教育部大学计算机课程改革项目规划教材

C/C++程序设计案例教程
—— 基于计算思维

辛士光　贾丽娟　王　乾　主编

张殿龙　审

高等教育出版社

教育部大学计算机课程改革项目规划教材

C/C ++ 程序设计案例教程
——基于计算思维

C/C ++ Chengxu Sheji Anli Jiaocheng
——Jiyu Jisuan Siwei

辛士光　贾丽娟　王　乾　主编
张殿龙　审

高等教育出版社·北京

内容提要

本书以培养学生的计算思维能力为目的，注重工程实践和编程能力训练，通过对实际案例的剖析、算法分析和程序代码的编写，结合C/C++语言的工程应用引出相关知识点，让学生在实例中了解本章重点阐述的内容。通过对小的案例进行分析、编程，让学生掌握C/C++的相关语法和语句。再通过综合案例让学生掌握如何分析问题和使用C/C++解决问题。这样可以有效地帮助学生理解、掌握面向对象方法的核心思想，从而逐步培养学生面向对象的思维方式。

本书共15章，从C/C++程序设计语言的数据类型、运算符、表达式、三种控制结构、数组、字符串指针、函数和结构体类型开始介绍，进而深入到面向对象程序设计的类与对象、继承与多态、C++输入/输出流类库和图形界面编程的讲解。

本书配有多媒体课件、微课件、例题和习题源代码等教学资源，可从高等教育出版社网站下载，网址为http://computer.cncourse.com。

本书可作为高等学校各专业C/C++程序设计课程的教材，也可作为从事计算机相关工作的各类人员的参考用书。

图书在版编目（CIP）数据

C/C++程序设计案例教程：基于计算思维/辛士光，贾丽娟，王乾主编．--北京：高等教育出版社，2014.3

ISBN 978-7-04-033192-9

Ⅰ.①C… Ⅱ.①辛… ②贾… ③王… Ⅲ.①C语言-程序设计-高等学校-教材 Ⅳ.①TP312

中国版本图书馆CIP数据核字(2014)第023718号

策划编辑 唐德凯	责任编辑 唐德凯	封面设计 于文燕	版式设计 杜微言
插图绘制 尹 莉	责任校对 刘春萍	责任印制 张福涛	

出版发行	高等教育出版社	咨询电话	400-810-0598
社 址	北京市西城区德外大街4号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	北京奥鑫印刷厂	网上订购	http://www.landraco.com
开 本	787mm×1092mm 1/16		http://www.landraco.com.cn
印 张	24.75	版 次	2014年3月第1版
字 数	610千字	印 次	2014年3月第1次印刷
购书热线	010-58581118	定 价	38.80元

物 料 号 33192-00

前　言

C/C ++语言既具有高级语言好学易懂的特点，又可以像低级语言那样直接访问内存。因此，学习C/C ++语言不仅可以从硬件上了解计算机的工作过程以及内存对数据的组织形式，还可以通过软件编程控制计算机解决实际问题。

从高级语言的发展过程可以看出，C/C ++语言的生命力是最长的，之后的大多数编程语言（如Java、C#等）都是基于C语言的语法。这是因为C/C ++语言具有丰富的数据类型和运算符，同时还可以自定义数据类型，解决问题非常方便；另外，C/C ++语言的可移植性好，代码质量高，可以用来编写操作系统等系统程序；在工程控制上，C语言也是很好的嵌入式编程语言，在应用程序的开发上，C ++语言已经成为主要的开发工具。

本书具有如下主要特点：

1. 基于计算思维

本书以培养学生的计算思维能力为目的，以实践性综合案例为基础，从程序设计的算法分析中训练、培养学生分析问题、解决问题的逻辑思维能力和编程思想。本书精心设计知识、能力、素质培养“三位一体”的程序设计教学模式，通过综合案例剖析、算法分析和程序代码的编写，结合C/C ++语言的工程应用，不断引出新的知识点，再通过若干简单案例的讲解，不仅让学生掌握C/C ++语言的语法和语句，更重要的是培养学生面对实际问题建立模型和描述算法的能力，通过约简、嵌入、转化和仿真等方法对复杂事物进行抽象、分解的能力，将复杂问题归纳推导至他们所熟悉的简单问题上的能力，为今后在交叉学科（如自然科学、社会科学）等二元或多元学科解决问题打下基础。

