

高等学校计算机应用规划教材

C 语言程序设计

(第 3 版)(微课版)

王先水 杜丽芳 刘艳 编著

清华大学出版社

北 京

内 容 简 介

本书采用 Visual C++ 6.0 的编译环境进行开发。“以学生动手能力为基础,以运用知识解决问题为突破口,以基础知识+上机实训+项目实训模式组织教学,以培养应用型技术人才为目标”的理念组织教材编写。

全书共 11 章。第 1 章介绍 C 语言的发展及特点, C 语言程序的基本结构, C 语言的字符集、标识符和关键字, 使用 Visual C++ 6.0 开发 C 语言程序的过程; 第 2 章介绍 C 语言基本数据类型、常量和变量、运算符和表达式, 不同数据类型的转换; 第 3 章介绍输入/输出函数、算法和顺序结构程序设计基本方法; 第 4 章介绍选择结构程序设计基本方法; 第 5 章介绍循环结构程序设计基本方法; 第 6~8 章介绍数组、函数和指针的基本概念、基本原理和基本应用, 通过这 3 章的学习学生能灵活运用函数、数组和指针编写程序, 能实现科学计算和实际工程设计中的常用问题。第 9 章介绍结构体与共用体及链表的基本概念、结构体数组、结构体指针的使用方法; 第 10 章介绍文件的基本操作方法; 第 11 章介绍 ATM 机自动取款系统开发的思路, 按照“需求分析—系统架构—总体设计—详细设计—代码设计—程序运行—系统测试”的过程进行系统、完整、详细的讲解, 还对“学生成绩管理系统、电话簿管理系统”综合实训进行了需求分析、系统架构、总体设计、详细设计的提示分析, 其中代码设计、程序运行和系统测试留给学生作为课程设计去完成。

本书可作为普通高等学校计算机专业及相关专业教材, 也适合作为高等职业院校教材, 还可作为程序开发人员和编程爱好者自学的参考用书, 以及全国计算机等级考试的辅导用书。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计: 微课版 / 王先水, 杜丽芳, 刘艳 编著. —3 版. —北京: 清华大学出版社, 2020.8
高等学校计算机应用规划教材
ISBN 978-7-302-55054-9

I. ①C… II. ①王… ②杜… ③刘… 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(2020)第 051689 号

责任编辑: 刘金喜

封面设计: 高娟妮

版式设计: 孔祥峰

责任校对: 马遥遥

责任印制: 丛怀宇

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 24 字 数: 614 千字

版 次: 2011 年 8 月第 1 版 2020 年 8 月第 3 版 印 次: 2020 年 8 月第 1 次印刷

定 价: 68.00 元

产品编号: 086353-01

前 言

人们应用计算机技术解决实际问题已日益普遍，计算机的应用已渗透到人们工作和生活的各个领域。在高等教育逐步实现大众化的过程中，越来越多的高校面向市场，为行业、企业培养各级各类高级实用型人才。在高等学校应用技术型人才培养模式、培养目标、教学内容和课程体系框架下，由长期从事 C 语言教学的老师精心组织编写了本书。

1. 主要特色

1) 组织合理

本书从 C 语言程序设计的基本原理及程序设计的基本思想出发，紧扣基础，面向应用，循序渐进引导学生学习程序设计的思想和方法；在编写过程中突出“三基”（基本概念、基本原理、基本应用）的讲解与应用；对例题的算法分析做了详细讲解；给出例题在 Visual C++ 6.0 编译环境下的运行结果。

在课程内容选择上，遵循学生能力培养的基本规律，采用“观察—联想—变换”的思想构建教学内容，使 C 语言程序设计知识、软件工程基础知识、数据结构基本知识融入综合项目实训中。

2) 理念先进

本书以 C99 标准，采用 Visual C++ 6.0 为编译器的编译环境进行开发。“以学生动手能力为基础，以运用知识解决问题为突破口，以基础知识+上机实训+项目实训模式组织教学，以培养应用型技术人才为目标”的理念组织教材编写。

3) 通俗易懂

本书为编程语言的入门教材，侧重于以语法为基础、以算法为“灵魂”、以编程为创新、以编码为规范培养学生良好的编程习惯和软件开发能力。例题编排上由浅入深、逐层递进。在第 1~5 章基础上讲解的例题，到第 7、8 章又运用新的方法进行了详细讲解，使学生对知识形成系统性。

2. 内容安排

全书共分为 11 章。第 1~10 章介绍 C 语言程序设计基础知识和上机实训；第 11 章为综合实训；最后有 3 个附录。其具体内容如下。

第 1 章介绍 C 语言的发展及特点，C 语言程序的基本结构，C 语言的字符集、标识符和关键字，使用 Visual C++ 6.0 开发 C 语言程序的过程。

第 2 章介绍 C 语言基本数据类型、常量和变量、运算符和表达式，不同数据类型的转换。

第 3 章介绍输入/输出函数、算法和顺序结构程序设计基本方法。

第4章介绍选择结构程序设计基本方法。

第5章介绍循环结构程序设计基本方法。

第6~8章介绍数组、函数和指针的基本概念、基本原理和基本应用,通过这3章的学习,学生能灵活运用函数、数组和指针编写程序,能实现科学计算和实际工程设计中的常用问题。

第9章介绍结构体与共用体及链表的基本概念、结构体数组、结构体指针的使用方法。

第10章介绍文件的基本操作方法。

第11章介绍ATM机自动取款系统开发的思路,按照“需求分析—系统架构—总体设计—详细设计—代码设计—程序运行—系统测试”的过程进行系统、完整、详细的讲解,还对“学生成绩管理系统、电话簿管理系统”综合实训进行了需求分析、系统架构、总体设计、详细设计的提示分析,其中代码设计、程序运行和系统测试留给学生作为课程设计去完成。

3. 读者对象

本书可作为普通高等学校计算机专业及相关专业教材,也适合作为高等职业院校教材,还可作为程序开发人员和编程爱好者自学的参考用书,以及全国计算机等级考试的辅导用书。

本书在编写过程中得到了武汉工程科技学院的大力支持,全书在武汉工程科技学院信息工程学院院长张友纯的指导下,由计算机科学与技术教研室王先水、杜丽芳、刘艳老师共同编写完成。

本书在编写过程中,参考了C语言程序设计的相关书籍及杂志等资料,引用了相关教材的部分内容,吸取了同行的宝贵经验,在此谨表谢意。由于编者水平有限,书中难免有不妥和疏漏之处,敬请各位读者提出宝贵意见和建议。

本书教学资源可通过扫描下方二维码获取。

服务邮箱:476371891@qq.com。



教学资源下载

编者

2019年10月10日于武汉

目 录

第 1 章 C 语言概述	1
1.1 程序及程序设计语言	1
1.1.1 程序及程序设计	1
1.1.2 程序设计语言	2
1.2 C语言的发展及特点	3
1.2.1 C语言的发展概述	3
1.2.2 C语言的特点	4
1.3 C语言程序的基本结构	5
1.4 C语言字符集、标识符和关键字	9
1.4.1 C语言字符集	9
1.4.2 C语言标识符	10
1.4.3 C语言关键字	11
1.5 C语言程序的开发环境	12
1.5.1 Visual C++ 6.0集成开发环境介绍	12
1.5.2 在Visual C++ 6.0环境下建立和 运行C语言程序的步骤	13
本章小结	16
易错提示	17
习题1	17
第 2 章 C 语言数据类型和表达式	20
2.1 C语言的数据类型	20
2.1.1 整型数据	21
2.1.2 实型数据	23
2.1.3 字符型数据	24
2.2 常量和变量	27
2.2.1 常量	27
2.2.2 变量	29
2.3 运算符和表达式	32
2.3.1 算术运算符和算术表达式	32

