

C / C ++ 程序设计

计算思维的运用与训练

主 编 齐苏敏

副主编 公冶小燕 姜海涛 王 抒

齐邦强 叶传秀



北京理工大学出版社

计算机基础教育“十三五”系列规划教材·程序设计类


C/C++ 程序设计：

计算思维的运用与训练

主 编 齐苏敏

副主编 公冶小燕 姜海涛 王 抒

齐邦强 叶传秀

 北京理工大学出版社

BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 提 要

本书以计算思维的运用与训练为目标,以程序设计实训为手段,介绍运用 C/C++ 语言分析问题和解决问题的方法与技巧。本书内容分为 C 语言篇与 C++ 语言篇,由简入难,有助于读者实现从面向过程程序设计到面向对象程序设计的顺利过渡。

C 语言篇:按 C 语言的知识点设计面向过程程序设计与模块化设计的学习与训练,从结构化程序的基本要素、逻辑思维、重复思维、过程封装、数据封装、批量数据处理、数据的间接访问等方面解析结构化程序设计的过程,并在实训中介绍其思维方法与编程技巧。C++ 语言篇:按 C++ 语言的知识点设计面向对象程序设计与训练,从 C++ 语言的面向对象思维、数据与过程的封装、对象运算、代码重用等方面分析面向对象程序的组成,并在实训中介绍其思维方法及软件开发过程。

本书可作为高等院校本专科“C/C++ 程序设计”课程的教学用书,也可以作为 C/C++ 程序设计的自学用书。

版权专有 侵权必究

图书在版编目 (CIP) 数据

C/C++ 程序设计:计算思维的运用与训练/齐苏敏主编. —北京:北京理工大学出版社, 2019.8 (2019.9 重印)

ISBN 978 - 7 - 5682 - 7389 - 3

I. ①C… II. ①齐… III. ①C 语言 - 程序设计 - 高等学校 - 教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 176433 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 /

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 19.25

字 数 / 450 千字

版 次 / 2019 年 8 月第 1 版 2019 年 9 月第 2 次印刷

定 价 / 48.00 元

责任编辑 / 梁铜华

文案编辑 / 曾 仙

责任校对 / 周瑞红

责任印制 / 李志强

图书出现印装质量问题,请拨打售后服务热线,本社负责调换

前 言

程序设计是信息科学与技术等专业的一门重要课程，是一门具有较强实践性的课程。然而，目前程序设计的教学模式以及教材设计内容以课堂教学为主，教学内容侧重于计算机语言的语法，在教授学生如何运用计算思维分析问题和解决问题方面较弱，难以激发学生的学习积极性。

本书根据工程认证的标准，以计算思维的运用与训练为目标，以程序设计实训为手段，着重培养学生运用 C/C++ 语言分析问题和解决问题的能力。C++ 语言是对 C 语言的扩充与完善，是程序设计思维方式的提升。对 C++ 语言的学习不必重复 C 语言的内容，而应重点学习与理解面向对象的思想，学习如何运用面向对象思维来分析问题、解决问题。

本书基于教学团队的实践教学平台而设计，所对应的课程“C/C++ 程序设计”全面采用实验教学为主的教学模式，通过电子教室系统，在实验机房的编译环境讲授 C/C++ 程序的编辑、调试、运行，实现 C/C++ 程序设计知识点的讲授与训练，并通过教学平台布置在线作业、监督在线作业的提交状况、查看在线成绩、定期设置在线考试等。不同的专业、不同的学习群体可以根据实际情况来调整学习计划与授课方式。

本书的主要特点如下：

1. 注重思维，强调思想

本书共分 12 章，其中 C 语言篇为第 1~8 章，C++ 语言篇为第 9~12 章。C 语言篇着重介绍面向过程程序设计的逻辑思维、重复思维、数据封装方式、过程封装方式、数据间接访问方式，以及数据存储方法。C++ 语言篇着重介绍 C++ 语言对 C 语言的扩充内容、面向对象的程序设计思想，解析类的封装性、继承性与多态性，以便学生在实训中掌握运用面向对象思维来分析问题、解决问题的方法。

2. 精选示例，强调实训

本书以示例引出知识点，然后讲解如何运用相关知识点，最后以实训实例来讲解怎样利用计算思维来分析问题和解决问题。每章都提供了丰富的实训实例分析与解决问题的过程，并在每个实训案例后附有相关实训练习。

