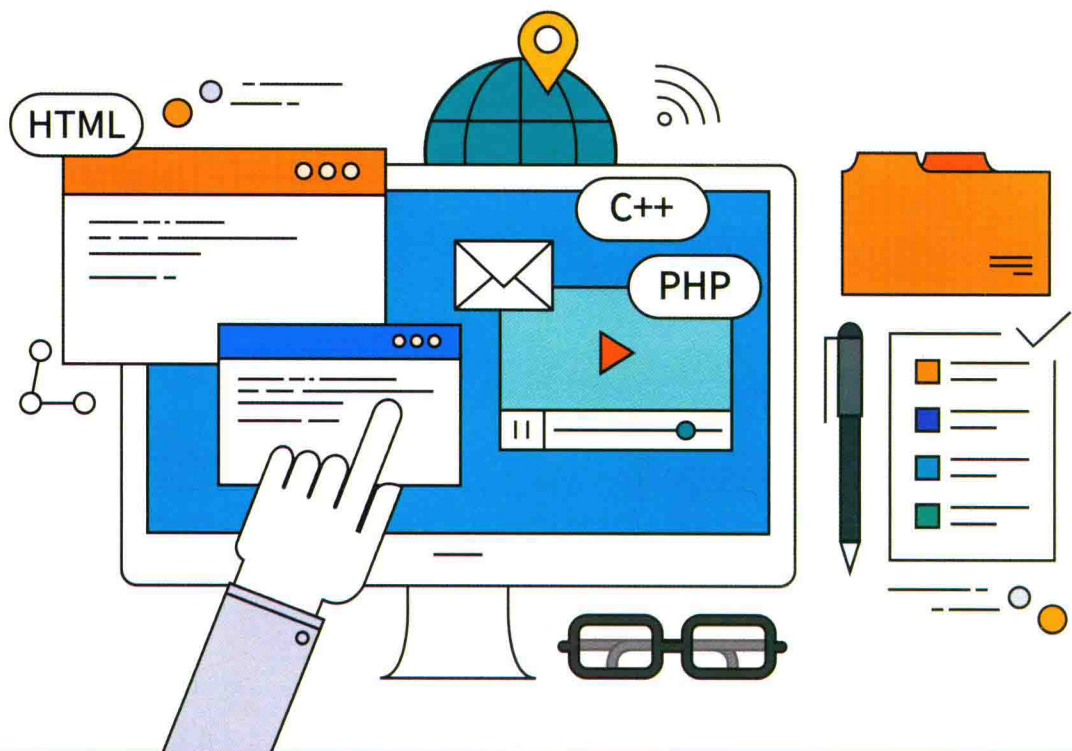




“十二五”普通高等教育本科国家规划教材
高等教育计算机学科“应用型”教材



C/C++ 程序设计教程—— 面向过程分册 (第3版)

郑秋生 夏敏捷◎主编 王黎明◎主审



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

“十二五”普通高等教育本科国家级规划教材
高等教育计算机学科“应用型”教材

C/C++程序设计教程——面向 过程分册（第3版）

郑秋生 夏敏捷 主 编
周雪燕 罗 菁 副主编
王黎明 主 审

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

《C/C++程序设计教程》系列教材分为面向过程和面向对象两个分册。面向过程分册系统阐述了C++语言中过程化程序设计的思想、语法、方法。主要内容包括C++程序设计的基础知识,基本数据类型和表达式,C++的程序控制语句,数组与函数,指针和引用,用户定义数据类型、文件等内容。书中内容讲解清晰,实例丰富,力避代码复杂冗长,注重算法设计和程序设计思想。简短的实例特别有助于初学者更好地理解、把握解决问题的精髓,帮助读者快速掌握程序设计的基本方法。

本教材的特点是实例丰富,重点突出,叙述深入浅出,分析问题透彻,既有完整的语法,又有大量的实例,突出程序设计的算法、方法,将C语言程序设计和C++语言程序设计有机地进行统一。特别适合作为计算机学科各应用型本科、专科的C语言程序设计和C++语言程序设计教材,也可作为其他理工科各专业的教材及相关技术人员的自学参考书。