2.3.2 赋值运算符与赋值表达式	33
2.3.3 自增自减运算符和自增自减表达式	34
2.3.4 逗号运算符和逗号表达式	35
2.3.5 条件运算符和条件表达式	36
2.4 数据类型转换	37
2.4.1 自动转换	38
2.4.2 赋值转换	38
2.4.3 强制转换	40
2.5 位运算	40
2.5.1 位运算概述	41
2.5.2 按位取反运算	41
2.5.3 移位运算	41
2.5.4 按位与、或和异或	42
本章小结	43
易错提示	44
习题2	44
第 3 章 顺序结构程序设计	49
3.1 程序设计的基本概念	49
3.2 C语言的语句	51
3.3 格式化输入/输出函数	52
3.3.1 格式化输出函数printf()	53
3.3.2 格式化输入函数	56
3.4 字符输入/输出函数	59
3.5 程序设计举例	61
本章小结	64
易错提示	65
习题3	65
第 4 章 选择结构程序设计	71
4.1 关系运算符与关系表达式	71

4.1.1 关系运算符	71	6.2.1 二维数组的定义和注意事项	149
4.1.2 关系表达式	72	6.2.2 二维数组的引用	151
4.2 逻辑运算符与逻辑表达式	72	6.2.3 二维数组的赋值	152
4.2.1 逻辑运算符	72	6.2.4 二维数组的应用	153
4.2.2 逻辑表达式	74	6.3 字符数组	158
4.3 if语句	74	6.3.1 字符数组的定义	158
4.3.1 单分支if语句	75	6.3.2 字符数组初始化	159
4.3.2 双分支if语句	77	6.3.3 字符数组输入输出	161
4.3.3 多分支if语句	78	6.3.4 字符串处理函数	163
4.3.4 if语句的嵌套	81	6.4 程序举例	168
4.4 switch语句	83	本章小结	173
4.5 程序设计举例	86	易错提示	173
本章小结	89	习题6	174
易错提示	89	第7章 函数与编译预处理	185
习题4	90	7.1 模块化程序设计	186
第5章 循环结构程序设计	98	7.2 定义函数	188
5.1 while语句	99	7.2.1 标准库函数	188
5.2 do-while语句	103	7.2.2 函数的定义	189
5.3 for语句	104	7.3 函数的调用	191
5.4 for语句与while语句和do-while 语句比较	107	7.3.1 函数调用形式	191
5.5 break语句和continue语句	108	7.3.2 函数调用方式	192
5.5.1 break语句	108	7.3.3 函数调用过程	195
5.5.2 continue语句	109	7.3.4 函数调用结果的返回	196
5.5.3 break语句和continue语句的区别	111	7.4 函数间数据传递	198
5.6 循环的嵌套结构	112	7.4.1 普通变量作为实参的值传递	199
5.6.1 双重循环的嵌套	112	7.4.2 数组名作为实参的地址传递	201
5.6.2 多重循环的嵌套	114	7.4.3 字符串作为实参的传递	202
5.7 程序设计举例	115	7.5 函数的嵌套调用	203
本章小结	123	7.6 递归函数与递归调用	206
习题5	123	7.7 变量作用域与存储方式	213
第6章 数组	134	7.7.1 变量作用域	213
6.1 一维数组	136	7.7.2 变量的存储方式	217
6.1.1 一维数组的定义	136	7.8 编译预处理	220
6.1.2 一维数组的引用	138	7.8.1 宏定义	220
6.1.3 一维数组赋初值	139	7.8.2 文件包含	223
6.1.4 一维数组的应用	141	7.8.3 条件编译	225
6.2 二维数组	149	本章小结	227
		习题7	228

第 8 章 指针	237		
8.1 指针与指针变量.....	237		
8.1.1 指针的概念.....	237		
8.1.2 指针变量.....	239		
8.1.3 指针变量的定义.....	239		
8.1.4 指针变量初始化.....	240		
8.1.5 指针运算符.....	241		
8.1.6 指针运算.....	243		
8.1.7 多级指针.....	247		
8.2 指针与数组.....	247		
8.2.1 一维数组元素的指针访问.....	248		
8.2.2 二维数组元素的指针访问.....	253		
8.2.3 指向一维数组的指针.....	258		
8.2.4 指针数组.....	261		
8.3 字符指针与字符串.....	264		
8.3.1 字符串的表现形式.....	264		
8.3.2 用字符指针处理字符串.....	266		
8.4 指针与函数.....	269		
8.4.1 指针作为函数参数.....	269		
8.4.2 指向函数的指针.....	270		
8.4.3 返回指针值的函数.....	272		
8.4.4 带参数的main()函数.....	273		
8.5 动态指针.....	274		
8.6 指针程序设计举例.....	276		
本章小结.....	279		
易错提示.....	280		
习题8.....	281		
第 9 章 结构体与共用体	290		
9.1 结构体的概念.....	290		
9.1.1 结构体类型的定义.....	290		
9.1.2 结构体类型变量的定义.....	291		
9.1.3 结构体类型变量的引用.....	295		
9.1.4 结构体类型变量的初始化.....	298		
9.2 结构体数组与链表.....	298		
9.2.1 结构体数组的定义与引用.....	298		
9.2.2 结构体数组初始化和应用.....	300		
9.2.3 链表.....	300		
9.3 共用体的概念.....	307		
9.3.1 共用体类型的定义.....	307		
9.3.2 共用体类型变量的定义.....	308		
9.3.3 共用体类型变量的引用.....	309		
9.4 程序设计举例.....	311		
本章小结.....	316		
习题9.....	316		
第 10 章 文件	324		
10.1 文件的概述.....	324		
10.1.1 文件的分类.....	324		
10.1.2 文件的缓冲区.....	325		
10.1.3 文件的存取方式.....	326		
10.1.4 文件类型的指针.....	326		
10.2 文件的常用操作.....	327		
10.2.1 文件的打开与关闭.....	327		
10.2.2 文件的读写.....	329		
10.2.3 文件的定位.....	335		
10.2.4 文件的其他操作.....	337		
本章小结.....	338		
习题10.....	338		
第 11 章 综合实训	341		
11.1 简单的银行自动取款机系统.....	341		
11.1.1 问题描述.....	341		
11.1.2 总体设计.....	341		
11.1.3 详细设计.....	342		
11.1.4 设计代码.....	349		
11.1.5 系统运行界面.....	354		
11.1.6 系统测试.....	357		
11.2 学生成绩管理系统.....	357		
11.2.1 设计要求.....	357		
11.2.2 设计架构.....	357		
11.2.3 设计方法.....	359		
11.2.4 代码设计.....	362		
11.3 电话簿管理系统.....	362		
11.3.1 设计要求.....	362		
11.3.2 设计架构.....	362		
11.3.3 设计方法.....	363		
11.3.4 代码设计.....	365		
11.4 综合实训题目.....	365		

附录 I 常用字符与 ASCII 码对照表	367	附录 III 常用 C 语言库函数	369
附录 II C 语言运算符的优先级和 结合方向	368	参考文献	373

C语言概述

【学习目标】

1. 了解程序设计语言的发展和 C 语言的特点。
2. 掌握 C 语言程序的基本结构。
3. 能正确运用 C 语言的标识符及关键字。
4. 能熟练运用 Visual C++ 集成开发环境创建、编译、连接和运行 C 语言源程序。

自 1946 年第一台计算机问世以来, 计算机学科的发展逐步引起人们的高度关注, 计算机学科的应用越来越广泛。人们使用计算机管理大量的数据, 处理纷繁复杂的办公事务, 使用计算机完成复杂的科学计算, 加快科学研究的进程, 使用计算机实现网络通信, 拉近人们的空间距离。计算机本身是无生命的机器, 要使计算机能为人类完成各种各样的工作, 就必须让它执行预先编写好并存储于计算机内存的程序, 这些程序是依靠程序设计语言编写出来的。但在众多的程序设计语言中, C 语言有其独特之处, 既具备低级语言即汇编语言的特点, 也具有直接操作计算机硬件的功能, 还是目前盛行的嵌入式系统中应用的语言之一。C 语言不仅能编写操作系统软件, 还能编写应用软件, 是一种高级语言, 学习起来很容易, 也是众多高级语言学习的基础语言。如果我们认真学习本书, 认真思考本书介绍的知识, 并在本书的指导下认真上机实践, 将会很快掌握使用 C 语言编写程序的方法, 并逐渐领悟到 C 语言的精妙所在。你想编写出人们喜爱的实用程序吗? 你想成为一个优秀的程序设计员吗? 那就让我们一起走进 C 语言的世界吧!

