“十二五”江苏省高等学校重点教材

HZ BOOKS
华章教育

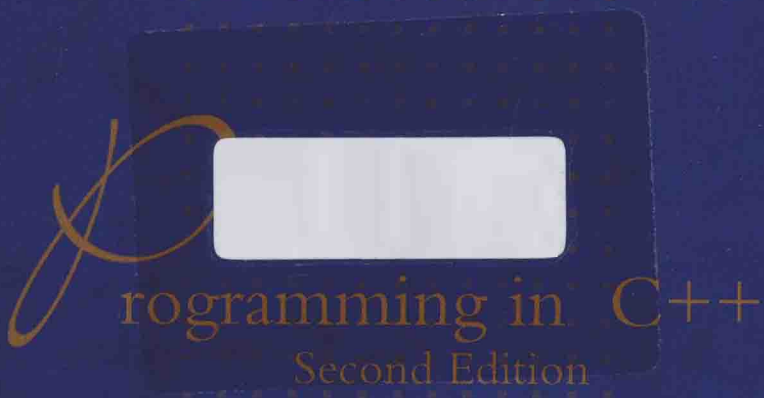
高等院校精品课程系列教材

C++程序设计教程

第2版

皮德常◎编著

精品课程



机械工业出版社
China Machine Press

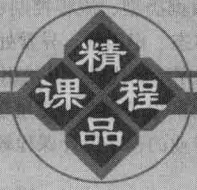
“十二五”江苏省高等学校重点教材

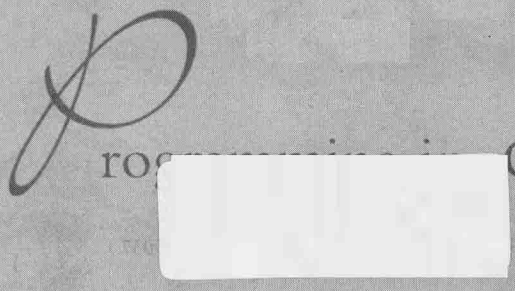
高等院校精品课程系列教材

C++程序设计教程

第2版

皮德常 编著





图书在版编目（CIP）数据

C++ 程序设计教程 / 皮德常编著 . —2 版 . —北京：机械工业出版社，2014.2
（高等院校精品课程系列教材）

ISBN 978-7-111-45476-2

I. C… II. 皮… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字（2014）第 011467 号

C++ 是一种实用的程序设计语言，是高校学生学习程序设计的一门必修专业课程，同时也是编程人员最广泛使用的工具。学好 C++，可以很容易地触类旁通其他语言，如 Java 和 C# 等。

本书针对初学者和自学者的特点，在总结过去教学和实践经验的基础上编写而成。写作风格别具一格，语言流畅、风趣，恰如其分的举例易于读者理解和掌握 C++ 程序设计，同时，在写作中还特别注重培养学生的独立思考能力。教材结合实例讲解了 C++ 的基本概念和方法，力求将复杂的概念用简洁、通俗、有趣的语言来描述，做到了深入浅出、循序渐进，从而使学生体会学习的快乐，以及在快乐中学习。

全书共 12 章，主要包括 C++ 基本数据类型、流程控制、函数、数组、指针、结构体、文件操作、类的基础部分、类的高级部分、继承、多态、虚函数、异常处理、通过 ODBC 对数据库编程、课程设计等。书中列举了数百个可供直接使用的程序示例代码，并给出了运行结果，使学生在学习时更为直观。

本书配有适当的习题，并提供了该书的电子教案，超星学术视频网站还提供了作者的授课录像，特别适合用作大学计算机专业和非计算机专业的程序设计课程教材，也非常适合那些具有 C 编程经验又想转向 C++ 编程的读者阅读。

机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码 100037）
责任编辑：佘 洁
北京诚信伟业印刷有限公司印刷
2014 年 2 月第 2 版第 1 次印刷
185mm × 260mm • 22.5 印张
标准书号：ISBN 978-7-111-45476-2
定 价：39.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991 88361066 投稿热线：（010）88379604
购书热线：（010）68326294 88379649 68995259 读者信箱：hzjsj@hzbook.com