本教材配有免费课件资源,有需要的读者可到华信教育资源网(www.hxedu.com.cn)下载使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C/C++程序设计教程. 面向过程分册 / 郑秋生, 夏敏捷主编. —3 版. —北京: 电子工业出版社, 2017.8
高等教育计算机学科“应用型”教材

ISBN 978-7-121-31748-4

I. ①C… II. ①郑… ②夏… 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(2017)第124114号

策划编辑: 张贵芹 刘 芳

责任编辑: 康 霞

印 刷: 三河市华成印务有限公司

装 订: 三河市华成印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本: 787×1092 1/16 印张: 21.25 字数: 544千字

版 次: 2007年9月第1版

2017年8月第3版

印 次: 2017年8月第1次印刷

定 价: 39.80元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254511, 82470125@qq.com。

编 委 会

主 任：蒋宗礼

副主任：周清雷 甘 勇 王传臣

委 员：（按姓氏音序为序）

陈志国 贾宗璞 普杰信 钱晓捷

王爱民 王清贤 翁 梅 邬长安

徐久成 张红梅 张亚东 郑秋生

秘书组：钱晓捷 张 旭

第 3 版说明

本教材第 1、2 版是由河南省计算机学会和电子工业出版社共同组织教学一线教师，编写的一套面向应用型本科教学的教材，属于高等教育计算机学科应用型规划教材。教材充分体现了教师多年的教学研究成果和教学经验，具有较好的质量和社会影响力。2007 年出版第 1 版和 2011 年出版第 2 版后，在郑州大学、中原工学院、河南科技大学、郑州轻工业学院和河南工程学院等河南省高校推广使用，得到任课教师和读者的普遍欢迎。通过电子工业出版社和新华书店的努力，被国内许多高校广泛使用。

为了更好地推进高等学校本科教学质量与教学改革工程，结合重点专业建设、精品课程和精品教材建设目标，以及培养学生的工程能力和创新能力的需要，对教材进行修订再版。第 2 版根据教学中发现的问题和读者的建议反馈，以及为了达到提升学生的程序设计能力、调试能力的目的再次进行了编写。第 3 版的主要变化表现在以下几个方面。

(1) 增加了程序调试的内容。大多数学生在学习过程中的上机环节，将课本的源代码直接输入计算机，如果程序运行正确就完成任务，而不去理解算法、思考更多的问题，对程序理解和编程能力的提高作用不大；如果程序不能够运行或运行不正确，由于缺乏程序调试、解决问题能力的训练，学生往往放弃该程序的调试，不仅丢掉了一次思考、学习和提高的机会，也使实际的程序设计能力培养受到影响，致使学生在后续教学环节——课程实验、课程设计、毕业设计和项目开发中仍然不会编程，始终缺乏程序调试和解决问题的实践能力。针对以上问题和现象，需要进行教学内容和教学方法的改革，切实在教学过程中，重视学生程序调试能力和技巧的培养与锻炼，所以第 3 版中每章都增加了一节上机调试的内容，让学生循序渐进地学习和掌握程序调试的基本技巧和知识，逐渐具备程序调试和实际项目开发的能力，具体安排如下。

第 1 章介绍 Visual C++ 6.0 集成环境及上机调试步骤。

第 2 章介绍编译过程常遇到的错误原因。

第 3 章学会单步调试、查看变量和表达式。

第 4 章介绍 Step Into (Out) 两种调试方式。学会参数传递时，查看实参和形参的值，并通过 Call Stack 窗口查看函数的调用关系。

第 5 章介绍如何在集成环境中建立多文件，学会查看不同存储类型变量的值、变量内存地址。

第 6 章举例介绍位置断点、条件断点、数据断点三种类型断点的调试方法。

第 7 章查看结构体成员内存分配，引出共用体内存分配问题。

第 8 章查看指针变量的值，并通过 Memory 窗口查看该地址的内容，以及介绍如何查看函数的地址。

第 9 章用 UltraEdit 软件抓图演示文本文件和二进制文件存储时的区别，读取文件时数据在内存的表现。

(2) 教材中的语法和程序代码遵循 C11/C++11 国际标准, 以适应 C/C++ 语言的发展。

例如, `main` 函数的返回类型一律指定为 `int` 型, 并在 `main` 函数的末尾加一个“`return 0;`”, 使用标准命名空间 `using namespace std`。