2. 改进内容组织结构

在内容结构上，本书各章从能够涵盖本章知识点的案例剖析入手，通过对案例的分析，引出具体的算法，根据算法编写程序，再对程序进行调试运行，最后对结果进行分析。通过综合案例的剖析，使学生对本章所涉及的知识点有总体上的认知。随后对本章主要知识点进行具体讲解，并配有若干例题。在各章节的最后，设计了贯穿整个教材的综合案例，使学生了解一个问题从简单功能再到复杂功能的实现，引导学生体会由浅入深、循序渐进地扩展及优化程序功能的编写过程。

3. 从结构化程序设计思想过渡到面向对象程序设计思想

对于没有学习过C/C ++语言的人员来说，不适合直接学习面向对象的编程方法。由于结构化程序设计思想是面向对象程序设计思想的基础，所以本书以结构化程序设计思想的C语言的知识为铺垫，过渡到面向对象的C ++语言编程。

4. 从C语言到C ++语言

C ++语言是从C语言发展演变而来的一种面向对象的程序设计语言，C ++语言兼容C语言。本书在介绍C ++语言语法知识的同时，也介绍了C语言的使用规则。这样的组织，既可以使学生在今后的工作中能够用C语言进行底层的嵌入式应用程序或驱动程序的编程，也可

以用 C ++语言进行大型应用软件系统的开发。

5. 风格统一

本书每章最后的综合案例都是基于一个问题的求解，由浅入深。书中的程序代码采用统一的编写规范，标识符定义、代码格式力求规范和统一，在实现功能的基础上，注重程序的健壮性和易读性。

6. 配有实验与实训教材

本书配有实验与实训教材《C/C ++程序设计案例教程实验与实训指导》，主要包括上机实验、综合实践项目两部分。实验内容紧密配合本书的章节知识点，难度分配合理；综合实践内容注重体现项目的工程性、创新性，以提高学生的实践能力和计算思维能力。实验与实训教材的附录中，配有主教材的习题答案。

本书由辛士光、贾丽娟、王乾主编，张殿龙主审。第 1 ~ 5 章由辛士光编写，第 6 ~ 10 章由王乾编写，第 11 ~ 15 章由贾丽娟编写。赵一峰老师参与了本教材的校对工作，在此表示衷心的感谢。

因编者水平有限，书中的错误与疏漏之处在所难免，恳请广大读者批评指正。

编 者

2013 年 11 月

目　录

第1章　初识C/C++程序设计语言

本章导读

- C语言和C++语言（常简称为C++）的发展过程。
- C++程序的基本框架。
- 如何使用程序设计进行计算问题的求解。
- 面向过程和面向对象编程。
- C++是如何在C语言的基础上添加面向对象概念的。
- 程序设计有哪些基本要素。
- C语言和C++的开发环境。

1.1　程序设计语言概述

程序设计语言（Programming Language）是一组用来定义计算机程序的、被标准化的语法规则，用来向计算机发出指令。一种程序设计语言让程序员能够准确地定义计算机所需要使用的数据，并精确地定义在不同情况下所应当采取的行动。语言的基础是一组记号和一组规则。程序设计语言包含3个方面：语法、语义和语用。语法表示程序的结构或形式，即表示构成语言的各个记号之间的组合规律，但不涉及这些记号的特定含义，也不涉及使用者；语义表示程序的含义，即表示按照各种方法所表示的各个记号的特定含义，但不涉及使用者；语用表示程序与使用者的关系。

按照不同的分类方法，程序设计语言可分为不同的类别。

（1）低级语言和高级语言。低级语言包括机器语言和汇编语言。低级语言与特定的机器有关，功效高，但使用复杂、费时、易出差错。机器语言中每条指令都用0和1组成的二进制序列来表示。汇编语言是机器语言符号化的结果，采用助记符来表示机器语言的指令，助记符一般使用代表某种含义的英文字母的缩写，与机器语言相比，更容易记忆。高级语言的表示方法要比低级语言更接近于待解问题的表示方法，其特点是在一定程度上与具体机器无关，易学、易用、易维护。

（2）过程式语言和非过程式语言。过程式语言的主要特征是，用户必须指明一系列可执行的运算，以表示相应的计算过程，如FORTRAN、COBOL、Pascal等。非过程式语言不需要知道问题是如何求解的，只需要描述需要求解的问题是什么，然后便可用程序设计语言来实现，如SQL。