前　言

本书第 1 版受到了众多读者喜爱，许多读者和授课教师纷纷与作者联系，作者受益多多，深表感谢。第 2 版针对初学者和自学者的需求，结合读者反馈和作者近几年科研成果，基于 C++ 最新标准 0x，采用 Visual Studio 2010 对 C++ 的知识点进行了全面的修订和改版。

本书的特点如下：

1）本书主要讲解 C++ 程序设计的编程方法，它是计算机科学与技术专业学生的编程基础。

2）本书是作者教学经验的结晶。作者 18 年来一直从事程序设计方面的教学和科研工作，主讲 C、C++、Java 等程序设计方面的多门课程，积累了丰富的教学经验。“从实践到理论，再从理论到实践，循序渐进”是作者教学的心得体会，编写教材也不例外。作者深知学生学习的薄弱环节和学习特点，将自己的知识、授课方法和教学经验整理成书，使更多的学生受益，是作者的梦想和追求。

3）在内容安排上，本书尽量提前讲解文件操作（许多书都是在最后讲解这部分内容）。因为文件是很实用也是比较难学的一章，所以这种安排也为学生进行课程设计和实验做了铺垫。

4）在作业安排上，从易到难，环环相扣。作者在教学中发现，许多学生学过 C++ 却不会编程。因此，本书设计了许多与实际有关的习题，并且它们彼此相关。

5）强调课程设计。C++ 课程应该有课程设计，我们在本书的最后给出一个课程设计要求，希望学生能独立、认真完成。这对提高学生的编程能力，巩固学过的知识大有裨益。

6）力求通俗易懂。本书的编写目的是让学生通过自学或在教师的讲授下，能够运用 C++ 语言的核心要素，进行程序设计。因此，本书围绕着如何进行 C++ 编程展开。为了便于学生学习，作者力求该书的语言通俗易懂，将复杂的概念采用浅显的语言描述，做到易学、易用、有趣，从而便于学生理解和掌握 C++ 编程思想和方法。

7）强调程序的可读性。本书中的程序全部采用统一的程序设计风格。例如，类名、函数名和变量名的定义做到“见名知义”；采用缩排格式组织程序代码并配以尽可能多的注释。希望学生能够模仿这种程序设计风格。

8）包含大量的程序示例，并给出运行结果。凡是程序开头带有程序编号的程序，都是完整的程序，可以直接在计算机上编译运行。

9）采用醒目的标记来显示知识点。这些标记是注意、警告和思考等，它们穿插全文，帮助学生尽快找到重要的信息。

注意：值得读者关注的地方，也是作者在教学中发现学生容易搞错的知识点。

警告：这是容易混淆的知识点。

思考：提出问题，引导学生思考，以培养思考能力。

本书的电子教案采用 PowerPoint 2003 制作，可以在讲课时用多媒体投影演示，这样可部分取代板书。教师不仅可以使用本教案，还可以方便地修改和重新组织其中的内容以适应自己的教学需要。使用本教案可以减少教师备课时编写教案的工作量，以及因板书所耗费的时间和精力，从而提高单位课时内的知识含量。

我们向使用本书的教师免费提供电子教案，需要本教案的教师可以直接与机械工业出版社联系。

在编写本书的过程中，作者得到了许多同事的帮助，他们是王珊珊、臧冽、张志航、郑洪源、陈丹等，他们给作者提出了许多宝贵的意见和建议。作者的研究生马程、张玉、方卓然、张伟、王强、程冉、李文等人，为本书做了大量的程序验证工作。在教学过程中，作者也得到了许多学生问题的启发，促使作者在写书的过程中，注意有的放矢，便于学生理解和掌握。