(3) 修订了第 2 版的一些印刷错误和不准确的表述方法。

(4) 修改、增加了一些例题、习题和章节内容, 希望本版能更加适合教师的教学。

通过十年的使用, 本教材的任课教师、学生和读者都给予了许多肯定、鼓励, 也提出了许多有意义的建议、意见和再版的意愿, 在此代表所有作者感谢读者对本教材的厚爱 and 关心。

编 者

2017 年 5 月

前 言

本教材的主要作者都是有着丰富教学经验的一线教师，从事 C/C++ 程序设计课程教学多年，深知学生在学习 C++ 程序设计这门课程后，对程序设计方法、算法设计、调试程序、习题解答的茫然和问题，因此本书在介绍理论知识、相关概念和语言语法时，始终强调其在程序设计中的作用，使语言语法与程序设计相结合。同类书籍大部分偏重于对语言语法和概念的介绍，虽然在书中有针对一个语法和知识点的程序实例，但学生对每章内容在实际程序设计中的作用缺乏了解，而本书每章后都附有针对性较强的应用实例分析，尽可能使初学者在学习每章的内容后拿到题目，即能够独立进行程序设计、解决实际问题，而不至于无从下手。本书有以下五个鲜明特点。

第一，改变了传统的教学模式。先讲 C 语言程序设计，再讲 C++ 对 C 语言的扩展、面向对象的程序设计。本教材将 C/C++ 语言的学习很好地融合在一起，让读者把面向过程和面向对象的程序设计方法有机地结合在一起，面向过程和面向对象两分册都统一使用 Visual C++ 6.0 编译器。

第二，改变了传统教材以语言、语法学习为重点的缺陷，本教材从基本的语言、语法学习上升到程序的“设计、算法、编程、调试”层次。为了让学生更好地掌握程序开发的思想、方法和算法，书中提供了大量简短精辟的代码，有助于初学者学习解决问题的精髓。在每章后都有一节关于程序综合设计的内容，有一个或多个较大的程序，以帮助学生更好地掌握程序设计方法和提高解决实际问题的能力。

第三，教材强调程序的设计方法，大量例题有流程图、N-S 图和 UML 图，并且每章都有上机调试的内容。教材突出程序的算法和设计，而不仅是语法和编程，培养学生的程序设计能力和程序调试技能，养成好的编程习惯，为专业程序员的培养打下良好的基础。

第四，培养学生面向对象程序设计的能力，引导学生建立程序设计的大局观，帮助学生掌握从客观事物中抽象出 C++ 类的方法。通过系统的学习，使学生的编程能力上一个台阶，具备解决复杂问题的程序设计能力。

第五，根据当前实际大型软件项目开发的需要，扩充了异常处理、模板等内容，新增 STL 标准模板库，并通过流行的 UML 工具设计 C++ 类。

本教材的编写充分考虑了目前应用型本科 C/C++ 语言程序设计课程教学的实际情况和存在的问题。第一，学生在大一阶段的基础课程较多，不可能投入过多的精力来学习本门课程；第二，大学生对这门课学习的期望值很高，但对学习时可能遇到的困难估计不足；第三，大学生现有的上机实践条件大大改善，特别有利于贯彻先进的精讲多练的教学思想；第四，学生学会了语言的语法，仍不具备解决实际问题的能力，学生的程序设计、算法设计、编程、调试能力相对较差。本教材作者正是考虑了学生的这些实际问题，从而精心编写了这一套面向应用型本科的 C/C++ 程序设计教程，特别适合于分两个学期系统讲授 C/C++ 程序设计。第 1 学期讲授面向过程分册，第 2 学期讲授面向对象分册。