3. 由简入难，内容紧凑

本书由简入难，有助于学生熟练掌握 C 语言，并逐步转向对 C++ 语言的学习，实现从面向过程编程到面向对象编程的顺利过渡。C 语言和 C++ 语言的连续学习，旨在帮助学生高效、快速地掌握 C/C++ 语言，训练运用两种语言分析问题、解决问题的能力，为后续的专业课程学习打好坚实的基础，同时为学习更深层次的软件开发与更新的信息技术留出充足的时间。



本书由齐苏敏担任主编，公冶小燕、姜海涛、王抒、齐邦强、叶传秀担任副主编。其中，第1、2章由王抒编写，第3、4章由姜海涛编写，第5、7章由公冶小燕编写，第6章由叶传秀编写，第8~12章由齐苏敏编写，齐邦强和齐苏敏负责对全书统稿。

本书的编写和出版得到了山东省教学改革项目和曲阜师范大学教材建设基金的资助，得到了编者所在教学团队的支持，在此深表感谢。

由于编者水平有限，书中难免有疏漏之处，恳请读者批评指正，不胜感激。联系方式：qfasm@126.com。

编 者

2019年5月

上篇 C 语言篇

第 1 章 计算思维与 C 语言概述	(3)
1.1 计算思维与程序设计语言	(3)
1.1.1 二进制思维与程序设计语言的分类	(3)
1.1.2 计算思维与程序设计	(5)
1.1.3 计算思维在 C 语言程序设计中的应用	(5)
1.2 C 程序的基本结构	(5)
1.2.1 注释	(6)
1.2.2 预处理	(6)
1.2.3 主程序	(6)
1.3 C 程序的编译	(7)
1.3.1 UNIX 操作系统的 CC 编译器	(8)
1.3.2 Linux 操作系统的 GCC 编译器	(8)
1.3.3 Windows 操作系统的集成开发环境	(8)
第 2 章 C 语言程序的基本元素与顺序结构程序设计	(10)
2.1 C 语言程序的基本元素	(10)
2.1.1 变量和常量及数据类型	(10)
2.1.2 表达式	(13)
2.1.3 数据的输入与输出	(15)
2.2 顺序结构程序设计	(18)
2.2.1 语句	(18)
2.2.2 构造程序	(19)
2.2.3 编程风格	(20)
2.3 实训与实训指导	(20)



实训 1	字母大小写的转换	(20)
实训 2	两数相除	(21)
实训 3	交换两个数的值	(23)
实训 4	输出随机数	(24)
第 3 章	逻辑运算与选择结构程序设计	(26)
3.1	选择结构程序的构成	(26)
3.1.1	关系运算符与关系表达式	(27)
3.1.2	逻辑运算符与逻辑表达式	(28)
3.1.3	if...else 选择结构	(28)
3.1.4	条件运算符和条件表达式	(30)
3.1.5	switch...case 选择结构	(31)
3.2	实训与实训指导	(33)
实训 1	判断奇数/偶数	(33)
实训 2	根据利润计算应发放的奖金	(35)
实训 3	判断闰年	(38)
实训 4	利用海伦公式计算三角形面积	(40)
第 4 章	重复运算与循环结构程序设计	(42)
4.1	三种循环语句	(42)
4.1.1	while 语句	(43)
4.1.2	do...while 语句	(43)
4.1.3	for 语句	(44)
4.1.4	使用 break 语句退出循环	(46)
4.1.5	使用 continue 语句跳过循环体语句	(47)
4.2	批量数据处理——数组	(48)
4.2.1	一维数组	(48)
4.2.2	字符串与字符数组	(50)
4.2.3	二维数组	(53)
4.3	实训与实训指导	(56)
实训 1	数制转换	(56)
实训 2	输出乘法表	(57)
实训 3	兑换硬币	(59)
实训 4	冒泡排序	(60)
第 5 章	过程封装——函数	(62)
5.1	函数的定义和调用	(62)
5.1.1	函数的定义	(63)
5.1.2	函数的返回	(64)



