



国家精品在线开放课程教材
高等学校计算机技术类课程规划教材

C与C++程序设计

戴波 ◎主 编

陈文字 丘志杰 卢光辉 ◎副主编



C and C++
Programming



北京大学出版社
PEKING UNIVERSITY PRESS



国家精品在线开放课程教材

高等学校计算机技术类课程规划教材

C 与 C++ 程序设计

主 编 戴 波

副主编 陈文字 丘志杰 卢光辉



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本教材一共 10 章,第 1 章介绍软件开发过程及 Visual Studio 2015 开发环境的使用,第 2~6 章介绍 C 语言基础知识及面向过程技术,第 7~10 章介绍 C++ 语言基础知识及面向对象技术。

本教材基础理论浅显易懂,编程案例趣味性强,适合没有编程基础以及学习了 C 语言或者 C++ 语言,仍然不能根据问题独立编写程序的初学者。本教材将理论与实践紧密结合,从分析问题,到寻找解题思路,再到编程、调试、运行,都借助实际案例展开。初学者可以通过扫描二维码来观看视频讲解,从而掌握相关的基础知识及编程过程;并通过视频中的详细演示学会编程方法与调试技术,避免初学者在学习初期常犯的编译错误和后期的运行错误。

本教材提供配套 PPT、视频讲解、编程题库及 QQ 答疑等丰富资源,既适合传统教学使用,又适合翻转课堂教学法的课前预习和课堂编程与讨论;同时,也适合没有教师辅导的独立学习者使用。

图书在版编目(CIP)数据

C 与 C++ 程序设计 / 戴波主编. —北京: 北京大学出版社, 2018. 1
(高等学校计算机技术类课程规划教材)

ISBN 978-7-301-29031-6

I. ①C… II. ①戴… 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 307019 号

书 名	C 与 C++ 程序设计 C YU C++ CHENGXU SHEJI
著作责任者	戴 波 主编
策 划 编 辑	孙 晔
责 任 编 辑	温丹丹
标 准 书 号	ISBN 978-7-301-29031-6
出 版 发 行	北京大学出版社
地 址	北京市海淀区成府路 205 号 100871
网 址	http://www.pup.cn 新浪微博: @北京大学出版社
电 子 信 箱	zyjy@pup.cn
电 话	邮购部 62752015 发行部 62750672 编辑部 62756923
印 刷 者	三河市博文印刷有限公司
经 销 者	新华书店
	787 毫米×1092 毫米 16 开本 19.75 印张 470 千字
	2018 年 1 月第 1 版 2018 年 1 月第 1 次印刷
定 价	48.00 元

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

举报电话: 010-62752024 电子信箱: fd@pup.pku.edu.cn

图书如有印装质量问题, 请与出版部联系, 电话: 010-62756370

高等学校计算机技术类课程规划教材

编 委 会

主 任：蒋宗礼 马殿富

副主任：李文新 孙 晔

委 员(按姓氏拼音排序)：

蔡之华 陈文智 陈志刚 杜小勇

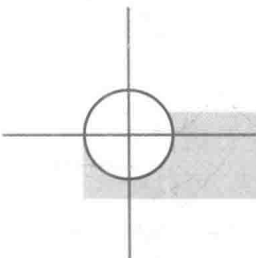
胡学钢 李凤霞 李仁发 林 菲

刘宏伟 马 啸 秦磊华 石 冰

谭国真 陶先平 魏晓辉 王 泉

吴黎兵 肖鸣宇 俞 勇 袁景凌

翟玉庆 张春元 张国印 张星明



前 言

一、编写背景

MOOC(Massive Open Online Courses,大型开放式网络课程,一般称为“慕课”)的兴起,让众多非计算机专业的学生也可以通过网络学习计算机专业课程。为了照顾不同层次的学习者,MOOC 的计算机课程虽然通俗易懂,但是深度与难度却不够;而加大深度与难度后,对于没有教师指导与辅导的初学者来说,要学好并能够使用则极为困难,二者不能兼得。因此,对于计算机专业学生来说,通过 MOOC 学习达到计算机程序设计课程的专业要求,还存在困难。实际情况也显示,不少提前通过 MOOC 学习达到优秀成绩的学生,在进入同一门专业课程学习时仍然会感觉难度很大,他们对许多问题的认识不到位,甚至是错误的。对于非计算机专业的 MOOC 学习者来说,要想深入掌握并灵活应用程序设计语言解决问题,还需要专业训练指导和实验实践。