（3）通用语言与专用语言。如Java、C#、FORTRAN、Pascal、C等都是通用语言，应用领域单一的语言称为专用语言，如APT等。

（4）面向过程语言与面向对象语言。如C语言是面向过程的语言，Java、C++、C#等为

面向对象的语言。

程序设计语言有许多种，为什么要学习C/C++语言？C/C++语言到底是什么样的一种程序设计语言？了解了C/C++语言的诞生、发展以及其地位，就会知道学习它的重要性。

1.2　C/C++程序设计语言概述

20世纪70年代早期，美国贝尔实验室的Dennis Ritchie致力于UNIX操作系统的开发。为了完成这项工作，Ritchie希望有一种语言能将低级语言的效率、硬件访问能力和高级语言的通用性、可移植性融合在一起，于是他在原B语言的基础上开发了C语言。

与C语言一样，C++也是在贝尔实验室诞生的。1983年，贝尔实验室的Bjarne Strou-strup在C语言的基础上推出了C++。C++进一步扩展和完善了C语言，成为一种面向对象的程序设计语言。C++提出了一些更为深入的概念，这些面向对象的概念更容易将问题空间直接地映射到程序空间，为程序员提供一种与传统结构程序设计不同的思维方式和编程方法。但同时，这也增加了整个语言的复杂性，使得掌握起来有一定难度。

为了实现不同系统的程序移植（例如希望编写的C++程序能够在Windows XP、Red Hat Linux和Macintosh OS X上运行），就需要编写的程序遵循某种公开的标准。美国国家标准协会（American National Standards Institute，ANSI）在1990年成立了一个委员会，专门制定C++标准（也制定了C语言标准）。国际标准化组织（ISO）与ANSI创建了联合组织ANSI/ISO，致力于制定C++标准。1998年，ISO、IEC（International Electrotechnical Committee，国际电工技术委员会）和ANSI批准了第一个标准ISO/IEC 14882:1998，称为C++ 98标准。该标准不仅描述了已有的C++特性，还对该语言进行了扩展，添加了异常、运行阶段类型识别（RTTI）、模板和标准模板库（STL）。2003年发布了C++标准第二版ISO/IEC 14882:2003，称为C++ 03。该版本并没有改变语言特性，只是对第一版修订了错误，减少了多义性，因此常使用C++ 98表示C++ 98/C++ 2003。

用C/C++编写的程序运行速度快，并且具有很好的可移植性。编写程序最好使用集编辑、编译、调试和运行等功能于一体的集成开发环境（IDE），以提高编程效率。目前，Windows下比较流行的C/C++集成开发环境有Borland C++ 5.5、Symantec C++ 7.0、Microsoft Visual Studio 2012、Dev-C++ 5.0等，UNIX/Linux下比较流行的集成开发环境是GCC 4.7.0。此外，还有跨平台的开源集成开发环境Code::Blocks 10.05。

C++融合了3种不同的编程方式：C语言代表的面向过程编程；C++在C语言的基础上增加的类所代表的面向对象编程；C++模板支持的泛型编程。使用C++的原因之一就是利用其面向对象的特性。要利用好这些特性，必须对标准C的基本类型、运算符、控制结构和语法规则有较深入的了解。

C/C++一直以来被专业公司和编程者广泛关注，TIOBE世界编程语言2014年1月排行榜中，C语言和C++分别排名第一和第四，如表1-1所示。这也说明了C/C++语言很高的热度。

表 1-1 2014 年 1 月编程语言前 15 名排行

2014 年 1 月排名	2013 年 1 月排名	排名变化	程序设计语言
1	1	=	C
2	2	=	Java
3	3	=	Objective - C
4	4	=	C++
5	5	=	C#
6	6	=	PHP
7	7	=	(Visual) Basic
8	8	=	Python
9	10	↑	JavaScript
10	22	↑↑↑↑↑↑	Transact - SQL
11	12	↑	Visual Basic. NET
12	11	↓	Ruby
13	9	↓↓↓↓	Perl
14	14	=	Pascal
15	17	↑↑	MATLAB

1.2.1 程序实例

```
// ch01_1. cpp
#include < iostream >                    //编译预处理
using namespace std;                     //使用名字空间 std
int main( )                              //主函数,程序入口
{                                        //函数体开始
    cout ≪ "hello world!";               //屏幕显示 hello world!
    cout ≪ endl;                         //开始一新行
    return 0;                            //函数结束运行
}                                        //函数体结束
```