5.1.3	函数的调用	(65)
5.1.4	函数调用过程	(66)
5.1.5	函数参数的值传递	(67)
5.1.6	函数的声明	(70)
5.2	局部变量和全局变量	(73)
5.2.1	代码块	(73)
5.2.2	局部变量	(74)
5.2.3	全局变量	(76)
5.2.4	作用域规则	(78)
5.3	变量的存储类型	(80)
5.3.1	auto 变量	(80)
5.3.2	register 变量	(80)
5.3.3	static 变量	(81)
5.3.4	extern 变量	(84)
5.4	实训与实训指导	(86)
实训 1	直角三角形	(86)
实训 2	一元二次方程的根	(88)
实训 3	完美数	(89)
实训 4	玫瑰花数	(91)
第 6 章	数据的间接访问——指针	(95)
6.1	指针的基本概念	(95)
6.1.1	指针与地址	(95)
6.1.2	指针变量的定义	(96)
6.1.3	指针变量的初始化与赋值	(96)
6.1.4	指针变量的访问	(97)
6.2	指针与数组	(98)
6.2.1	用指针操作数组	(98)
6.2.2	动态内存分配	(99)
6.2.3	数组作为函数参数	(101)
6.3	指针数组和指向指针的指针	(103)
6.3.1	指针数组	(103)
6.3.2	指向指针的指针	(105)
6.4	指针作为函数的形参	(106)
6.5	函数指针和指针函数	(108)
6.5.1	函数指针	(108)
6.5.2	函数指针作为函数参数——回调函数	(111)
6.5.3	指针函数	(112)
6.6	实训与实训指导	(115)



实训 1 将整型数转换为字符串	(115)
实训 2 日期转换函数	(116)
实训 3 字符串排序	(118)
实训 4 函数指针应用	(120)
第 7 章 函数的自我调用——递归	(122)
7.1 递归	(122)
7.1.1 递归的思想	(123)
7.1.2 递归的递推	(123)
7.1.3 递归的回归	(124)
7.1.4 递归的条件	(125)
7.1.5 递归的实现	(125)
7.2 迭代与递归	(126)
7.2.1 递归实现	(126)
7.2.2 迭代实现	(127)
7.2.3 递归与迭代的关系	(128)
7.3 实训与实训指导	(129)
实训 1 走台阶	(129)
实训 2 换汽水	(130)
实训 3 排列数	(132)
实训 4 汉诺塔	(134)
第 8 章 数据封装——用户自定义数据类型	(136)
8.1 结构体的定义与使用	(136)
8.1.1 定义结构体类型和结构体变量	(137)
8.1.2 初始化结构体变量	(139)
8.1.3 访问结构体成员	(140)
8.1.4 结构体作为函数参数或返回值	(141)
8.2 结构体实训与实训指导	(144)
实训 1 计算三维空间中两点之间的距离	(144)
实训 2 利用结构体数组存储一元多项式并输出	(145)
实训 3 创建简单链表	(147)
实训 4 使用链表存储一元多项式并输出	(148)
8.3 使用 FILE 结构体类型的文件操作	(152)
8.3.1 文件的打开与关闭	(152)
8.3.2 文本文件的读写	(154)
8.3.3 二进制文件的读写	(157)
8.3.4 文件的格式化输入与输出	(159)
8.4 共用体	(161)



8.4.1 共用体类型及其变量的定义	(161)
8.4.2 共用体的使用	(163)
8.5 枚举类型	(166)
8.5.1 枚举及其变量的定义	(167)
8.5.2 枚举的使用	(168)

下篇 C++ 语言篇

第9章 面向对象思维与 C++ 语言概述	(173)
9.1 面向对象思维	(173)
9.1.1 C 语言的面向过程思维	(173)
9.1.2 C++ 语言的面向对象思维	(174)
9.1.3 面向对象的基本概念	(174)
9.2 C++ 语言对 C 语言的扩充	(175)
9.2.1 C++ 常量	(176)
9.2.2 C++ 的基本输入输出	(176)
9.2.3 C++ 修饰符类型	(178)
9.2.4 C++ 字符串	(179)
9.2.5 C++ 引用	(182)
9.2.6 C++ 重载函数	(184)
9.2.7 C++ 函数模板	(185)
9.2.8 C++ 动态内存	(186)
9.2.9 C++ 异常处理	(187)
9.3 C++ 程序的编译	(189)
9.4 实训与实训指导	(189)
实训1 旋转魔方阵	(189)
实训2 删除重复字符	(192)
实训3 字符串全排列	(195)
实训4 求两组整数的异或集	(196)
第10章 数据与过程的封装——类及其实训	(199)
10.1 定义类与对象	(199)
10.1.1 定义类	(200)
10.1.2 声明对象	(203)
10.1.3 操作对象	(203)
10.1.4 对象的内存分配与 this 指针	(204)
10.2 对象的构造与析构	(205)
10.2.1 构造函数	(206)