对于计算机专业的学生来说,目前程序设计语言仍然采用大量的课时进行基础理论教学,少量的课时进行上机训练。这样做存在的问题是,由于初学者没有上机实践的基础,理论学习太抽象,因此只能死记硬背,不能学以致用;等到了上机训练时一点点问题就会被难倒,无法完成一个简单的上机编程实验。这样很容易打击初学者的信心,从而让他们丧失学习兴趣。有的学校试点所有的课程都在机房上,目的是让学生有更多进行上机训练的机会。但实际情况是,有的教师只是换教室而已,教学方法还和以前相同,问题同样存在;有的教师发现学生一做编程题,问题就很多,教师一节课还没有讲什么,下课时间就到了,结果编程没有指导好,而理论课的时间也没有了。翻转课堂教学法可以很好地解决这个问题。首先,学生可以预习简单的基础理论课程。为了照顾部分自学能力比较差的同学,任课教师提前录制基础理论讲解视频,配合 PPT 动画和预习检测题(配答案),让学生每周花 2 个学时的时间自学基础理论。其次,学生可以自由选择最适合自己的资料进行自学,并通过完成简单的预习检测题来检测自己的学习效果。最后,课堂上,教师先通过几分钟的考试来检测学生的预习效果,并根据检测结果,灵活地调整上课内容。如果学生预习效果好,则直接给出应用型的问题引导学生思考、编程解决问题;如果学生预习效果不好,则先进行基础理论的复习,可以只针对难点进行细致地讲解,然后再引导学生理解如何通过编程解决问题的方式灵活地运用基础理论知识解决实际问题。

C 语言和 C++ 语言既有区别又有联系,具有广泛的应用场合,可以用来编制各种系统软件与应用软件,是大多数学校的计算机专业必修课程,但一般分成 2 个学期完成教学。这样做带来的问题是,学时数紧张。学生对 C 语言还没有掌握好,经过 1 个或者 2 个学期后再学习 C++ 语言,可能 C 语言的知识又忘记了,教师还得花时间讲授 C 语言基础。有的学校只讲授 C++ 语言,学生具备面向对象的编程思想,但对面向过程的编程思想不了解,对于复

杂问题的解决存在困难。电子科技大学计算机学院进行试点,将这两门课程进行梳理,合并成一门课程,学时数比以前两门课程的总和少,但是学习效率更高。学生学习了 C 语言后立刻进行半学期考试,然后再开始 C++ 语言的学习。通过分析二者的区别与联系,可以使學生快速进入 C++ 语言的学习。学生一边学习 C++ 语言面向对象编程思想,一边将 C 语言的面向过程编程思想使用起来,可以达到较好的学习效果。

电子科技大学计算机学院通过两年的实践发现,在有教师辅导和引导的情况下,将所有的上课时间用来进行思考、讨论、编程解决问题,这大大提高了学生课堂积极性和对时间的利用率,达到了很好的学以致用效果。

二、本书特色

(1) C 语言与 C++ 语言结合起来学习,所需的学时数量减少。同一个问题在 C 语言中出现时,用的是面向过程的方法解决;在 C++ 语言中出现时,用的是面向对象的方法解决。学习者可以从中体会面向对象思想的优势,通过对比地学习达到更好的学习效果。

(2) 可以作为计算机专业学生的翻转课堂教学和 MOOC 学习者的进一步学习与补充资料。

(3) 本教材为学习者提供预习资料(预习 PPT 与预习讲解视频),课堂练习资料(课堂 PPT)以及习题库等数字资源,便于学习者学习。

(4) 本教材是互联网+教材,学习者可以通过扫描书中的二维码观看视频。编者针对初学者(计算机专业和非计算机专业),根据学习内容与进度,提供完整的问题分析、设计、编程、调试过程的视频,学习者可以跟着视频进行编程训练和学习调试技巧。