本分册共分 9 章，第 1 章主要讨论 C++ 语言的特点和编辑环境，第 2~8 章主要介绍用 C/C++ 进行过程化程序设计的基本方法，内容包括表达式及运算符、数据类型、函数、数组、

指针等，第9章主要介绍文件处理方法。

为了方便使用本教材的教师备课，我们还提供了配套的电子教案，公开放在网站上，供任课教师自由下载使用。相信我们多年的教学经验会对广大师生的教和学有所帮助。建议本分册的教学为60学时，其中理论教学为44学时，课内上机实践为16学时。课外上机不少于32学时。

本教材的编写得到了河南省计算机学会的大力支持，河南省计算机学会组织了河南多所高校编写了高等教育计算机学科“应用型”系列教材。参编本教材编写的高校有中原工学院、郑州大学、河南科技大学。

本分册第1、2、3章由夏敏捷和周雪燕编写，第4、5章由刘姝编写，第6章由潘惠勇编写，第7章由郑秋生编写，第8、9章和附录由罗菁和马宗梅编写。全书最终由郑秋生修改并统稿，并由王黎明主审。为本书提出改进意见和建议的老师有郑州大学的钱晓捷和卢红星教授，在此谨向他们表示衷心的感谢。

由于编者水平有限，加之时间仓促，书中难免有错，敬请广大读者批评指正，在此表示感谢。E-mail: zqs@zut.edu.cn。

编者
2017年5月

目 录

第 1 章 C++概述	1
1.1 计算机程序设计语言的发展	3
1.1.1 机器语言阶段	3
1.1.2 汇编语言阶段	3
1.1.3 高级语言阶段	4
1.1.4 从 C 到 C++	5
1.2 过程化程序设计	5
1.3 面向对象的程序设计	6
1.3.1 基本概念	7
1.3.2 面向对象程序设计的特点	7
1.4 简单的 C/C++程序介绍	8
1.5 程序开发的过程	11
1.6 C/C++上机实践	12
1.6.1 Visual Studio 2013 集成 开发环境	12
1.6.2 开发 C/C++的程序过程	15
习题 1	17
第 2 章 数据类型、运算符和表达式	19
2.1 保留字和标识符	21
2.1.1 保留字	21
2.1.2 标识符	21
2.2 数据类型	21
2.3 常量与变量	23
2.3.1 常量	23
2.3.2 变量	25
2.4 基本运算符和表达式	26
2.4.1 基本运算符和表达式简介	26
2.4.2 算术运算符和算术表达式	27
2.4.3 赋值运算符和赋值表达式	28
2.4.4 关系运算符和关系表达式	29
2.4.5 逻辑运算符和逻辑表达式	30
2.4.6 位运算符和位运算表达式	31
2.4.7 条件运算符和条件表达式	33
2.4.8 逗号运算符和逗号表达式	33
2.4.9 sizeof 运算符和 sizeof 表达式	33
2.5 数据类型转换	33
2.5.1 隐式转换	34
2.5.2 显式转换	34
2.6 简单的输入/输出实现方法	34
2.6.1 格式化输入/输出 ——scanf()和 printf()	34
2.6.2 I/O 流	37
2.6.3 cin	38
2.6.4 cout	38
2.6.5 输出控制符	39
2.7 认识上机过程中的错误	41
习题 2	42
第 3 章 C++流程控制	45
3.1 算法与流程图	47
3.1.1 算法的概念	47
3.1.2 算法的描述	47
3.2 语句和程序的三种基本结构	48
3.2.1 语句	48
3.2.2 程序的三种基本结构	49
3.2.3 结构化算法	49
3.3 顺序结构程序	50
3.4 选择结构程序	51
3.4.1 if 语句	51
3.4.2 嵌套 if 语句	53
3.4.3 switch 语句	55
3.5 循环结构程序设计	57
3.5.1 while 语句	57
3.5.2 do-while 语句	60
3.5.3 for 语句	62
3.5.4 循环的嵌套	64
3.5.5 跳转语句	67
3.5.6 三种循环的比较	69
3.6 常用算法及应用实例	69
3.6.1 累加与累乘	69
3.6.2 求最大数、最小数	70