1.1 程序及程序设计语言

我们都知道, 计算机完成的任何工作, 都是计算机运行程序的结果, 而计算机运行的程序又都是使用计算机语言, 即程序设计语言来编写的。自从计算机诞生以来, 程序设计语言已经经历了机器语言、汇编语言和高级语言 3 个主要发展阶段。

1.1.1 程序及程序设计

要让计算机按人们的意图处理问题, 人们必须事先确定解决问题的策略, 也就是要设计好

算法,然后再用计算机语言来描述设计好的算法。用计算机语言描述的用于解决某一具体问题的符号序列(代码序列)就是程序。

一个计算机程序主要反映两方面问题:一是描述问题的对象及它们之间的关系;二是描述对这些对象进行处理的规则。对象及它们之间的关系属于数据结构的内容,处理是求解某个问题的算法。因此程序的描述常用下列式子:

$$\text{程序}=\text{数据结构}+\text{算法}$$

程序设计是根据计算机要完成的任务,提出相应的需求,在此基础上设计数据结构和算法,然后再编写相应的程序代码并测试该代码运行的正确性,直到能够得到正确的运行结果为止。程序设计是人们借助计算机语言,告诉计算机要做什么(即要处理哪些数据),如何处理(即按什么步骤来处理)。程序设计非常讲究方法,良好的设计方法能够大大提高程序的高效性、合理性,程序设计有一套完整的方法,这一套完整的方法也称为程序设计方法学。程序设计方法学在大型软件设计中是非常重要的,同时也是软件工程的组成部分之一。

算法是为完成一项任务所应当遵循的逐步的、规则的、精确的、无歧义的描述,其总步数是有限的。简而言之,算法是解决一个问题采取的方法和步骤的描述。算法应体现有穷性、确定性、有零个或多个输入、有一个或多个输出和有效性的特点。

在进行程序设计时,由于人们的思维方式不同,所设计的程序有所不同,不同的程序在执行时其效率也是不同的,影响程序执行效率的因素主要有时间效率和空间效率。高效程序的设计基于良好的信息组织(保证占尽量少的内存单元)和优秀算法(保证使用尽量少的的时间)。

1.1.2 程序设计语言

程序设计语言的发展经历了机器语言、汇编语言和高级语言 3 个主要阶段。了解程序设计语言的发展,更加有助于读者加深对程序设计语言的认识,能更好地运用程序设计语言来解决一些实际问题。

1. 机器语言

机器语言是人们最早使用的程序设计语言。由于计算机硬件只能识别和处理 0 和 1 所组成的代码,因此机器语言是 0 和 1 两个代码组成的机器指令序列,控制硬件完成指定的操作。

例如,以下是某计算机的两条机器指令。

加法指令: 10000000

减法指令: 10010000

用机器语言编写的程序能够被计算机直接识别并执行,程序的执行效率特别高,这是机器语言的最大优点。但机器语言与人们习惯使用的自然语言相差太大,难读、难记、难写、难修改,用它来编写程序很不方便,并且硬件设备不同的计算机的机器语言也有差别,编写的程序缺乏通用性。编写机器语言程序时,要求程序员必须相当熟悉计算机的硬件结构,因而现在人们一般不使用机器语言编写程序来解决一些实际应用问题。

2. 汇编语言

20 世纪 50 年代中期,为了减轻人们使用机器语言编程的负担,开始使用一些助记符号来表示机器语言中的机器指令,于是便产生了汇编语言。助记符号采用代表某种操作的英文单词

的缩写。例如，上例中的两条指令用汇编指令描述如下。

```
ADD A, B(其中, A、B 表示的是两个操作数)
```

```
SUB A, B
```

上述两条汇编指令计算机不能直接执行，它必须经过一个汇编程序的系统软件翻译成机器指令后才能执行。用汇编语言编写的程序称为汇编语言源程序，将汇编语言源程序翻译成机器能执行的程序称为汇编程序。

汇编语言指令和机器语言指令之间具有一一对应的关系，因此不同计算机的汇编语言也不尽相同，并且程序编写时仍需要对计算机的内部结构比较熟悉，但相对于机器语言就简单多了。

早期的操作系统软件主要是用汇编语言编写的，汇编语言和机器语言一样，对不同的计算机硬件设备，需要使用不同的汇编语言指令，因此汇编语言程序不利于在不同计算机系统之间移植。所以，现在的汇编语言一般在专业程序设计员中使用，主要用于控制系统、病毒的分析与防治、设备驱动程序的编写。

3. 高级语言

汇编语言和机器语言是面向机器的，它们属于低级语言的范畴。人们在使用它们设计程序时，要求对机器比较熟悉。为了克服低级语言的缺点，将程序设计的重点放在解决问题(即算法)方面，于是产生了面向过程和面向对象的高级语言，如 C、C++、Visual C++、Java、C#等语言。由于高级语言是面向过程或面向对象的计算机语言，所以它们在描述上非常接近于人们习惯使用的自然语言，并且它们不受计算机硬件结构的制约，具有良好的移植性。因此，用高级语言编写的程序可以适用于不同硬件设备的计算机，这给人们用计算机语言编程来解决实际应用问题带来了极大的方便。

1.2 C 语言的发展及特点

C 语言是目前国际上广泛流行的一种基础的结构化的程序设计语言，它简洁、紧凑、使用灵活。用 C 语言编写的程序执行效率高，可移植性好，不做修改或稍加修改就能用于各种型号的计算机和各种操作系统。

C 语言不仅能开发系统软件和应用软件，而且还能实现汇编语言的大部分功能。近几年 C 语言更成了嵌入式系统开发的首选语言，深受广大程序设计者的喜爱。

1.2.1 C 语言的发展概述

C 语言是在 20 世纪 70 年代初由美国贝尔实验室的 Dennis M. Ritchie 设计的，并首先安装在 UNIX 操作系统的 DEC PDP-11 计算机上实现，因而最初的 C 语言是为了描述和实现 UNIX 操作系统而设计的。

到了 1973 年，K.Thompson 和 Dennis M. Ritchie 两人合作用 C 语言将 UNIX 中 90%以上的内容进行了改写(即 UNIX 第 5 版)。后来，人们对 C 语言进行了多次的改进，其主要还是在贝尔实验室内部使用，直到 1975 年 UNIX 第 6 版公布后，C 语言的突出优点才引起人们的广泛关注。

1978年,由美国(AT&T)贝尔实验室正式发表了C语言。同时,由B.W.Kernighan和D.M.Ritchie合著了著名的*The C Programming Language*一书,简称为*K&R*,也称为*K&R*标准。但是,在*K&R*中并没有定义一个完整的标准C语言,后来由美国国家标准协会(American national standards institute, ANSI)在此基础上制定了一个C语言标准,于1983年发表,称为ANSI C。

ANSI C标准于1989年被采用,该标准一般称为ANSI/ISO Standard C,于是1989年定义的C标准定义为C89。其中详细说明了使用C语言书写程序的形式,规范对这些程序的解释,包括C语言的表示法、C语言的语法和约束、解释C程序的语法规则、C程序输入和输出的表示,一份标准实现了限定和约束。

到了1995年,出现了C语言的修订版,增加了库函数,形成了初步的C++语言,C89便成了C++语言的子集。由于C语言的不断发展,在1999年又推出了C99,C99在保留C语言特性的基础上增加了一系列新的特性,形成了C99标准。

目前,在微机上比较流行的C语言版本主要有Microsoft C/C++、Turbo C、Quick C、Visual C/C++。

这些C语言版本不仅实现了ANSI C标准,而且在此基础上各自做了一些扩充,使之更加方便、完美。

本书以Visual C++ 6.0为集成开发环境,在C99标准和Visual C++ 6.0环境下对C语言做介绍。Visual C++ 6.0是Microsoft公司推出的在Windows环境下,进行应用程序开发的可视化与面向对象程序设计软件开发工具。