程序设计语言的学习和编程技术的掌握,都必须在大量的编程练习基础上逐渐掌握和提高,即使学习基础理论和观看教师的视频编程过程,在没有编程练习之前只能有一个大概理解,而真正的理解还要在自己做了一些编程练习之后,才能够进行深入掌握。但是如何保障编程练习的数量及品质呢?编者给学习者提供了可以练习的网址,学习者注册后可以根据里面的章节进行编程练习。此外,习题库中还有大量的习题供学习者参考。具体学习方法如下:

(1) 登录网站。登录北京大学出版社首页,单击“下载专区”导航条,在“课件下载”专区中选择“理工”(<http://www.pup.cn/dl/newsmore.cfm?sSnom=d204>),单击“《C 与 C++ 程序设计》的习题库”后即可进入习题库网站。

(2) 注册。学习者在注册的时候要选择“教材学习者的练习题:程序设计(C 与 C++)”这个班级,等待教师同意。同时,学习者可以通过 QQ 群(群号:341588984)提醒教师同意通过教材练习题的注册。

(3) 学习者在注册通过后,通过“课程中心”下面的“我的课程”,进入“教材学习者的练习题:程序设计(C 与 C++)”的作业列表,就可以看到各章作业和练习题。每章作业题量不大,难度居中;但每章练习题的题量较大,难度从简单到较难的程度都有。遇到不会的题目,学习者可以等学完下一章之后再回过头来做。学习者每次提交作业后都可以看到分数及错误提示信息,根据错误提示信息可以多次修改和提交。

此外,如果学习者在教材中发现错误或者在学习过程中遇到问题,都可以通过 QQ 群向教师反馈,或者和 QQ 群中的成员进行讨论。

三、教学安排

本书共 10 章,除了第 1 章没有预习资料,其他章均提供预习资料。每一章(除了第 1 章)都有基础理论学习(预习内容)、课堂练习题(教师在课堂引导学生深入学习的资料)。课程参考学时如下表所示。教材第 2~6 章介绍 C 语言与技术部分,第 7 章介绍 C++ 语言与 C 语言的主要区别,第 8~10 章介绍 C++ 语言与技术部分。

各章学时安排

各章内容	基础理论(预习时间)	深入与提高(课堂用时)
第 1 章 C 语言程序设计概述	0 分钟	4 学时
第 2 章 基本数据类型及运算	80 分钟	4 学时
第 3 章 控制语句	50 分钟	6 学时
第 4 章 数组和结构	68 分钟	4 学时
第 5 章 指针	64 分钟	8 学时
第 6 章 函数	112 分钟	8 学时
第 7 章 C++ 语言编程基础	47 分钟	4 学时
第 8 章 类与对象	80 分钟	8 学时
第 9 章 继承、派生与多态	93 分钟	14 学时
第 10 章 模板、命名空间和异常处理	42 分钟	4 学时
总学时	636 分钟(14 学时)	64 学时

四、教学方法

本教材可以采用下面两种教学方法。

1. 传统教学法

第 1 章的教材内容及对应的 PPT 可以作为课堂教学内容,教材习题可以作为学生的课后作业。

第 2~10 章,每一章的第一部分为基础理论,可以作为课堂的教学内容;第二部分为课堂练习题,可以作为学生的课后作业。

2. 翻转课堂教学法

第 1 章的教材内容及对应的 PPT 可以作为课堂教学内容,教材习题可以作为学生的课后作业。

第 2~10 章,每一章的第一部分为基础理论(有配套的预习 PPT 与预习讲解视频),可以作为学生每周的课前预习资料;第二部分为课堂练习题,作为课堂上教师引导学生完成的课堂作业,完成后通过学生展示的方式,共同解答并总结。课堂练习题也提供了参考答案供教师与读者参考,可以通过扫描书中的二维码获得。课堂练习题中前面的题目给出基础理论容易出错的知识点,通常以读程序、查错改错、程序填空等形式给出。在基础知识掌握得