3.6.3	求素数	71	5.2.4	外部类型	131
3.6.4	穷举法	73	5.3	编译预处理	132
3.6.5	递推与迭代	74	5.3.1	宏定义命令	132
3.7	应用实例	77	5.3.2	文件包含命令	135
3.8	程序的调试	79	5.3.3	条件编译命令	135
3.8.1	进入调试	80	5.4	程序的多文件组织	137
3.8.2	单步调试	80	5.4.1	头文件	137
3.8.3	查看变量、表达式的值	81	5.4.2	多文件结构	137
3.8.4	停止调试	81	5.4.3	多文件结构程序示例	138
习题 3		82	5.5	多文件程序的创建与调试	140
第 4 章	函数	85	习题 5		142
4.1	函数的定义	87	第 6 章	数组	145
4.1.1	函数定义	87	6.1	数组的概念	147
4.1.2	函数的返回值	88	6.1.1	数组与数组元素	147
4.2	函数的调用	89	6.1.2	数组的维数	147
4.2.1	函数调用的形式及过程	89	6.2	一维数组的定义及应用	147
4.2.2	函数的声明	90	6.2.1	一维数组的定义和初始化	147
4.2.3	函数调用的参数传递方式	92	6.2.2	一维数组的操作	149
4.3	函数的嵌套调用和递归调用	94	6.2.3	数组的越界问题	150
4.3.1	函数的嵌套调用	94	6.2.4	一维数组的应用	150
4.3.2	函数的递归调用	95	6.2.5	基于范围的 for 循环	154
4.4	内联函数和函数重载	100	6.3	字符数组的定义及应用	155
4.4.1	内联函数	100	6.3.1	字符数组和字符串	155
4.4.2	函数重载	101	6.3.2	字符串处理函数	157
4.5	函数的参数	105	6.3.3	字符数组应用举例	159
4.5.1	函数参数的求值顺序	105	6.4	二维数组	161
4.5.2	具有默认参数值的函数	106	6.4.1	二维数组的定义	161
4.6	应用实例	107	6.4.2	二维数组的初始化	162
4.7	函数的调试	112	6.4.3	二维字符数组	164
习题 4		114	6.4.4	二维数组应用	165
第 5 章	作用域和存储类型	117	6.5	数组作为函数参数	169
5.1	作用域	119	6.5.1	向函数传递一维数组	169
5.1.1	作用域分类	119	6.5.2	向函数传递二维数组或 多维数组	172
5.1.2	局部变量与全局变量	121	6.6	数组应用实例	173
5.1.3	动态存储方式与静态 存储方式	125	6.7	断点调试方法	175
5.2	变量的存储类型	126	习题 6		183
5.2.1	自动类型	126	第 7 章	结构体、共用体和枚举类型	189
5.2.2	寄存器类型	127	7.1	结构体的定义及使用	191
5.2.3	静态类型	128	7.1.1	结构体的定义	191