1.2.2 C语言的特点

C语言是工科院校学生必修的一门计算机语言。掌握C语言已经成为计算机开发人员的一项基本功。C语言之所以能被世界计算机界广泛接受,正是由于它自身具备的突出特点。从语言体系和结构上讲,它与Pascal等语言相类似,是结构化程序设计语言。但从用户应用、实现难易程度、程序设计风格等角度来看,C语言的特点又是多方面的。

1) C语言的优点

(1) 适应性强,应用范围广。它能适应从8位微型机到巨型机的所有机种,可用于系统软件到涉及各个领域的应用软件。

(2) 语言本身简洁,使用灵活,便于学习和应用。在源程序表示方法上,与其他语言相比,一般功能上等价的语句,C语言的书写形式更为直观、精炼。

(3) 语言的表达能力强。C语言是面向结构化程序设计的语言,通用直观;运算符达34种,涉及的范围广、功能强。其既可直接处理字符、访问内存物理地址、进行位操作,也可直接对计算机硬件进行操作,反映了计算机的自身性能,足以取代汇编语言来编写各种系统软件和应用软件。鉴于C语言兼有高级语言和汇编语言的特点,也可称其为“中级语言”。

(4) 数据结构类型丰富。C语言具有现代化语言的各种数据结构,且具有数据类型的构造能力,因此,便于实现各种复杂的数据结构的运算。

(5) 程序设计结构化。C语言是一种结构化语言,层次清晰,具有顺序、选择、循环3种程序控制结构,易于调试和维护,并以函数作为主要结构成分,便于程序模块化,符合现代程

序设计风格。

(6) 运行程序质量高, 程序运行效率高。试验表明, C 语言源程序生成的运行程序的效率仅比汇编程序的效率低 10%~20%, 但 C 语言编程速度快, 程序可读性好, 易于调试、修改和移植, 这些优点是汇编语言所无法比拟的。

(7) 可移植性好(与汇编语言相比)。C 语言可以方便地在不同操作系统平台之间转换使用。统计资料表明, C 语言编译程序 80%以上的代码是公共的, 因此稍加修改就能移植到各种不同型号的计算机上。

2) C 语言的缺点

C 语言存在的不足之处是: 运算符和运算优先级过多, 不便于记忆; 语法定义不严格, 编程自由度大, 编译程序查错纠错能力有限, 给不熟练的程序员带来一定困难; C 语言的理论研究及标准化工作也有待推进和完善。因此, C 语言对程序设计人员的素质要求相对要高。

综上所述, C 语言把高级语言的基本结构与低级语言的高效实用性很好地结合起来, 不失为一个出色而有效的现代通用程序设计语言。它一方面在计算机程序语言研究方面具有一定价值, 由它引出了许多后继语言; 另一方面, C 语言对整个计算机工业和应用的发展都起了很重要的推动作用。正因为如此, C 语言的设计者获得了世界计算机科学技术界的最高奖——图灵奖。

1.3 C 语言程序的基本结构

为了更进一步理解 C 语言程序结构的特点, 下面通过几个 C 程序来进行说明。这几个程序由简到难, 表现了 C 语言源程序在组成结构上的特点。虽然其有关内容还没有介绍, 但从这几个例子中了解到组成一个 C 语言程序的基本框架和书写格式要求。

【例题 1.1】 在显示器上输出: “The university welcomes you!”

算法分析: 在主函数中用输出函数 printf()原样输出以上文字。

程序代码:



```
#include <stdio.h>           //编译预处理命令
int main()                   //定义主函数
{                             //函数开始标志
    printf("The university welcomes you! \n"); //输出字符串信息
    return 0;                 //函数执行完返回函数值 0
}                             //函数结束标志
```

#include 称为文件包含命令, 扩展名为.h 的文件称为头文件。其作用是将声明输入/输出函数所在的 stdio.h 包含到本程序文件中。

int main()中的 main 是函数名且是 C 程序的主函数名, 每一个 C 程序必须有且只能有一个 main()函数, main()函数是 C 语言编译系统使用的专用名字。int 是 main()函数的返回值类型——整型(C99 标准)。

main()函数后面用花括号({})括起来的部分是函数体即程序实现的功能。该函数的功能是在显示器上打印一串字符 “The university welcomes you!”。

printf("The university welcomes you! \n");是 C 程序的输出函数,该函数的执行结果是将英文双引号内的字符串送到显示器上显示出来,且\n在此是输出一个不显示的换行符(转义字符)。

printf()函数是由 C 编译系统定义的标准函数,用户在程序中直接调用,但调用前必须在程序的开头用包含命令 include 将 stdio.h 输入/输出头文件包含到程序中,否则在编译时系统会出现报错信息。

return 0 是 C99 标准,是与 int main()对应的,表明函数执行完后返回一个整型数值。但 C89 标准的 C 程序结构中没有此语句,是因为 main()函数返回的是一个空值,即 void main()。

分号(;)是 C 语言语句的结束标志。

// 是 C 语言中的注释。在 Visual C++ 6.0 的开发环境中有两种注释方法:一种是“//”称为行注释;另一种是“/* */”称为块注释。注释的作用是帮助他人理解阅读程序。读者在学习编程时应养成添加注释的习惯。

程序运行结果如图 1.1 所示。



图 1.1 例题 1.1 程序运行结果

如果我们编写的程序只能在显示器中原样打印出字符串,那就没有什么实际意义。编写程序是要解决数据的计算或比较复杂的问题的求解,并将结果打印在屏幕上。下面介绍求两个整数加法的程序。

【例题 1.2】编写程序:计算两个指定整数的和并将结果打印在屏幕上。

算法分析:设置 3 个变量,value1、value2 用来存放两个整数,sum 用来存放这两个数的和。

程序代码:



```
# include <stdio.h>
int main()
{
int value1,value2,sum;//定义三个整型数据变量 value1,value2,sum
value1=5;//给变量 value1 赋初值 5
value2=2;//给变量 value2 赋初值 2
sum=value1+value2;//计算 value1 和 value2 两个变量的和存在变量 sum 中
printf("The sum is%d\n",sum);//输出两数的和
return 0;
}
```

本题同例题 1.1 相比,有以下不同之处。

- (1) 定义三个整型数据符号(称为变量)value1、value2 和 sum。
- (2) 分别对数据符号(变量)value1、value2 赋给整型数值 5、2。
- (3) 将符号 value1 和符号 value2 相加得到的值保存在符号 sum 中。

printf()函数英文双引号内的%d 位置上输出一个具体的整型数值,其值是逗号后的变量 sum 的值。其中,%是格式符号,d 是整型数值。

程序运行结果如图 1.2 所示。



图 1.2 例题 1.2 程序运行结果

思考：

(1) 若将 `printf("The sum is %d\n",sum)` 中的 `sum` 换成 10，则程序输出的结果是什么？请同学们上机调试一下。

(2) 若要输出算术算式，则如何修改 `printf()` 函数英文双引号内的表现形式。

(3) 例题 1.2 实现了两个给定数值的加法运算，若要实现任意两个整型数据的加法运算，则程序需做哪些方面的修改？请看例题 1.3。

【例题 1.3】编写程序：要求计算任意两个数的和并以算术形式输出。

算法分析：如何将一个动态的值赋给变量呢？这就需要调用输入函数 `scanf()`。

程序代码：



```
# include <stdio.h>
int main()
{
int value1,value2,sum;
printf("请通过键盘输入两个整数：\n"); //程序运行时在屏幕上打印提示信息
scanf("%d%d",&value1,&value2); //通过键盘输入两个任意的整型数值
sum=value1+value2; //计算两数的和
printf("%d+%d=%d\n",value1,value2,sum); //以算术形式输出
return 0;
}
```

本题同例题 1.1、例题 1.2 相比，有以下不同之处。

(1) `scanf()` 函数的功能是从键盘缓冲区中提取两个整型数据，放到变量 `value1`、`value2` 所在的内存单元中。`scanf` 是输入函数，当程序从上往下运行到 `scanf` 输入语句时，会等待我们通过键盘往键盘缓冲区输入数据，输入完成按 `Enter` 键后开始提取数据到指定地址的内存单元。