10.2.2	复制构造函数	(208)
10.2.3	析构函数	(208)
10.3	const 与数据保护	(211)
10.3.1	常数据成员	(211)
10.3.2	常成员函数	(212)
10.3.3	常对象	(213)
10.3.4	对象的常引用	(213)
10.4	类的静态成员与数据共享	(214)
10.4.1	静态数据成员	(215)
10.4.2	静态成员函数	(215)
10.4.3	静态常量成员	(217)
10.5	类的对象成员——类的组合	(217)
10.5.1	声明类的对象成员	(218)
10.5.2	在成员函数中使用类的对象成员	(219)
10.6	友元	(220)
10.6.1	友元函数	(220)
10.6.2	友元类	(221)
10.7	实训与实训指导	(222)
	实训 1 分数类	(222)
	实训 2 时钟类	(224)
	实训 3 随机数类	(226)
	实训 4 约瑟夫环类	(228)
第 11 章	对象运算——运算符重载及其实训	(233)
11.1	运算符重载的方法	(233)
11.1.1	运算符重载函数作为友元函数	(234)
11.1.2	运算符重载函数作为成员函数	(235)
11.1.3	运算符重载的限制	(237)
11.2	特殊运算符的重载	(237)
11.2.1	输入/输出运算符重载	(238)
11.2.2	赋值运算符重载	(240)
11.2.3	函数调用运算符重载	(241)
11.2.4	下标运算符重载	(242)
11.3	自定义类型与基本类型之间的转换	(243)
11.3.1	基本类型到自定义类型的转换	(243)
11.3.2	自定义类型到基本类型的转换	(244)
11.4	实训与实训指导	(245)
	实训 1 二维数组类	(245)
	实训 2 布尔类	(246)



实训 3 随机类运算符重载	(251)
实训 4 一元多项式类	(252)
第 12 章 代码重用——类的继承、多态与模板	(258)
12.1 类的继承与派生	(258)
12.1.1 基类的定义	(259)
12.1.2 派生类的定义与继承方式	(260)
12.1.3 派生类的构造函数和析构函数	(261)
12.1.4 重定义基类的成员函数	(261)
12.1.5 派生类的定义	(262)
12.1.6 类的多层派生	(264)
12.2 多重继承	(266)
12.2.1 多重继承的声明	(269)
12.2.2 多重继承派生类对象的定义	(270)
12.2.3 虚基类	(271)
12.3 多态性与虚函数	(272)
12.3.1 派生类对象向基类对象的转换	(273)
12.3.2 虚函数	(274)
12.3.3 纯虚函数	(277)
12.3.4 抽象类	(277)
12.4 类模板和泛型编程	(278)
12.4.1 类模板的定义	(279)
12.4.2 类模板的实例化	(281)
12.4.3 类模板的友元	(282)
12.5 实训与实训指导	(284)
实训 1 泛化的链表类	(284)
实训 2 图书馆系统中的读者类	(286)
实训 3 读者库	(290)
附录	(292)
参考文献	(295)



上篇 C 语言篇

第 1 章

计算思维与 C 语言概述

随着计算机技术的迅猛发展，计算机应用从早期的数学计算发展到各种媒体信息的处理，现已渗透到了人们工作、生活的各个角落，计算思维也被认为是与理论思维、实验思维并列的第三种思维模式。

1.1 计算思维与程序设计语言

计算机系统由硬件和软件组成。硬件是计算机的“躯体”，而软件是计算机的“灵魂”，计算机系统的运行需要软件的驱动，而缺少了硬件的计算机系统是不能独立存在的，硬件和软件相辅相成，二者缺一不可。软件的开发需要程序的设计与编写，程序是为完成某一特定任务而定义的一组指令的序列。要想计算机完成一项新任务，就需要设计与编写程序，让计算机自动执行。

卡内基·梅隆大学前计算机系主任、微软公司前副总裁周以真（Jeannette M. Wing）教授指出：“计算思维是运用计算（机）科学的基础概念去求解问题、设计系统和理解人类行为的一系列思维活动的统称。”计算思维建立在计算过程的能力和限制之上，由人或机器执行。

1.1.1 二进制思维与程序设计语言的分类

计算机系统采用二进制来表示数值型信息。二进制的每一位基值只有 0 和 1，因易于电子器件实现而应用于计算机系统。也就是说，计算机系统是 0 和 1 的世界，使用 0 和 1 表示数值，也表示逻辑（0 为假，1 为真），从而实现自动化。0 和 1，以及逻辑运算代表的二进