比较好之后,教师可以在课堂上给出由浅入深的编程练习题,供学生练习。不同基础的学生可以完成不同的作业题数。

五、致谢

本教材由戴波担任主编,陈文字、丘志杰、卢光辉担任副主编。其中,戴波编写了第 1~9 章的课堂练习题及答案;陈文字编写了第 1~6 章的基础理论;丘志杰编写了第 7~10 章的基础理论;卢光辉对于全书架构提出了许多宝贵意见,并参与编写了部分章节的案例;张东祥参与编写了第 9~10 章的基础理论部分案例。2016 级研究生宋玉为本教材的整理做了不少工作,包括格式调整和图表的绘制。最后,感谢北京大学出版社孙晔副社长及编辑温丹老师为本书的出版所做的大量工作。

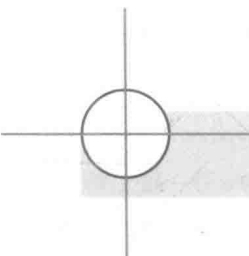
本教材的课堂练习题不少具有启发性、趣味性的 C 语言题目,采用了苏小红的《C 语言程序设计》PPT 中的案例;C++ 的面向对象概念讲解采用了金旭亮的《面向对象软件开发实践》,中国大学 MOOC 的内容;游戏思想借鉴了 Araon Reed 的《Learning XNA 4.0》。在此,对本书给予帮助的这些优秀教师及作者表示衷心感谢。

六、参考书籍

1. 陈文字,等. C 语言程序设计[M]. 成都:电子科技大学出版社,2011.
2. 沈显军,等. C++ 语言程序设计教程[M]. 北京:清华大学出版社,2010.
3. 苏小红,等. C 语言程序设计[M]. 北京:高等教出版社,2015.
4. AraonReed. *Learning XNA 4.0*[M]. O'Reilly Media,2011.



方法介绍



目 录

第1篇 C语言程序设计

第1章 C语言程序设计概述	3
1.1 引言	3
1.2 程序设计语言的发展历程	4
1.3 软件开发基本过程	6
1.4 算法的表示形式	6
1.5 C语言程序基础知识	9
1.6 C/C++语言程序的编写和运行	13
小结	20
习题	21
第2章 基本数据类型及运算	22
2.1 基本数据类型	22
2.2 常量	24
2.3 变量	28
2.4 运算符与表达式	30
2.5 混合运算与类型转换	43
2.6 数据的输入输出	44
课堂练习题	50
小结	54
习题	55
知识补充与扩展	55
第3章 控制语句	58
3.1 程序的三种基本结构	58
3.2 复合语句	59
3.3 if条件分支语句	60
3.4 switch多路开关语句	64
3.5 for循环	65
3.6 while循环和do...while循环	66
3.7 循环嵌套	68
3.8 break,continue和goto语句	69

课堂练习题	72
上机实验	73
小结	74
习题	75
知识补充与扩展	75
第 4 章 数组和结构	80
4.1 一维数组	80
4.2 二维数组	83
4.3 字符数组	87
4.4 结构及结构变量	91
4.5 结构数组	95
课堂练习题	95
上机实验	97
小结	98
习题	99
知识补充与扩展	100
第 5 章 指针	124
5.1 指针的概念和定义	124
5.2 指针运算	127
5.3 指针和数组	130
5.4 字符串指针	135
5.5 指针数组	138
课堂练习题	141
上机实验	142
小结	143
习题	144
知识补充与扩展	144
第 6 章 函数	149
6.1 函数定义和调用	153
6.2 函数参数传递	157
6.3 函数返回指针	161
6.4 递归函数	163
课堂练习题	165
上机实验	167
小结	168
习题	169
知识补充与扩展	169