程序运行结果如图 1-1 所示。

```
hello world!
Press any key to continue
```

图 1-1 程序运行结果

C++的输入和输出函数为 cin 和 cout，同时 C++也能够使用 C 语言的 scanf( )、printf( )和其他所有标准的 C 输入和输出函数（需要包含 C 语言的 stdio. h 文件）。

C++这个词在中国大陆的程序员圈子中通常被读做“C 加加”，而西方的程序员通常读做“C plus plus”或“CPP”，这也是 C++源程序的扩展名为什么是 . cpp 的原因。

C 语言也常简称为 C，C++语言常简称为 C++，本书中不做区分。

1.2.2　C++程序的基本框架

通常，使用C++编程要先将程序组织成一个一个的任务，然后设计独立的函数来处理这些任务。程序实例ch01_1.cpp的源程序中，只有一个名为main()的函数，其所包含的元素如下：

- 注释：以前缀“//”标识。
- 编译预处理指令：#include。
- 指定名字空间：using namespace。
- 函数头：int main()。
- 函数体：用一对花括号“{}”括起来的代码。
- 输出语句：使用cout显示信息的语句。
- 结束语句：函数体中的return语句。

ch01_1.cpp示例程序去掉附加代码后的基本结构如下（见图1-2）：

```
int main()
{
    语句;
    return 0;
}
```

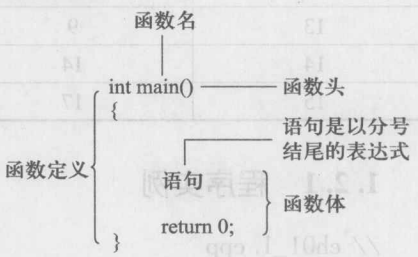

图1-2　main()函数的基本结构

main()函数：C/C++源程序必须包含一个名为main()的函数，程序运行是从main()函数开始执行，在main()函数中结束。

C++预处理和iostream文件：#include是编译预处理命令，通过“包含”实现把另一文件的内容插入到当前#include语句的位置。C++的输入对象和输出对象定义在iostream类文件中，因此要在程序中进行输入和输出，需要有如下两行代码：

```
#include <iostream>
using namespace std;
```

语句和分号：语句是要执行的操作，为了编译代码，编译器需要知道一条语句何时结束，分号即为语句结束符（也可以理解为两条语句之间的分隔符）。

注释：C++注释以双斜杠（//）开头。注释是程序员为读者提供的说明，通常用于标注程序的一部分或解释代码的某个方面，编译器会自动忽略注释。C++也能够识别C注释，C注释包括在“/*”和“*/”之间，可以实现跨越多行的注释。

1.2.3　C/C++编程流程

用程序设计语言编写的源程序，通过编译器排错后翻译成目标程序（二进制代码），并经过链接才能运行程序获得结果。翻译的方式有两种：一种是解释型，如Java语言，计算机边读程序边翻译成机器代码，然后立即执行；另一种是编译型，是先将全部源程序翻译成机器代码，保存在可执行程序文件中，然后启动该程序，运行得到结果。C和C++都是编译型的程序设计语言。

一般的编程流程为：编辑（Edit）→编译（Complie）→连接（Link 或 Make 或 Build）→调试（Debug）。该过程循环往复，直至完成，如图 1-3 所示。