制思维是计算思维的基础。

因此，人类世界中的信息只有表示为 0 和 1 的组合，才能进入计算机系统。早期的程序员就使用二进制来编写程序让计算机执行，从而完成某种任务。这种二进制语言称为机器语言。机器语言是机器指令的集合。机器指令就是计算机能够直接识别并执行的指令。计算机的机器指令是一个二进制编码。例如，应用 8086 CPU 完成计算 $s = 768 + 12288 - 1280$ 的三条机器指令如下：

```
10110000000000000000000011
```

```
000001010000000000110000
```

```
001011010000000000000101
```

假如将指令误写成以下形式，运算将错误。

```
10110000000000000000000011
```

```
000001010000000000011000
```

```
001011010000000000000101
```

不难看出，用机器语言编写程序是非常困难的。无论是记住这些二进制编码，还是找出其中的一些错误，都非常麻烦。

早期的程序员们很快就发现了使用机器语言带来的麻烦，于是，汇编语言产生了。汇编语言是汇编指令的集合。汇编指令采用了类似人类所使用的自然语言的语法来表示这些指令，从而便于程序员阅读和记忆。例如，将寄存器 BX 的内容传送到寄存器 AX 的机器指令是“1000100111011000”，而对应的汇编指令则为“MOV AX, BX”。很明显，后者更便于程序员阅读和记忆。由于计算机只能识别机器指令，因此需要将采用汇编语言编写的程序翻译成计算机能够识别的指令序列，这一工作由专门程序来完成。

汇编指令与机器指令基本上——对应，它的执行与机器语言一样受硬件底层平台的限制。更重要的是，用多条指令实现一个程序的编写过于烦琐。于是，高级语言产生了！高级语言是对汇编语言的进一步抽象，它更接近于人类使用的自然语言，同时又不依赖于计算机硬件，编出的程序能在不同体系结构的计算机上执行。例如，求两个数的最大值的 C 语言代码如下所示：

```
if(a > b)
    max = a;
else
    max = b;
```

可见，高级语言更接近于人类的自然语言描述。但由于计算机只能识别机器语言，因此用高级语言编写的程序也需要经过专门的编译器程序翻译成机器指令，才能在计算机上执行。

综上所述，程序员发现使用更接近于自然语言的高级语言更容易实现计算机软件的设计与开发，只需运用计算思维来解决问题，而无须考虑计算机硬件平台的要求与限制。

1.1.2 计算思维与程序设计

以计算机学科为代表的计算思维又称构造思维，其以设计和构造为特征，最本质的内容是抽象与自动化。抽象与自动化反映了计算的根本问题，即什么能被有效地自动进行。程序是让计算机自动执行的指令序列，数据、指令和程序是计算思维最基本的内容，程序设计与构造是一种计算思维。

程序设计语言和计算思维相辅相成，语言是思维的体现，思维是语言的载体。对于程序设计课程来说，学生应掌握的编程能力是计算思维和技能化知识的综合体，实践操作是对计算思维能力结果的一种验证。

1.1.3 计算思维在 C 语言程序设计中的应用

C 语言是一种面向过程的程序设计语言，是目前世界上普遍流行、使用非常广泛的高级程序设计语言之一，由美国贝尔实验室的 Dennis M. Ritchie 设计。当前最新的 C 语言标准为 C11，在它之前的 C 语言标准为 C99。鉴于 C 语言对底层硬件操作方面的优势，C 语言广泛应用于操作系统（如 Windows、Linux、UNIX 等操作系统）、工业控制等软件的开发。另外，C 语言具有绘图能力强、可移植性好的特点，并具备很强的数据处理能力，因此也适用于二维、三维图形动画软件的开发。

计算思维是模型与算法相结合的思维过程，是抽象与自动化实现的过程。在 C 语言程序设计中，先把待解决的问题抽象成与其相应的模型，然后确定算法，编写程序求解问题，最后由机器自动执行。一个 C 语言程序的设计与构造，首先将简单的语句组合成复杂的结构化语句，进而抽象为函数，最后将多个函数组合为复杂的程序。函数机制有利于实现穷举法、递推法、递归法、回溯法、迭代法、分治法、贪心法和动态规划法等典型算法，同时使程序更简短而清晰，有利于维护，并提高代码的重用性，从而提高程序开发的效率。

1.2 C 程序的基本结构

先来看一个最小的 C 程序——hello.c，以此来了解 C 程序的基本结构。该程序能在屏幕上输出以下内容：

```
Hello,World!
```

代码清单：

```
/* 文件名:hello.c           ]—注释  
功能:输出"Hello,World!" * /  
#include <stdio.h >       ]—预处理命令
```