感谢您选择本书，欢迎您对本书的内容提出批评和修改建议，作者将不胜感激。作者的电子邮件地址：dc.pi@163.com。

皮德常

2014 年 1 月

教学建议

教学章节	教学要求	课　时
第 1 章 C++ 程序设计基础	掌握 C++ 编程风格 掌握 C++ 程序的词法单位 掌握 C++ 的基本数据类型 掌握变量与常量 掌握运算符和表达式 掌握语句 掌握输出和输入方法 了解枚举	6
第 2 章 C++ 的流程控制	了解算法的基本概念和表示方法 掌握选择结构程序设计、switch 语句 掌握循环结构程序设计	6
第 3 章 函　　数	掌握函数的定义和调用 掌握函数的声明 掌握函数的参数传递和返回值 掌握局部变量和全局变量 掌握变量的存储类别、默认参数、引用 掌握函数重载、函数模板、内联函数 掌握递归调用（根据授课对象灵活选择） 了解函数的调试方法 了解编译预处理	7 ~ 9
第 4 章 数　　组	掌握一维数组、多维数组 掌握数组做函数参数 掌握字符数组与字符串 掌握处理字符和字符串 了解标准 C++ 的 string 类	8
第 5 章 指　　针	掌握指针的概念和定义 掌握指针与数组 掌握指针与函数 掌握指针数组与指向指针的指针 了解内存的动态分配和释放 了解 void 和 const 修饰指针变量	8 ~ 9
第 6 章 结构体与链表	了解抽象数据类型 掌握结构体的定义及应用 了解用 typedef 定义类型 掌握单向链表（根据授课对象灵活选择）	4
第 7 章 文 件 操 作	掌握文件的基本概念、打开和关闭 掌握采用流操作符读写文件 掌握流对象做参数 掌握出错检测 掌握采用函数成员读写文件 了解多文件操作 掌握二进制文件 掌握随机访问文件	4

（续）

教学章节	教学要求	课　时
第 8 章 类的基础部分	掌握面向对象程序设计基本思想 掌握类的概念与定义方法 掌握多文件组织 掌握内联函数、构造函数、析构函数 掌握重载构造函数 掌握对象数组 了解抽象数组类型	6 ~ 8
第 9 章 类的高级部分	掌握静态成员 掌握友元函数 掌握对象赋值问题 掌握拷贝构造函数 掌握运算符重载 了解对象组合	7 ~ 8
第 10 章 继承、多态和虚函数	掌握继承 掌握保护成员和类的访问 掌握构造函数和析构函数 掌握覆盖、虚函数 掌握多重继承 了解多继承 了解类模板	6 ~ 8
第 11 章 异 常 处 理	掌握异常概念和定义 掌握基于对象的异常处理	2
第 12 章（选讲） 数据库程序设计	掌握采用 C++ 编写一个完整数据库管理系统	2 ~ 4

说明：教学内容分为核心知识模块（前 11 章）和提高模块（第 12 章），其中核心知识模块建议教学学时为 64 ~ 72，提高模块建议学时为 2 ~ 4。平时作业建议都在计算机上完成，上机实验不低于 60 机时，课程设计不低于 40 机时。不同的学校可以根据各自的教学要求和计划学时，对教学内容进行取舍。

目　录

第 1 章　C++ 程序设计基础

C++ 是在 C 的基础上扩充而成的，因其独特的机制在计算机领域有着广泛的应用。本章主要讲述 C++ 的基本知识，主要包括词法单位、数据类型、变量和常量、运算符和表达式、语句等基础知识，最后介绍简单的输入与输出方法。

1.1　为什么要学习 C++ 程序设计

随着计算机软硬件技术的发展，计算机应用规模不断提高，在软件开发语言和工具方面不断地推陈出新，新语言、新工具层出不穷。目前，国内许多高校，无论是计算机专业还是非计算机专业，都开设了 C++ 语言程序设计课程，并且将它作为一门专业必修课程。

C++ 是 C 的扩充版本。C++ 对 C 的扩充是由 Bjarne Stroustrup 于 1980 年在美国新泽西州玛瑞惠尔的贝尔实验室提出来的，起初，他把这种语言称为“带类的 C”，到 1983 年才改名为 C++。