7.1.2	结构体变量的定义	192	8.5.2	返回值为指针类型的函数	245
7.1.3	结构体变量的使用	194	8.5.3	函数指针	246
7.1.4	结构体变量的初始化	195	8.6	const 指针	249
7.1.5	结构体数组	195	8.6.1	指向常量的指针变量的 定义与使用	249
7.1.6	结构体和函数	200	8.6.2	指针常量	250
7.2	共用体的定义与使用	201	8.6.3	指向常量的指针常量	251
7.2.1	共用体的概念	201	8.7	结构体指针	251
7.2.2	共用体变量的定义	202	8.7.1	结构体指针的概念	251
7.2.3	共用体变量的使用	202	8.7.2	指向结构体数组元素 的指针	252
7.3	枚举类型	204	8.7.3	结构体指针作为函数参数	253
7.4	typedef 定义类型	206	8.8	链表	255
7.5	应用实例	207	8.8.1	链表概述	255
7.6	程序调试	211	8.8.2	链表的基本操作	256
7.6.1	结构体变量各成员变量的 内存分配情况	212	8.9	引用	263
7.6.2	共用体变量各数据成员的 内存分配情况	213	8.9.1	引用及声明方法	263
习题 7		214	8.9.2	用引用作为函数的参数	265
第 8 章	指针和引用	217	8.9.3	如何使一个被调函数同时 返回多个值	266
8.1	指针与指针变量	219	8.9.4	用 const 限定引用	267
8.1.1	地址与指针的概念	219	8.9.5	用引用作为函数返回值	268
8.1.2	指针变量	220	8.9.6	引用总结	269
8.1.3	指针变量的运算	224	8.10	综合应用实例	269
8.1.4	void 指针	226	8.11	上机调试	274
8.1.5	C++11 的扩展	227	8.11.1	指针变量值的调试查看	274
8.2	指针与数组	227	8.11.2	函数地址的调试查看	275
8.2.1	一维数组与指针	227	8.11.3	引用的调试查看	276
8.2.2	二维数组与指针	229	8.12	小结	277
8.2.3	指向数组的指针	231	习题 8		277
8.2.4	指针数组	232	第 9 章	文件	283
8.2.5	指向指针的指针	233	9.1	C 语言文件概述	285
8.3	字符指针与字符串	234	9.2	文件结构体与文件指针	286
8.3.1	字符数组与字符指针	234	9.3	文件的打开与关闭	287
8.3.2	字符指针数组	236	9.3.1	文件的打开 (fopen 函数)	287
8.4	动态内存分配和释放	236	9.3.2	文件的关闭 (fclose 函数)	288
8.4.1	C++ 中堆的使用	237	9.4	文件的读写	289
8.4.2	C 语言中动态分配空间 的函数	239	9.4.1	按字符读写文件 (fgetc 和 fputc 函数)	289
8.5	指针与函数	240	9.4.2	字符串的读写 (fgets 和 fputs 函数)	293
8.5.1	指针变量与数组名作 函数参数	240			

9.4.3	格式化的读写 (fscanf 和 fprintf 函数)	295
9.4.4	数据块的读写 (fread 和 fwrite 函数)	298
9.5	文件的定位	300
9.5.1	位置指针复位 (rewind 函数)	301
9.5.2	位置指针随机定位 (fseek 函数)	302
9.5.3	检测当前位置指针的位置 (ftell 函数)	305
9.6	文件的检测	305

9.6.1	文件读写错误检测 (ferror 函数)	305
9.6.2	清除文件错误标志 (clearerr 函数)	306
9.6.3	文件结束检测 (feof 函数)	306
9.7	常用文件操作函数小结	306
9.8	程序设计举例	307
9.9	上机调试	313
习题 9		318

附录 A	ASCII 码表	321
------	----------------	-----

附录 B	C++ 的库函数	323
------	----------------	-----

参考文献		326
------	-------	-----

第 1 章

C/C++概述

C++语言是一种高效率实用的程序设计语言，使用它既可进行过程化程序设计，也可进行面向对象的程序设计。本章主要介绍计算机语言的发展历程、过程化程序设计和面向对象的程序设计方法、C++程序的基本结构及程序开发的步骤。

通过本章学习，应该重点掌握以下内容：

- 计算机语言的发展历程。
- 过程化程序设计和面向对象的程序设计的基本思想和主要特点。
- 简单的 C/C++程序结构。
- C/C++程序开发的步骤。

1.1 计算机程序设计语言的发展

计算机语言通常是能完整、准确和规则地表达人们意图，并用于指挥或控制计算机工作的“符号系统”。当使用计算机解决问题时，首先将解决问题的方法和步骤按照一定的顺序和规则用计算机语言描述出来，形成指令序列，然后由计算机执行指令，完成所需的功能。