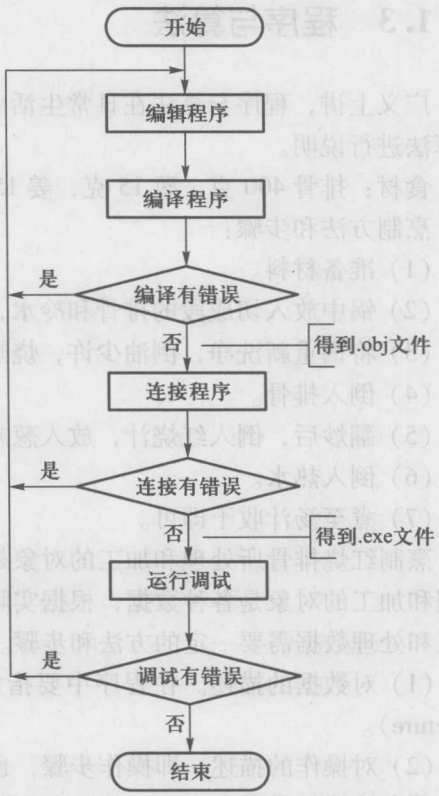

图 1-3 C/C++编程流程

程序员编写的程序称为源程序或源代码（Source Code），通过编辑工具输入计算机。C++源程序以文本文件的形式保存在扩展名为 .cpp 的文件中（用 C 语言编写的源程序文件的扩展名为 .c）。程序被编译后，会生成目标代码（Object Code），存放在扩展名为 .obj 的目标文件中。目标文件是计算机能够识别的指令集合，但是目标代码还不能在具体的计算机上运行，因为目标代码只是一个一个独立的程序段，程序段中用到的 C++ 库代码和其他资源还没有挂接上，需要相互衔接成适应一定操作系统的可执行程序整体。通过连接可以把这些程序段整合在一起转换为扩展名为 .exe 的可执行程序。

1.2.4 C/C++语言的特点

1. C 语言的特点

（1）C 语言具有结构化语言的特点，并具有结构化的程序控制语句。

（2）C 语言程序的主要构成单位为函数。函数可以在程序中被定义，以完成独立的任务，并可被独立地编译成代码，以实现程序的模块化。

（3）C 语言运算符丰富，灵活地使用运算符可以实现在其他高级语言中难以实现的运算。

（4）C 语言数据类型丰富，能够表示各种复杂的数据结构。

（5）C 语言允许直接访问物理地址，可以直接对硬件进行操作，实现汇编语言的大部分功能，这使得 C 语言成为编制系统软件的基本语言。

（6）C 语言语法限制不严格，程序设计自由度大。

（7）C 语言程序生成的目标代码质量高，程序执行效率高，可移植性好。

2. C++的特点

C++的主要特点表现在以下几个方面：

（1）C++是一个更好的 C，保持了 C 的简洁、高效和接近汇编语言的特点，对 C 的类型系统进行了改革和扩充。因此，C++比 C 更安全，C++的编译系统能检查出更多的类型错误。

（2）C++兼容 C，使得许多 C 代码不经修改就可为 C++所用。

（3）C++既支持面向过程的程序设计，也支持面向对象的程序设计。

1.3　程序与算法

广义上讲，程序与算法在日常生活中无处不在。下面以红烧排骨的烹制过程为例，对程序与算法进行说明。

食材：排骨400克，葱15克，姜15克，李锦记秘制红烧汁60克，热水400克。

烹制方法和步骤：

（1）准备材料。

（2）锅中放入切成段的排骨和冷水，先用大火烧开去掉浮沫。

（3）将锅重新洗净，倒油少许，烧热。

（4）倒入排骨。

（5）翻炒后，倒入红烧汁，放入葱末、姜丝。

（6）倒入热水。

（7）煮至汤汁收干即可。

烹制红烧排骨所处理和加工的对象是各种食材，需要采用一定的烹制方法和步骤。程序所处理和加工的对象是各种数据，根据实际需要，这些数据需要采用不同的存储和组织方式，而加工和处理数据需要一定的方法和步骤。因此，一个程序应该包括以下两方面的内容：

（1）对数据的描述。在程序中要指定数据的类型和数据的组织形式，即数据结构（Data Structure）。

（2）对操作的描述。即操作步骤，也就是算法（Algorithm）。

著名计算机科学家N. Wirth提出这样一个经典公式：程序＝数据结构＋算法。实际上，一个程序除了以上两个主要的要素外，还应当采用某种程序设计方法进行设计，并且用一种计算机语言来表示。因此，算法、数据结构、程序设计方法和语言工具4个方面是一个程序员所应具备的知识。

1.4　面向过程程序设计

从程序设计方法的角度来看，程序设计就是把要完成的任务组织成一定形式的操作序列。同一任务采用不同的程序设计方法可以组织成不同的操作序列。

程序中所处理的数据确定之后，需要考虑的就是算法了，也就是要解决如何设计算法的问题。当算法规模较大时，可以考虑将它按功能拆分，形成一个个小规模的任务，将这些小规模任务依次实现，并通过某种调用关系把这些任务整合在一起，形成完整的程序。面向过程编程的示意图如图1-4所示。

例1.1　学生成绩的数据保存在score.txt文件中，如果成绩及格，则收集起来，然后把收集到的及格成绩从大到小排序并显示出来。

可以把求解过程简单地划分为3部分：输入（Input）、处理（Processing）、输出（Output），简称IPO。写成算法就是：

第一层（总体结构）