在计算机刚发明时，人们采用打孔机直接进行机器指令程序设计，当程序长达几百条指令时，采用这种方法就很困难了。后来人们设计了用符号表示机器指令的汇编语言，从而能够处理更大更复杂的程序。到了 20 世纪 60 年代出现了结构化程序设计方法（C 语言就采用这种方法），这使得人们能够容易编写较为复杂的程序。但是，一旦程序设计达到一定的程度，即使结构化程序设计方法也变得无法控制，其复杂性超出了人的管理限度。例如，一旦 C 程序代码达到了 25 000 行至 100 000 行，系统就变得十分复杂，程序员很难控制，而设计 C++ 语言的目的就是为了解决这个问题，其本质就是让程序员理解和管理更大、更复杂的程序。因此，采用支持面向对象的 C++ 语言进行程序设计是时代发展的需要。

C++ 吸收了 C 和 Simula67（一个古老的计算机语言）的精髓，它具有 C 所无法比拟的优越性。C++ 在维持 C 原来特长（如效率高和程序灵活）的基础上，借鉴了 Simula67 的面向对象思想，将这两种程序设计语言的优点相结合。C++ 的程序结构清晰、易于扩展、易于维护同时又不失效率。目前，C++ 的应用已超出了当初设计其的目的，被成功地应用到数据库、数据通信等系统，并成功地构造了许多高性能的系统软件。C++ 与 C 相比，具有三个重要的特征，从而使其优越于 C。

第一个特征是支持抽象数据类型（Abstract Data Type，ADT），在 C++ 中 ADT 表现为类，是对对象的抽象，而对象是数据和操作该数据代码的封装体，它提供了对代码和数据的有效保护，可防止程序其他不相关的部分偶然或错误地使用对象的私有部分，这是 C 所无法实现的。

第二个特征是多态性，即一个接口，多重算法。C++ 既支持早期联编又支持滞后联编，而 C 仅支持前者。

最后一个特征是继承性。继承性一方面保证了代码复用，确保了软件的质量；另一方面也支持分类的概念，从而使对象成为一般情况下的具体实例。

这三个特性，我们将在后面的章节给予详细的讲解。

C++ 对 C 基本上完全兼容，很多用 C 写的应用程序都可以在 C++ 环境中使用，因此 C++ 不是一个纯粹的面向对象程序设计语言，它即支持面向对象的程序设计方法，又支持面向过程的程序设计方法。

目前许多系统软件，如操作系统，数据库管理系统（DBMS）等都采用 C++ 编写，所以

从事有关软件开发、自动控制和计算机应用的人员，不掌握 C++ 简直寸步难行。一句话：掌握 C++ 编程已成为许多专业学生的必然选择。

C++ 有很多版本，国内比较流行的是微软公司推出的 Visual C++，本教材采用的是 Microsoft Visual Studio 2010，简称 VS 2010。

1.2 简单的 C++ 程序举例

【例 1-1】 下面通过一个简单程序来分析 C++ 程序的基本构成和特点。为了便于解释程序，我们给程序加了行号，这在写程序时是不需要的。

```
1   #include  <iostream>
2   using  namespace  std;
3
4   int  main( )
5   {
6       int a, b;                  //定义两个变量
7
8       cout << "输入变量 a 和 b: " ;
9       cin >> a >> b;             /* 从键盘输入 a 和 b 的值 */
10      cout << "a + b = " << a + b << endl;
11
12      system("pause");
13      return 0;
14  }
```

将该程序以扩展名为 .cpp 的文件形式保存，经过编译、链接生成可执行文件，运行后显示如下信息：

```
输入变量 a 和 b: 22   88 [Enter]
a + b = 110
```

用户输入 22 和 88 并按回车键后，输出它们相加的和是 110。

上述 C++ 程序由编译预处理指令、程序主体和注释组成。

程序的第 1 行是编译预处理指令，include 称为文件包含，它使后面的 iostream 成为本程序的一部分，这样本程序可以直接使用包含文件中的函数。编译预处理是 C++ 提供的组织程序的一种工具，我们将在 3.13 节介绍。

第 2 行是用 using 告诉编译器名字空间的名字，上例中的 std 代表系统提供的标准名字空间，当采用“using namespace std;”说明以后，程序就可以访问 std 名字空间中的内容。