(2) `scanf("%d%d",&value1,&value2)` 语句中符号 `&` 是地址运算符，通过变量 `value1`、`value2` 的地址找到内存单元，并将从键盘缓冲区中提取到的两个整数放到相应内存单元中。

(3) `printf("%d+%d=%d\n",value1,value2,sum)` 语句中英文双引号内的 `%d` 位置上输出具体的值并分别对应其逗号后 3 个变量 `value1`、`value2`、`sum` 的值。其中的“+”和“=”按原样形式输出。

`printf()` 和 `scanf()` 两个函数分别称为格式输出函数和格式输入函数，其意义是按指定的格式输出输入值，具体使用方法在后续章节中进行学习。

在使用 `scanf()` 和 `printf()` 两个标准函数时，一定要将 `stdio.h` 头文件包含到 C 源程序中。

程序运行：

示例 1: 12

33

结果 1: 12+23=35

示例 2: 12 23

结果 2: 12+23=35

思考: scanf("%d%d",&value1,&value2)语句与 scanf("%d,%d",&value1,&value2)语句有没有区别?若有,则体现在输入两数时如何操作?请同学们上机调试一下。

【例题 1.4】编写程序:要求实现输入任意的两个整数输出其中的较大数。

程序分析:用一个函数来实现求两个整数中的较大者。在主函数中调用此函数并输出结果。

程序代码:



```
# include <stdio.h>
int main()                                /*主函数*/
{
int max(inta,int b);                      /* 对被调函数的 max 的声明*/
int value1,value2,largenumber;           /* 定义 3 个变量 */
printf("请通过键盘输入两个整数: \n"); /* 提示输入两个整数 */
scanf("%d%d",&value1,&value2);          /* 输入 value1,value2 变量的值 */
largenumber=max(value1,value2);          /* 调用 max()函数,得到的值赋给 largenumber*/
printf("largenumber=%d\n",largenumber);
return 0;
}
int max(inta,int b)                        /*定义 max()函数,其值为整型,形式参数 a、b 也是整型*/
{
int c;                                    /* max()函数中声明部分,定义本函数中用到的变量 c */
if(a>b) c=a;
else c=b;
return(c);                                /* 将 c 的值返回,通过 max 带回到调用函数的位置 */
}
```

本题同上面几个例子相比,有以下不同之处。

(1) C 语言源程序中包含两个函数,主函数 main()和被调用函数 max(),其中 max()函数是用户自定义的函数。

(2) max()函数的功能是实现 a 和 b 两数的比较且将较大数赋给变量 c,并通过 return 语句将 c 的值返回给主调函数 main()。

(3) max()函数放在主调函数 main()之后,则要求在主调函数 main()中进行声明,其作用是使编译系统能够正确识别和调用 max()函数;max()函数放在主调函数 main()之前,则主调函数 main()中可以不作声明。

有关函数的知识将在后续章节中进行详细介绍。

通过以上 4 个例子的阐述,可以看出 C 语言程序的基本结构有以下几个特点。

(1) C 语言程序是由函数组成的,每个函数实现相对独立的功能,函数是 C 语言程序的基本模块单元。每个 C 程序必有一个且只能有一个 main()函数,除 main()函数外,还可以有若干个其他函数。因此,函数是 C 程序的基本单位。函数可以是系统提供的库函数,如 printf()函数,也可以是用户自定义函数,如例题 1.4 中的 max()函数。编写 C 程序就是编写一个个的函数。C 的函数库十分丰富,ANSI C 提供了上百个库函数。

(2) 一个函数由函数头部和函数体两部分组成。函数头部由函数名、函数类型、函数属性、函数参数(形式参数)名及参数类型构成,一个函数名后面必须跟一对圆括号,括号内可写函数的参数及其类型,也可没有参数,例如, `main()` 的括号内没有参数, `max(int a, int b)` 的括号内有参数及其类型。函数体即函数头部下面的花括号内的部分。如果一个函数内有多对花括号,则最外层的一对花括号为函数体的范围。函数体一般由声明部分和执行部分构成,声明部分是对函数体要用到的变量及对其所调用函数的声明,执行部分是由若干个语句构成,实现 C 语言程序的功能。

(3) 主函数的位置是任意的,可以放在程序的开头、两个函数之间或程序的结尾。程序的执行总是从主函数 `main()` 开始,并以主函数 `main()` 结束。

(4) C 语言程序书写格式自由。一行内可以写几个语句,一个语句可以分写在多行上。

(5) 每个语句和声明部分的最后必须有一个分号。分号是 C 语句的必要组成部分,是 C 语句的结束标志。但预处理命令、函数头和花括号之后不能加分号。

(6) 标识符、关键字之间必须加一个空格以示隔开。若已经有明显的间隔符,也可不加空格来间隔。

(7) C 语言程序中用 `/*...*/` 和 `//` 的形式加注释,注释可以放在程序的任意位置,其作用是增加程序的可读性,不参与程序的编译和执行。

(8) C 语言程序以小写字母作为基本书写形式,并且 C 语言严格区分字母的大小写,同一字母的大小写被作为两个不同的标识符。

从书写清晰、便于阅读、理解、维护的角度出发,在书写程序时应遵循以下规则。

(1) 一个说明或一个语句占一行。

(2) 用 `{}` 括起来的部分,通常表示程序的某一层结构。`{}` 一般与该结构语句的第一个字母对齐,并单独占一行。

(3) 低一层次的语句相对高一层次的语句以缩进格式书写,以便读起来结构更清晰,增加程序的可读性。

(4) 为了增加程序的可读性和维护性,要添加必要的注释。

在编程时应力求遵循这些规则,以便养成良好的编程风格。

1.4 C 语言字符集、标识符和关键字

任何一种高级语言,都有自己的基本词汇符号和语法规则,程序代码都是由这些基本词汇符号并根据该语言的语法规则编写的,C 语言也不例外,C 语言规定了其所需的基本字符集和标识符。

1.4.1 C 语言字符集

字符是组成语言的最基本元素。C 语言字符集由字母、数字、空格、标点和特殊字符组成。在字符常量、字符串常量和注释中还可以使用汉字或其他可表示的图形符号。

(1) 英文字母:小写字母 `a~z` 共 26 个;大写字母 `A~Z` 共 26 个。

(2) 阿拉伯数字: `0~9` 共 10 个。

(3) 空白符：空格符、制表符、换行符统称空白符。空白符只在字符常量和字符串常量中起作用。在其他地方出现时，只起间隔作用，编译程序时对它们忽略不计。因此在程序中使用空白符与否对程序的编译不发生影响，但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

(4) 标点和特殊字符：+、-、*、/、%、=、{}、()、[]、_(下画线)、'(单引号)、.、:、?、~、<>、&、;、"、|、!、#、^。

1.4.2 C 语言标识符

C 语言的标识符用来表示函数、类型、常量及变量的名称，只起标识作用。标识符由下画线或英文字母及数字构成。C 语言的标识符有以下三类。

1. 保留字

保留字又称关键字，其每一个都有特定含义，不允许用户把它当作变量名使用，C 语言的保留字都是用小写英文字母表示，如表 1.1 所示。

表 1.1 C 语言保留字

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

C99 还定义了新增加的保留字，如 `_Bool`、`_Imaginary`、`restrict`、`_Complex`、`inline`。在 C89、C99 的 C 语言中，保留字也都是小写的。

2. 预定义标识符

除上述保留字外，还有一类具有特殊含义的标识符，它们被用作库函数名和预编译命令，这类标识符在 C 语言中称为预定义标识符。一般不要把这样的标识符再定义为其他标识符来使用。

预定义标识符包括预编译程序命令和 C 编译系统提供的库函数名。其中，预编译程序命令有 `include`、`define`、`undef`、`ifdef`、`ifndef`、`endif`、`line`。

3. 用户定义标识符

用户定义标识符是程序员根据自己的需要定义的一类标识符，用于标识变量、符号常量、用户定义函数、类型和文件指针等，是由字母、下画线和数字构成，但必须用字母或下画线开头。