计算机程序设计语言的发展，经历了从机器语言、汇编语言到高级语言的历程。

1.1.1 机器语言阶段

众所周知，在计算机内部采用二进制表示信息。机器语言（Machine Language）是用二进制代码表示的、计算机能直接识别和执行的一种机器指令的集合。它是面向机器的语言，是计算机唯一可直接识别的语言。用机器语言编写的程序称为机器语言程序（又称目标程序）。每一条机器指令的格式和含义都是由设计者规定的，并按照这个规定设计、制造硬件。一个计算机系统全部机器指令的总和称为指令系统。不同类型的计算机的指令系统不同。

例如，某种计算机的指令如下：

```
10110110 00000000    //表示进行一次加法操作
10110101 00000000    //表示进行一次减法操作
```

它们的前 8 位表示操作码，而后 8 位表示地址码。从上面两条指令可以看出，它们只是在操作码中从左边第 0 位算起的第 6 和第 7 位不同。这种机型可包含 256（即 2^8 ）个不同的指令。

用机器语言编写的程序能直接在计算机上运行，运行的速度快，效率高，但机器语言难以记忆，也难以操作，代码编程烦琐、易出错，而且编写的程序紧密依赖计算机硬件，程序的可移植性差。

机器语言是第一代计算机语言。

1.1.2 汇编语言阶段

为了克服机器语言的缺点，使语言便于记忆和理解，人们采用能反映指令功能的助记符来表达计算机语言，称为汇编语言（Assembly Language）。汇编语言采用的助记符比机器语言直观、容易记忆和理解。汇编语言也是面向机器的程序设计语言，每条汇编语言的指令对应了一条机器语言的指令，不同类型的计算机系统一般有不同的汇编语言。

例如，用汇编语言编写的程序如下：

```
MOV    AL  10D    //将十进制数 10 送往累加器
SUB     AL  12D    //从累加器中减去十进制数 12
.....
```

用汇编语言编写程序比用机器语言要容易得多，但计算机不能直接执行汇编语言程序，必须把它翻译成相应的机器语言程序才能运行。将汇编语言程序翻译成机器语言程序的过程叫做汇编。汇编过程是由计算机运行汇编程序自动完成的，如图 1-1 所示。

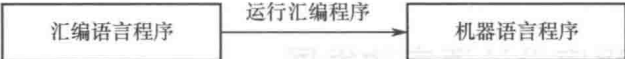


图 1-1 汇编过程

在计算机语言系统中，汇编语言仍然列入“低级语言”的范畴，它也依赖于计算机的硬件，可移植性差，但汇编语言比机器语言在很多方面都有优越性，如编写容易、修改方便、阅读简单、程序清楚等。针对计算机硬件而编制的汇编语言程序，能准确地发挥计算机硬件的功能和特长，程序精练且质量高，所以至今仍是一种常用的程序设计语言。

汇编语言是第二代计算机语言。

1.1.3 高级语言阶段

机器语言和汇编语言都是面向机器（计算机硬件）的语言（低级语言），受机器硬件的限制，通用性差，也不容易学习，一般只适用于专业人员。人们意识到，应该设计一种语言：它接近于数学语言或自然语言，同时又不依赖于计算机的硬件，编出的程序能在所有计算机上通用。高级语言（High-Level Language）是这样的语言。例如，用 C++语言编写的程序片断如下：

```
int i,j,k;           // 定义变量 i,j,k
cin >> i >> j;       // 输入 i,j 的值
k=i*j;              // 将变量 i,j 的值相乘，结果赋给变量 k
cout << k;           // 输出求积结果
```

如上例，使用高级语言编写程序时，不需要了解计算机的内部结构，只要告诉计算机“做什么”即可。至于计算机用什么机器指令去完成（即“怎么做”），编程者不需要关心。高级语言是面向用户的。