第 4 ~ 14 行是程序的主体，其中第 4 行是函数头，第 5 ~ 14 行称为函数体。一个 C++ 程序由一个或多个函数构成，并且在这些函数中，有且仅有一个主函数 main，它是程序执行的入口。任何一个函数由“{}”中的语句序列描述，如本例中的第 5 行和第 14 行，而它们括起来的第 6 ~ 13 行是程序执行的语句，并且任何一个语句都以“;”结束。C++ 程序严格区别字母的大小写。

注意： 程序第 12 行调用系统的暂停功能，当程序执行到此行时，在命令行窗口中输出“请按任意键继续 ...”的提示，并暂停程序，等待用户按一个键，然后继续执行。如果没有此行，在 VS 2010 环境下，程序将一晃而过。如果你采用的是低版本的开发环境，则不需要此行，一个程序运行结束后，将自动暂停。在此后的程序中，我们省略了该行，请读者根据开发环境来取舍。

1.3　注释方法

C++ 的注释形式有两种。一种是“ /**/”格式，这是 C 语言中的注释风格。因为 C++ 是 C 的扩充版，所以它支持 C 的注释方法。在 C 语言中，注释是以“/*”开头，以“*/”结束。编译器将忽略这两个符号之间的所有语句，如上例中的第 9 行采用的就是这种注释风格。另一种注释格式是双斜线（//），在双斜线之后的部分都会被视为注释，如程序中的第 6 行。注释是程序员用来说明程序或解释代码的语句。注释是程序的组成部分，但编译器在编译时忽略它，不构成可执行代码。它也属于编程风格中关键的一环。

许多程序员都会在源程序中尽可能少地添加注释语句，因为编写源代码本身已经很痛苦了。但是，养成加注释的习惯是很有帮助的。虽然程序员在编程时可能要花费额外的时间，但在以后将会节省很多时间。假设你辛苦数月，编写了一个 8 000 ~ 60 000 行代码的 C++ 程序，你完成了编码，并且调试成功，交给了客户使用，于是你继续做下一个项目。一年以后你需要对程序进行修正，当你打开数万行没有注解的源代码时，你发现有许多函数已经不知道它们的功能，当初设计的目的是什么都不知道。这时你就会想，要是当初加一些必要的注释就好了，但为时已晚，此时要做的只能是用较多的时间来理解源程序，或重写代码。

注意：不必为程序的每一行都加注释，也不必为一目了然的代码加注释，只要注解适当的代码，有助于他人理解即可。

C 注释方法能跨越多行，便于对多行注释。但对单行注释并不方便，程序员往往把两者结合起来使用：使用 C 注释方法完成多行注释，使用 C++ 注释方法完成单行注释。

1.4　编程风格

程序员使用标识符、空格、Tab 键、空行、标点符号、代码缩进排列和注释等来安排源代码的方式，就构成了编程风格中重要的组成部分。

当编译器对源程序进行编译时，它会将程序处理为一个长字符串。一条语句不在同一行或者操作符和操作数之间有空格，都不会影响编译。但阅读程序的人很难读懂这种书写不规范的程序。例 1-2 虽然没有什么语法错误，但却很难读。

【例 1-2】 一个令人难以理解的程序。

```
#include  <iostream>
using namespace std;
int  main( ){int a, b; cout << "输入变量 a 和 b: " ;       cin
>> a >> b; cout << "a + b = " << a + b << endl; return 0; }
```

上面的程序虽然没有违反 C++ 的语法规则，但却难以阅读。较为理想的编程风格应在编程时使用空格、空行和代码缩进排列，从而使别人能很快读懂你的程序。

警告：尽管编程风格的自由度很大，但还是应该遵循程序设计的国际常规，这样其他程序员才会很容易读懂你的程序。建议你模仿本书的编程风格。当然，你也可以参考国际知名公司（如微软、IBM 等）的风格。不同的公司编程风格不尽相同，可以参考其提供的样式文件，这对培养编程风格大有裨益。

1.5　C++ 程序的词法单位

本节介绍 C++ 程序使用的字符、关键字和标识符。

1.5.1 C++ 程序中的字符

标准的 C++ 程序采用 0x00 到 0x7F 范围内定义的 ASCII 码所表示的西文字符作为程序的基本字符单位，主要包括 26 个小写英文字母、26 个大写英文字母、10 个阿拉伯数字和其他一些符号，如 +、–、*、/ 等，其中每个 ASCII 码字符占用一个字节。