以下标识符是合法的：

`abc,a3,BOOK_1,sum5`

以下标识符是非法的：

`3x` 以数字开头

`x*y` 出现非法字符*

`-3ab` 以减号开头

在使用标识符时应注意以下几点。

(1) 标准 C 不限制标识符的长度，但受各种版本的 C 语言编译系统的限制，同时也受到具体机器的限制。例如，IBM-PC 的 MS C 规定程序中使用的标识符中只有前 8 个字符有意义，超过 8 个字符以外的字符不做识别。

(2) 在标识符中，大小写是有区别的。例如，BOOK 和 book 是两个不同的标识符。

(3) 标识符虽然由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量做到有相应的意义，以便于阅读理解，即“顾名思义”。

1.4.3 C 语言关键字

关键字是由 C 语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C 语言的关键字分为以下几类。

1. 类型说明符

类型说明符用于定义、说明变量、函数或其他数据结构的类型，如前面例题中用到的 `int` 等。

2. 语句定义符

语句定义符用于表示一个语句的功能，如例 1.4 中用到的 `if-else` 是条件语句的语句定义符。

3. 预处理命令

预处理命令用于表示一个预处理命令，如前述的 4 个例子中都用到的 `#include` 及后述的 `#define`。本书将在第 7 章重点介绍预处理命令。

C 语言除了上述介绍的标识符、关键字外，还有以下值得注意的常用词汇。

1. 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算的功能。运算符由一个或多个字符组成。

2. 分隔符

在 C 语言中，分隔符采用逗号和空格两种。逗号主要用于类型说明和函数参数列表中，分隔各个变量。空格多用于语句各单词间做间隔符。在关键字、标识符之间必须有一个以上的空格符做间隔符，否则将会出现语法错误。例如，若将 `int x` 写成 `intx`；则 C 编译器会把 `intx` 当成一个标识符处理，其结果就必然出编译错误。

3. 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种，将在后续章节中详细介绍。

4. 注释符

C 语言中的注释有两种：一种是以 `/*` 开头并以 `*/` 结尾的串；另一种是以 `//` 开始的后面跟着的字符串。程序编译时，不对注释做任何处理。注释用来向用户提示或解释程序的

作用，可放在程序中的任何位置。

1.5 C语言程序的开发环境

在 1.3 节我们认识了 4 个 C 语言的源程序，源程序不能直接在计算机上执行，需要用编译程序将源程序翻译成二进制形式的代码。C 语言的源程序的扩展名为.c 或.cpp。源程序经过编译程序翻译所得到的二进制代码称为目标程序，目标程序的扩展名为.obj。目标代码尽管已经是机器指令，但是其还没有解决函数调用问题，需要将各个目标程序与库函数连接，才能形成完整的可执行程序。目标程序与库函数连接，形成完整的可在操作系统下独立执行的程序称为可执行程序。可执行程序的扩展名为.exe，可能在 DOS 或 Windows 环境下直接运行。

可执行程序的运行结果是否正确需要经过验证，如果结果不正确则需进行调试。调试程序往往比编写程序更困难、更费时。这就是 C 语言程序的开发过程。C 语言的编译程序软件版本较多，本书采用 Visual C++ 6.0(中文版)为集成开发环境，介绍 C 程序的编辑、编译、连接和运行的基本过程。

1.5.1 Visual C++ 6.0 集成开发环境介绍

Visual C++ 6.0 提供了一个支持可视化编程的集成开发环境，是集源程序编辑、代码编译与调试于一体的集成开发环境，同时也是 Windows 环境中最主要的应用系统之一。Visual C++ 不仅是 C/C++ 的集成开发环境，而且与 Windows 32 紧密相连，因此利用 Visual C++ 可以完成各种各样的应用程序的开发，从底层软件直到上层直接面向用户的软件。而且 Visual C++ 强大的调试功能也为大型复杂软件的开发提供了有效的排错手段。

安装好 Visual C++ 6.0 环境后，启动，弹出如图 1.3 所示的可视化窗口界面。Visual C++ 6.0 窗口与 Windows 窗口一样，由标题栏、菜单栏、工具栏、工作区窗口、程序编辑区窗口和调试信息显示区窗口组成。

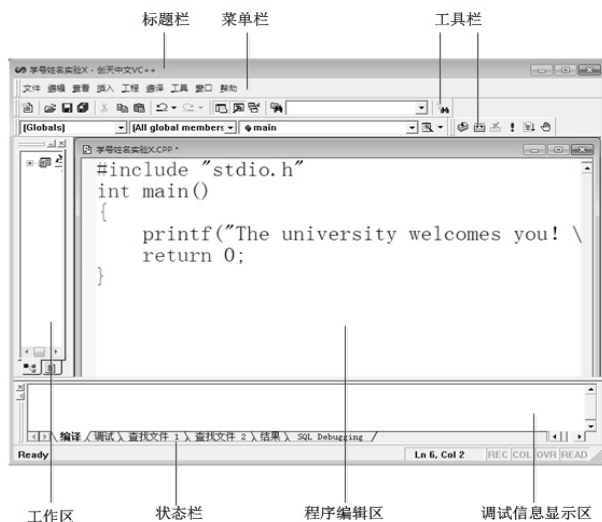


图 1.3 Visual C++ 6.0 集成开发环境

1.5.2 在 Visual C++ 6.0 环境下建立和运行 C 语言程序的步骤

在 Visual C++ 6.0 集成环境下建立 C 语言程序的方法有两种：一种是标准创建法，另一种是快捷创建法。

标准创建法创建 C 语言程序的基本步骤：首先是创建一个工程项目，然后在这个工程项目下创建 C 程序。

【例题 1.5】编写程序：在屏幕上打印 The university welcomes you! 字符串。

1. 建立一个 C 程序工程项目

(1) 启动 Visual C++ 6.0，选择“文件”菜单下的“新建”命令，在弹出的对话框中单击“工程”选项卡，如图 1.4 所示，选择工程类型为 Win32 console Application，并在右边的工程文本框中输入 C 程序工程项目名称为“例题 1_5”，单击位置文本的右下角按钮，选择程序所存放的位置，如 D:\学号姓名实验 1，单击“确定”按钮，打开如图 1.5 所示的对话框。

(2) 在图 1.5 的对话框中选择 An empty project，单击“完成”按钮，系统会显示新建工程信息，单击“确定”按钮即可。一个工程名为“例题 1_5”的工程项目创建好了，如图 1.6 所示。但该工程中没有任何文件，需要再创建一个 C 语言源文件加载到该工程中。



图 1.4 创建工程类型及工程名和路径

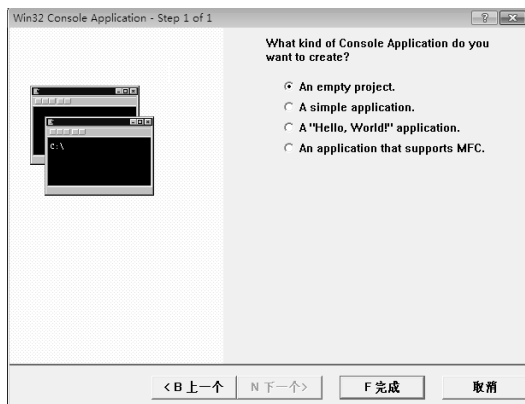


图 1.5 选择 Windows 应用程序类型

2. 添加 C 源程序文件到工程项目

(1) 选择“文件”菜单下的“新建”命令，在弹出的对话框中单击“文件”选项卡。单击 C++ Source File 选项后，在右边的“文件”框中输入文件名称，如例题 1_5，注意不能取名例题 1.5，因为“.”后面的 5 会被当成扩展名；可以取名为 1.5.cpp，这样最后的.cpp 会被当成扩展名，然后进入如图 1.7 所示的“新建 C++源文件”窗口。单击“确定”按钮，进入编辑 C 语言源程序的窗口工作区，如图 1.8 所示。

