<<dif<<'\n';。