1.5.2 标识符

标识符是程序员自己定义的“单词”，标准 C++ 的标识符由字母、下划线和数字组成，且第一个字符不能为数字，长度一般不超过 32 个，文件名只识别前 8 个字符。标识符区分大小写，同一个单词的不同大小写被编译器看作不同的标识符。在实际使用时应尽量采用有意义的单词作为标识符，达到见名知义。

虽然 C++ 编译器允许用户定义的标识符以下划线开始，但因为系统定义的内部符号以下划线或双下划线开始，所以自定义的标识符不提倡以下划线开始。

以下均是合法的并且达到见名知义的标识符：

```
studentName、StudentName、name_of_student、salary
```

通过变量名 studentName，可以很容易地看出该变量的意义。这种编程风格使得程序易于理解和维护，因为一个大的系统通常由数万行源代码构成，程序的可读性十分重要。定义标识符应尽量选择有含义的英文单词，以下变量命名就毫无意义：

```
abc、xyzw、a123、x888888
```

标识符是区分大小写的，变量 studentName 和 StudentName 是不同的变量。我们之所以将变量 studentName 中的“N”大写，是为了增强程序的可读性，如 studentname 是全部由小写字母构成的变量名，使人不易读懂，因此这不是最好的变量名。

总之，选择标识符时，要尽可能做到“见名知义”，选择有含义的单词符号作为标识符，使别人（包括你本人）容易读懂你的程序。

以下是非法的标识符：

```
8abc                    //标识符不能以数字开头
Student  Name           //标识符中间不能有空格
$bill                   //标识符不能以 $ 开头
```

1.5.3 关键字

关键字又称保留字，是系统定义的一些特殊标识符，它们具有特定含义，不允许程序员将它们挪作他用，如作为一般标识符使用。C++ 中常用的部分关键字如表 1-1 所示：

表 1-1 C++ 常用关键字

类 型	关 键 字
数据类型说明符和修饰符	bool、char、class、const、double、enum、float、int、long、short、signed、struct、union、unsigned、void、volatile
存储类型说明符	auto、extern、inline、register、static
访问说明符	friend、private、protected、public
语句	break、case、catch、continue、default、do、else、finally、for、goto、if、return、switch、throw、try、while
运算符和常量	delete、false、new、sizeof、true
其他	asm、explicit、namespace、operator、template、this、typedef、typename、using、virtual

还有一些不常用的关键字：bad_cast、bad_typeid、const_cast、dynamic_cast、except、mutable、reinterpret_cast、static_cast、type_info、typeid 和 wchar_t。我们会在后面的章节中逐步引入并介绍常用关键字的含义，其他内容读者可参考有关手册。

1.6　C++ 的基本数据类型

从程序设计语言原理的角度讲，C++ 是一种强类型的语言，必须严格遵循“先定义后使用”的原则。读者可以在后续的学习中慢慢品味该原则。

C++ 的数据类型分为两大类：基本数据类型和导出数据类型。

基本数据类型也称为 C++ 预定义的类型或内置数据类型，包括字符型（char）、整型（int）、单精度实型（float）、双精度实型（double）、布尔型（bool）和空类型（void）。这些预定义的类型，不仅定义了数据类型，还定义了常用的操作。

导出数据类型是由基本数据类型构造出来的数据类型，包括数组、指针、引用、结构体、共用体、枚举和类等。

字符型用来存放一个字符的 ASCII 码值，可以将其看成一个 8 位二进制码的整数，如大写字母 A 的 ASCII 码是 65。宽字符类型（wchar_t）也称为双字节字符，往往采用 Unicode 编码格式，该标准中的所有字符都是双字节的，这样可以统一处理西文、中文、阿拉伯文和其他语言的符号。但宽字符类型不属于基本类型，限于篇幅，本书不做介绍。