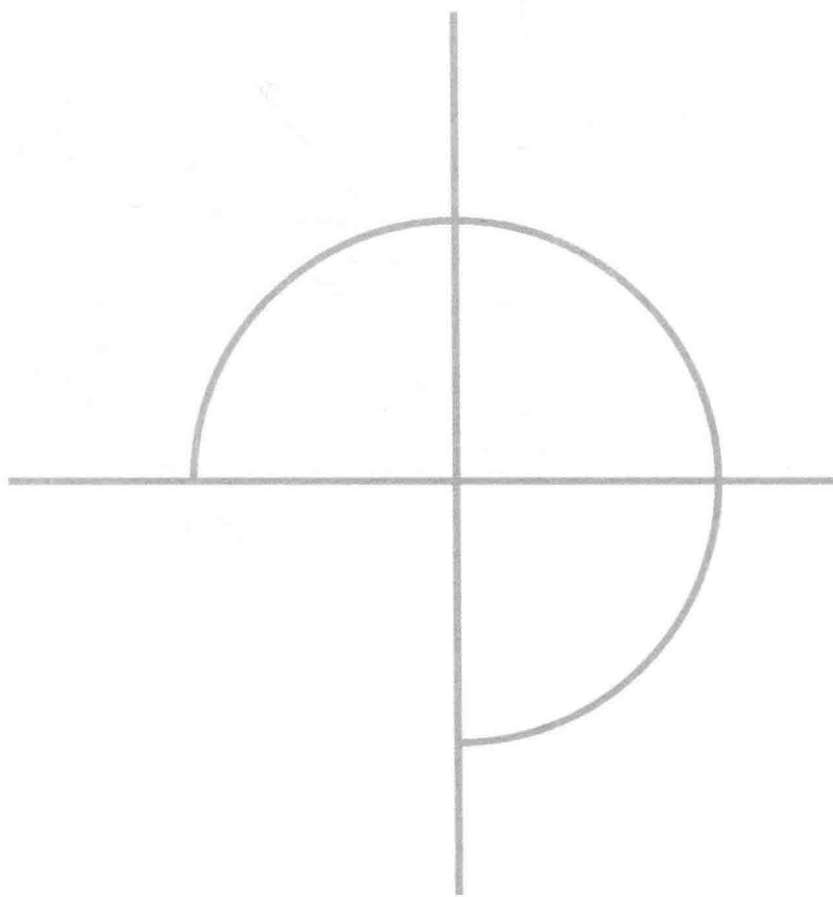
第2篇 C++ 语言程序设计

第7章 C++ 语言编程基础	191
7.1 面向对象的三个核心概念	191
7.2 C++ 语言中的 I/O	192
7.3 C++ 语言中的数据类型	193
7.4 C++ 语言中的内联函数	197
7.5 函数重载	197
7.6 带默认形参值的函数	198
7.7 C++ 语言中的动态内存分配和释放	199
课堂练习题	200
小结	202
习题	202
知识补充与扩展	203
第8章 类与对象	207
8.1 类类型的定义	207
8.2 类成员的访问控制	209
8.3 类类型的使用	211
8.4 构造函数的引入	214
8.5 析构函数的引入	215
8.6 重载构造函数的引入	217
8.7 复制构造函数的引入	219
8.8 对象数组	220
8.9 对象指针	221
8.10 this 指针	222
8.11 类类型作为参数类型的三种形式	223
8.12 静态成员	225
8.13 友元机制	228
8.14 类的组合	231
8.15 数据成员的初始化和释放顺序	231
8.16 常对象与常成员	232
课堂练习题	234
上机实验	237
小结	242
习题	243
第9章 继承、派生与多态	244
9.1 派生类的概念	244
9.2 公有继承	247

9.3 派生类的构造和析构	248
9.4 保护成员的引入	250
9.5 改造基类的成员函数	251
9.6 派生类与基类同名成员的访问方式	252
9.7 私有继承和保护继承	252
9.8 多继承	255
9.9 多态	259
9.10 运算符重载	260
9.11 赋值兼容规则	268
9.12 虚函数	271
课堂练习题	278
上机实验	280
小结	281
习题	282
第 10 章 模板、命名空间和异常处理	283
10.1 模板	283
10.2 命名空间	288
10.3 异常处理	293
课堂练习题	296
小结	296
习题	297
知识补充与扩展	297
网站推荐	304