(2) 编辑的 C 语言源程序如图 1.9 所示。

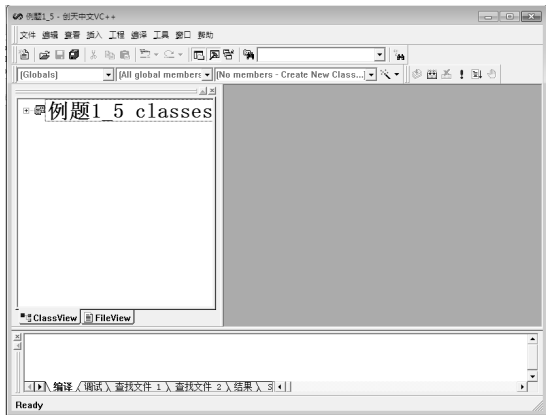


图 1.6 新建工程项目窗口

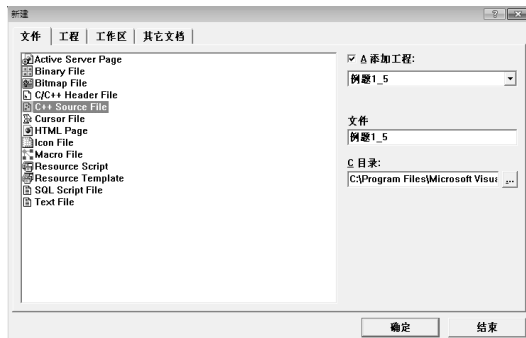


图 1.7 新建 C++源文件

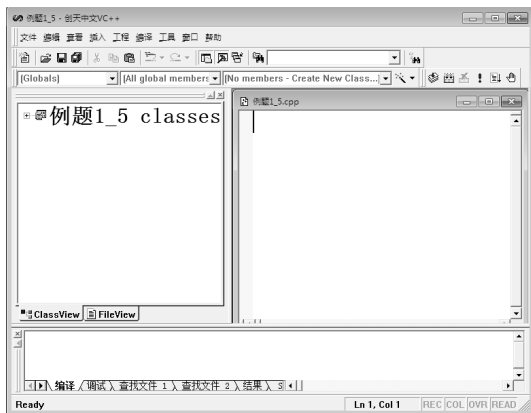


图 1.8 编辑 C 语言源程序窗口工作区

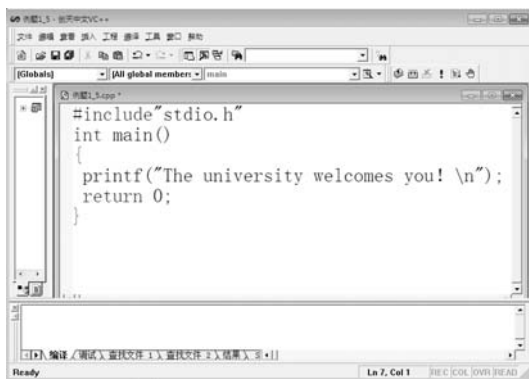



图 1.9 编辑的 C 语言源程序

3. 编译、连接和运行 C 程序

(1) 编译、连接 C 语言源程序。有两种方法：一种是通过菜单实现，另一种是通过工具栏的按钮实现。单击 Build MiniBar 工具栏中的 Compile 按钮，若提示信息窗口没有错误提示信息，说明没有语法错误并生成了目标文件，则可继续单击 Build 按钮，也说明生成的目标文件与 C 语言提供的库函数连接成功并生成可执行文件。若上述过程中出现错误信息提示，则需要修改直至无错误为止。编译和连接 C 程序如图 1.10 所示。

(2) 运行程序显示程序执行结果。有两种方法：一种是通过菜单实现，另一种是通过工具栏的按钮实现。单击 Build MiniBar 工具栏中的 Execute program 按钮，程序运行结果如图 1.11 所示。按任意键则显示结果窗口消失。

快捷式创建 C 程序的方法有两种：一种是新建扩展名为.txt 的文本文件，将其直接改扩展名为.cpp 的 C 程序，选择打开方式为 Visual C++ 6.0，编译连接运行。另一种是启动 Visual C++ 6.0，然后单击工具栏最左边的“新建文本文件”按钮，生成一个文本文件，最后将其保存为.cpp 或.C 的源程序文件。第二种方法具体说明如下。

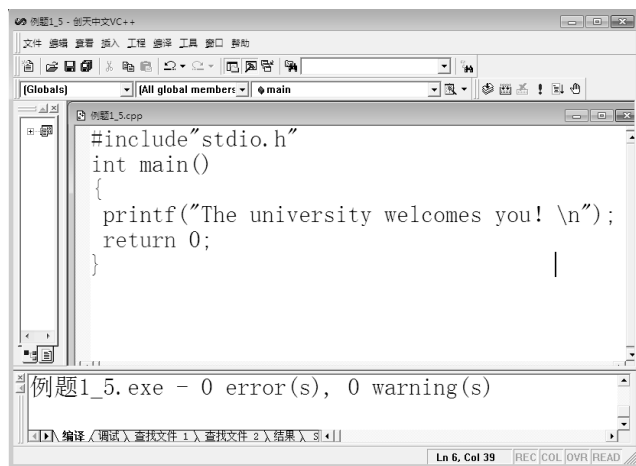


图 1.10 编译和连接 C 程序



图 1.11 程序运行结果

(1) 建立 C 程序文件。启动 Visual C++ 6.0, 并单击工具栏最左边的“新建文本文件”按钮, 生成一个文本文件, 然后保存或另存为 C 源程序文件, 即文件名以 .C 为扩展名, 如输入例题 5_1.C。注意, 这里一定要加 .C 扩展名, 否则将生成以 .TXT 为扩展名的文本文件, 不能进行编译运行, 接着输入源程序代码, 如图 1.12 所示。

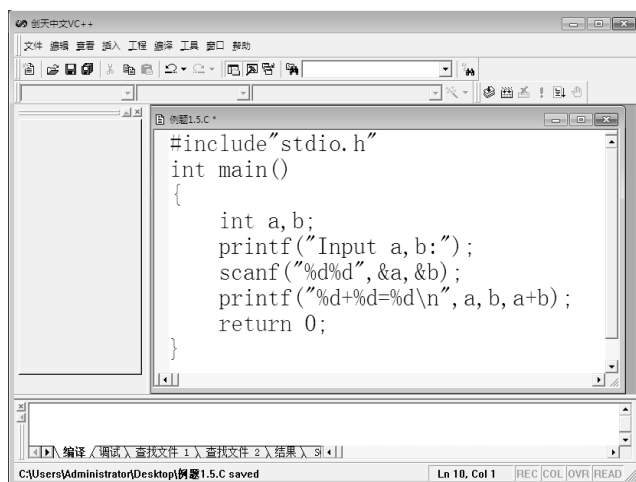


图 1.12 源程序编辑窗口

(2) 编译运行。源程序编辑完毕, 选择“文件”菜单中的“保存”命令进行存盘。按 F7 键或单击工具栏上的编译按钮, 可以编辑、连接程序而不运行程序。

由于 Visual C++ 6.0 有工作区的要求, 因此按下 F7 键后, 系统提示需要建立工作区, 如图 1.13 所示, 单击“是”按钮, 系统会自动建立工作区, 结果如图 1.14 所示。

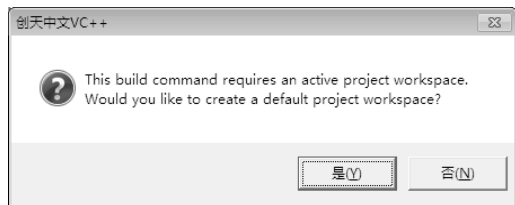


图 1.13 提示建立工作区

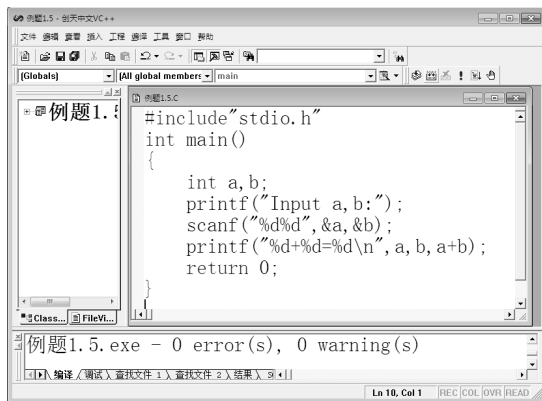


图 1.14 编译连接源程序

如果要编辑下一个 C 源程序, 由于新文件不会自动加入工作区, 所以一定要先关闭工作区, 接着按上述方法建立、编辑新的 C 源程序, 也可采用加注释的方法将调试运行的程序注释掉, 接着写下一个程序。