整型用来存放一个整数，一般占用 4 个字节，无符号数采用原码的形式表示，而有符号数采用补码表示。

实型用来存放实型数据，C++ 提供了 float 和 double 两种实型类型，因占用的字节数不同，其表示的数据范围也不同。

逻辑型也称布尔型。为了纪念英国的数学家乔治·布尔（George Boolean），在程序设计语言中引入了“布尔”类型。布尔型用来处理逻辑量，取值只有 true（真）和 false（假）两个，占 1 个字节，我们将非 0 值解释为 true，将 0 值解释为 false。

注意：从程序设计语言原理的角度讲，布尔变量的取值只能是真或假，但 C++ 中对它的定义很不严格，将 0 看做 false，将非 0 看做 true。这是为了向下兼容，即兼容它的子集 C 语言，因为 C 就是这样定义的。

空类型（void）用来定义指针，或用来说明函数的返回值类型，将在 5.7.1 节讲解。

C++ 允许在整数、字符或浮点类型前面加上修饰符：short（短类型）、long（长类型）、signed（有符号类型）和 unsigned（无符号类型）。表 1-2 列出了 C++ 中基本数据类型及其变量的取值范围。

表 1-2　C++ 中所有的基本数据类型

类　型	名　称	占用字节数	取值范围
bool	布尔型	1	true，false
[signed] char	有符号字符型	1	-128 ~ 127
unsigned char	无符号字符型	1	0 ~ 255
[signed] short[int]	有符号短整型	2	-32768 ~ 32767
unsigned short[int]	无符号短整型	2	0 ~ 65535
[signed]int 或 signed	有符号整型	4	-2^{31} ~ ($2^{31}-1$)

（续）

类　型	名　称	占用字节数	取值范围
unsigned[int]	无符号整型	4	0 ~ (2^{32}-1)
[signed]long[int]	有符号长整型	4	-2^{31} ~ (2^{31}-1)
unsigned long[int]	无符号长整型	4	0 ~ (2^{32}-1)
float	实型	4	-10^{38} ~ 10^{38}
double	双精度实型	8	-10^{308} ~ 10^{308}
long double	长双精度实型	8	-10^{308} ~ 10^{308}

注意 1：可选项问题。例如 [signed]char 中的“[signed]”代表可选项，即它等价于：signed char 或 char。

注意 2：有些教材将 void 类型作为基本数据类型，我们不赞成这种观点。因为 void 在 C++ 中只能用来修饰指针和函数，后面的章节会讲解它的用途。

1.7 变量与常量

1.7.1 变量

C++ 作为一种强类型的程序设计语言，在使用变量前应当先说明类型，然后再定义变量，以便于编译器分配内存空间，以及对程序中的数据类型和运算等进行常见错误检查，以提高程序的编译和运行效率。

在程序运行中，值可变的量称为变量。变量必须用标识符（即变量的名字）来标识。变量根据其取值范围的不同可分为字符型、整型、实型变量等。在程序运行时，系统将会给每个变量分配一段连续的内存单元，用来存放变量的值。

变量有三个要素：变量名、变量的内存空间和变量的值。

1. 定义变量

定义变量的一般格式为：

```
[<存储类别>] <数据类型> <变量名 1>[, <变量名 2>, …<变量名 n >];
```

其中，“<>”括起来的是必选项。存储类别将在 3.5 节介绍。下面定义几个变量：

```
bool  b;                          //定义 1 个布尔型变量 b
char gender, ch;                  //定义 2 个字符型变量 gender 和 ch
int a, b ;                        //定义 2 个整型变量 a 和 b
double dx ;                       //定义 1 个双精度实型变量 dx
float f ;                         //定义 1 个单精度实型变量 f
unsigned u ;                      //定义 1 个无符号整型变量 u
```

变量必须先定义后使用，原因如下：

1）变量定义后就具有了变量名和类型。编译系统根据类型给变量分配内存空间，并建立变量名和其内存空间的对应关系，于是可以通过变量名给变量的内存空间赋值或读取该内存空间中的值。

2）变量确定类型后，编译器可以对变量参与的运算做合法性检查。

2. 变量赋值

当使用变量时，变量必须有一个确定的值，给变量赋值的方法有以下两种：