第1篇

C语言程序设计



第1章

C语言程序设计概述

基础理论

1.1 引言

一台计算机由硬件系统和软件系统两大部分组成,硬件是物质基础,而软件是计算机的灵魂。用户、软件和硬件之间的关系如图 1-1 所示。



视频讲解

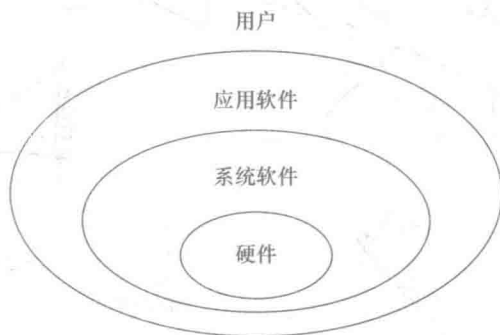


图 1-1 用户、软件和硬件之间的关系

软件是计算机系统中不可缺少的重要组成部分,根据软件的功能,软件可以分为系统软件和应用软件两大类。

(1) 系统软件。系统软件通常是指操作系统,如微软公司的 Windows、苹果公司的 Mac OS X、开源操作系统 Linux 等都是目前业界常用的操作系统。但有时候像甲骨文公司的 Oracle、微软公司的 SQL Server、开源软件 MySQL 等数据库管理软件也纳入系统软件的范畴。

(2) 应用软件。应用软件是在具体应用领域中为解决各类问题而编写的程序,由于应用领域广泛,因此应用软件的种类也很繁多。例如,办公软件有微软公司的 Office、金山公司的 WPS;即时通信软件有腾讯公司的 QQ、微信,Skype 公司的 Skype;多媒体软件有百度公司的百度影音、腾讯公司的 QQ 音乐;程序开发工具软件有微软公司的 Visual Studio、开源开发工具 Eclipse 等。

无论是系统软件还是应用软件,都是使用计算机程序设计语言来进行编写的。语言类型不同,其应用领域也有所不同。C 和 C++ 语言作为一种通用的高级程序设计语言,在系统软件和应用软件中都有着广泛的应用,也已经成为当今业界最流行的程序设计语言之一。



视频讲解

1.2 程序设计语言的发展历程

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,如图 1-2 所示。

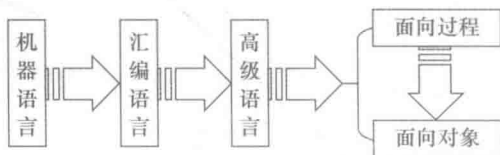


图 1-2 程序设计语言的发展历程

1. 机器语言

计算机只能够理解二进制代码,基于二进制代码所对应的程序设计语言称为机器语言。由于机器语言[如图 1-3(a)所示]可读性差,不便于阅读、编写和错误查找,而且可移植性差,因此利用机器语言进行程序设计是一项十分烦琐的工作。

2. 汇编语言

为了克服机器指令难读、难编、难记和易出错的缺点,人们使用助记符的方式[如图 1-3(b)所示,add 助记符代表加法]来表示二进制形式的机器指令。这样,人们能够较容易地读懂并理解程序,使得纠错及维护变得方便了,这种程序设计语言称为汇编语言,即第二代计算机语言。汇编语言仍然是面向机器的语言,使用起来还是比较烦琐,通用性也差。但是,用汇编语言编写的程序,其目标程序占用内存空间少、运行速度快,有着高级语言不可替代的用途。

3. 高级语言

机器语言和汇编语言都是面向实际计算机的语言,统称为低级语言。程序设计语言对机器的过分依赖,要求使用者必须对硬件结构及其工作原理都十分熟悉,这对非计算机专业人员来说是难以做到的,对于计算机的推广应用也不利。计算机事业的发展促使人们寻求与人类自然语言相接近且能为计算机所接受的通用易学的计算机语言。这种与自然语言相近并被计算机接受和执行的计算机语言称为高级语言[如图 1-3(c)所示]。