用高级语言编写的程序叫做高级语言源程序，计算机无法直接执行，必须翻译或解释成机器语言目标程序才能被计算机执行。翻译过程分为两步，即编译和连接，翻译过程如图 1-2 所示。



图 1-2 翻译过程

在图 1-2 中，高级语言经过编译后，得到目标程序（.obj），再与库程序连接生成可执行程序（.exe）。

程序设计语言从机器语言到高级语言的抽象，带来的主要好处如下。

- （1）高级语言接近自然语言，易学、易掌握，一般工程技术人员只要几周时间的培训就可以胜任程序员的工作。
- （2）高级语言为程序员提供了结构化程序设计的语法，使设计出来的程序可读性好、可维护性强、可靠性高。
- （3）高级语言远离机器语言，与具体的计算机硬件关系不大，因而所编写出来的程序可移植性好，代码重用率高。

(4) 由于把繁杂琐碎的事务交给了编译程序去做, 所以自动化程度高, 开发周期短, 并且程序员得到解脱, 可以集中时间和精力去设计算法和从事更重要的创造性劳动, 以提高程序的质量。

高级语言是第三代计算机语言。目前广泛应用的高级语言有多种, 如 BASIC、FORTRAN、PASCAL、C、C++、JAVA 及 C#等。

1.1.4 从 C 到 C++

C 语言是 AT&T 贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来的, 1972 年在一台 DEC PDP-11 计算机上实现了最初的 C 语言。C 语言最初用做 UNIX 操作系统的开发语言, UNIX 操作系统 90%的代码由 C 语言编写, 10%的代码由汇编语言编写。由于 UNIX 的成功和广泛使用, 也使 C 语言成为一种普遍使用的程序设计语言。

C 语言具有如下优点。

- (1) 语言简洁、紧凑、使用方便、灵活。C 语言只有 32 个关键字, 程序书写形式自由。
- (2) 运算符丰富, 数据结构丰富, 具有现代化语言的各种数据结构。
- (3) 具有结构化的控制语句 (如 if...else 语句、while 语句、for 语句)。
- (4) 语法限制不大严格, 程序设计自由度大。
- (5) C 语言允许直接访问物理地址。
- (6) 生成目标代码质量高, 程序执行效率高。
- (7) 用 C 语言编写的程序可移植性好。

C11 标准是 C 语言标准的第三版, 前一个标准版本是 C99 标准。

但是, C 语言也有它的局限性。

- (1) C 语言数据类型检查机制较弱, 这使程序中的一些错误不能在编译时被自动发现。
- (2) 当程序的规模大到一定程度时, 复杂性很难控制。

为了解决这些问题, 研制 C++语言的一个首要目标就是使 C++语言突破 C 语言的局限性。同时, 在 C++中引入了类等机制来支持面向对象的程序设计。所研制的这个语言最初被称为“带类的 C”, 1983 年取名为 C++ (C Plus Plus)。C++的喻义是对 C 语言进行“增值”。1994 年制定了 ANSI C++ 草案。后来又经过不断完善和发展, 曾有 C++98、C++03、C++11、C++14 等标准。其中 C++98 是第一个正式 C++标准, C++03 在 C++98 上面进行了小幅修订, C++11 (2011 年发布, 包含语言的新机能并且拓展 C++标准程序库) 则是一次全面的大进化, 历经多次修订成为今天的 C++, 且 C++仍在不断的发展中。

同样 C 语言也经历 C89、C99、C11 标准, 新标准提高了对 C++的兼容性, 并将新的特性增加到 C 语言中。

C++是由 C 语言发展而来的, 与 C 语言兼容。C++包含了 C 语言的全部特征、属性和优点, 是 C 语言的超集, 同时 C++添加了面向对象编程的完全支持, 是一种功能强大的面向对象程序设计语言。

1.2 过程化程序设计

程序设计的基本目标是用算法对问题的原始数据进行处理, 从而获得所期望的效果。