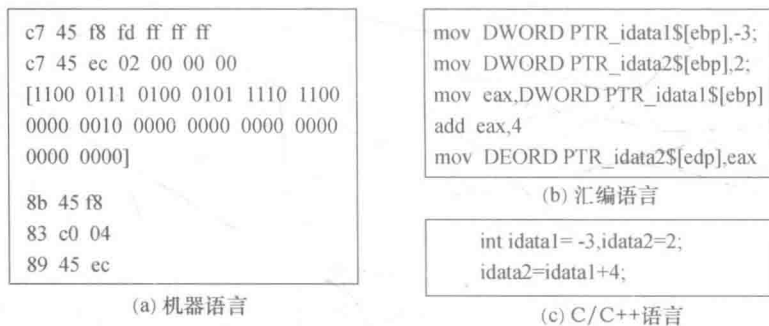


图 1-3 同一段程序的三种语言表示

高级语言是面向用户的语言,无论何种机型的计算机只要配备上相应的高级语言的编译或解释程序,则用该高级语言编写的程序就可以运行。从 1954 年第一个完全脱离机器硬

件的高级语言 FORTRAN 面世开始,六十多年来共有上千种高级语言出现。其中,影响较大、使用较普遍的有 FORTRAN、ALGOL、COBOL、BASIC、LISP、Pascal、C、PROLOG、Ada、C++、VC、VB、Delphi、Java 等。高级语言的发展经历了从早期语言到结构化程序设计语言,从面向过程到非过程化程序语言的过程。直到今天,程序设计语言还在不断地发展。

计算机并不能直接地执行使用高级语言编写的源程序,源程序在输入计算机时,通过“翻译程序”翻译成机器语言形式的目标程序之后,计算机才能识别和执行。这种“翻译”通常有两种方式,即编译方式和解释方式。

编译方式是指将程序的源代码“翻译”成目标代码(机器指令),目标代码可以脱离其语言环境独立执行,效率较高。但源代码一旦被修改,需要再重新编译生成新的目标文件(*.obj)才能执行,C 和 C++ 就属于这一类。编译过程如图 1-4 所示。

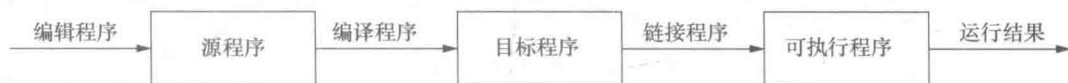


图 1-4 编译过程

解释方式是应用程序的源代码一边由相应语言的解释器“翻译”成目标代码(机器语言),一边执行,因此效率比较低。而且应用程序不能生成可独立执行的可执行文件,故不能脱离其解释器;但解释方式比较灵活,可以动态地调整、修改应用程序。解释过程如图 1-5 所示。

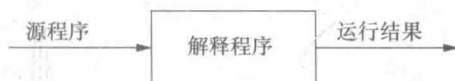


图 1-5 解释过程

目前,常用的程序设计语言有数百种之多,尽管任何一种程序设计语言均可作为编程工具完成编码、实现功能,但是由于每一种程序设计语言都有其自身的特点,它们对问题的处理及解决方式却不尽相同。选择程序设计语言时,应根据任务及语言的特点进行综合考虑。表 1-1 列举了一些常见的高级程序设计语言及其相应的应用领域。

表 1-1 程序设计语言的应用领域

语 言	应用领域
C 语言	操作系统(Linux)、驱动程序、Android
C++	电信级应用、网络应用(ACE)、浏览器(WebKit)
Object C	iPhone、iPod(兼容 C++)
swift	iPhone、iPod(iOS 上新的编程语言)
C#	Window 平台
ActionScript	Flash 游戏
HTML/JavaScript	Web 应用(如网游)
python	分布式应用程序