本章小结

本章介绍了程序的概念及程序设计语言的发展历程, 以及 C 语言的基本特点和 C 语言程序书写的基本原则; 通过四个例题的具体分析来介绍 C 语言程序的基本结构, 介绍了 C 语言字符集、标识符及关键字, 以及 Visual C++ 6.0 的集成开发环境。

计算机完成的各项工作都是运行程序的结果。计算机程序设计语言经历了机器语言、汇编语言、高级语言 3 个主要阶段, 程序设计方法也在从结构化程序设计方法向面向对象的程序设计方法发展。

C 语言是一种结构化的高级语言, 它简洁、紧凑、功能强大、使用方便、可移植性好。C 语言可以实现汇编语言的大部分功能, 也可以直接操作计算机硬件。

C 语言的字符集、标识符和关键字是后续课程中学习经常要用到的内容, 学习和掌握好这部分内容, 对后面章节的学习大有帮助并且能起到很好的作用。

用 Visual C++ 6.0 集成开发环境创建 C 语言的基本过程如下。

- (1) 创建工程项目。
- (2) 在该项目下创建一个 C 语言文件。
- (3) 编辑 C 语言源程序。
- (4) 编译 C 语言源程序得到目标程序, 将目标程序与 C 语言的库函数连接生成可执行程序, 运行可执行程序将得到程序的运行结果。

使用 C 语言编写程序, 必须遵循 C 程序的基本规则: C 程序由函数构成, 每个 C 程序有且只有一个主函数, C 程序总是从主函数开始执行; C 程序中的每条语句必须用分号结尾, 程序中的变量必须先定义后使用, 先赋值后引用, C 语言程序严格区分字母的大小写。

初学者编写 C 语言程序时, 要注重培养良好的书写风格, 要注意不遗漏每条语句后面的分号、字母的大小写不能混用, 要培养科学、严谨的学习作风, 更要在学习实践中形成自己独有

的编程风格。

易错提示

1. C 语言源程序有且只有一个主函数 `main()`，程序总是从主函数的第 1 条可执行语句开始执行。
2. C 语言源程序的基本结构是书写程序的基础，特别要注意花括号、头文件等的运用。
3. C 语言源程序的语句结束标志是分号“；”，书写或上机编程时不要遗漏。
4. 程序中使用标识符时要遵循标识符的相关约定，同时要区分字符的大小写。

习题 1

1. 程序设计语言经历了哪 3 个主要阶段？每个阶段有什么特点？
2. C 语言有哪些特点？这些特点你是怎样理解的？
3. 用一个事例简述 C 语言程序的基本结构。
4. 下列标识符哪些是合法的，哪些是非法的？若是非法的要指明其原因。
`3H_R` `_3H_R` `_3H&R` `H3R` `ifD.K.Jon` `a*b2` `Sstu`
5. 选择题
 - (1) C 语言是一种()。

A. 机器语言	B. 汇编语言	C. 高级语言	D. 以上都不是
---------	---------	---------	----------
 - (2) C 程序总是从()开始执行。

A. 程序中的第一条语句	B. 程序中第一条可执行语句
C. 程序中的第一个函数	D. 程序中的 <code>main()</code> 函数
 - (3) 下列叙述正确的是()。

A. C 语言源程序可以直接在 Windows 环境下运行
B. 编译 C 语言源程序得到目标程序可以直接在 Windows 环境下运行
C. C 语言源程序经过编译、连接得到可执行程序可以运行
D. 以上说法都是正确的
 - (4) 下列说法正确的是()。

A. C 语言程序书写时，不区分大小写字母
B. C 语言程序书写时，一个语句可分成几行书写
C. C 语言程序书写时，一行只能写一个语句
D. C 语言程序书写时，每行必须有行号
 - (5) C 语言规定，在一个源程序中，`main()`函数的位置()。

A. 必须在最开始	B. 必须在系统调用的库函数的后面
C. 可以是任意位置	D. 必须在最后

(6) C 语言程序经过编译后生成的文件名的后缀是()。

- A. .c B. .obj C. .exe D. .cpp

(7) 下面叙述中, 错误的是()。

- A. 分号是 C 语句的必要组成部分 B. C 程序的注释可以写在语句的后面
C. 函数是 C 程序的基本单位 D. 主函数的名字不一定用 main() 表示

(8) 关于 C 语言程序的注释以下描述错误的是()。

- A. 由 “/*” 开头, “*/” 结尾 B. 由 “/*” 开头, “/*” 结尾
C. 由 “//” 开头 D. 由 “/*” 或 “//” 开头

6. 判断题

- (1) 主函数是系统提供的标准函数。()
(2) 一个 C 程序可以有一个或多个主函数。()
(3) C 程序首先执行程序的第一个函数。()
(4) 调用大多数 C 语言标准函数, 可以不使用包含命令。()
(5) C 语言允许多条语句写在同一行。()
(6) 语句 `int number;` 和 `int Number` 定义的是同一个整型变量。()

7. 参考本章例题, 编写下列程序。

(1) 编写一个 C 程序, 要求在屏幕上打印以下信息:

```
*****
                          我是一名大学生
                          我热爱我的学校
*****
```

(2) 通过键盘输入一个实数, 要求在屏幕上输出该数的平方值(提示数 x 的平方可用 $x*x$ 来表示)。

(3) 在 Visual C++ 6.0 集成环境下运行本章的 4 个事例, 熟悉上机方法和步骤。

【实验 1】编程环境与简单程序的运行

1. 实验目的

- (1) 了解和使用 Visual C++ 6.0 集成开发环境。
(2) 能在 Visual C++ 6.0 集成开发环境编写运行简单的 C 程序。
(3) 掌握 C 语言源程序的建立、编辑、修改、保存编译和运行的基本步骤。

2. 实验预备

(1) Visual C++ 6.0 集成开发环境界面划分成 4 个主要区域: 菜单栏和工具栏、工作区窗口、代码编辑窗口和信息提示窗口。同学们启动 Visual C++ 6.0 开发环境后加以对照认识。

(2) 预习例题 1.5 在 Visual C++ 6.0 环境下建立和运行 C 程序的步骤。

3. 实验内容

- (1) 编写程序 1: 已知圆的半径, 求圆的周长和面积。
(2) 编写程序 2: 输入任意的 3 个数, 求这 3 个数的平均数。
(3) 编写程序 3: 输入矩形的两个边长, 求矩形的面积。

4. 实验提示

(1) 启动 Visual C++ 6.0, 在 D 盘下建立学号姓名实验 1 的项目名称, 建立学号姓名实验 1

的文件名并添加到“学号姓名实验 1”的项目中(标准方式建立 C 程序法)。或者在 D 盘下直接建立“学号姓名实验 1.C”的 C 程序(快捷方式建立 C 程序法)。

(2) 编辑 C 语言源程序。

```

程序 1 代码: #include<stdio.h>
int main()
{
    int r;
    float l,s;
    r=5;
    l=2*3.14159*r;
    s=3.14159*r*r;
    printf("r=%d,l=%f,s=%f\n",r,l,s);
    return 0;
}

程序 2 代码: #include<stdio.h>
int main()
{
    int a,b,c;
    float aver;
    printf("请输入 3 个数 a,b,c:");
    scanf("%d%d%d",&a,&b,&c);
    aver=(a+b+c)/3;
    printf("aver=%f\n",aver);
    return 0;
}

程序 3 代码: #include<stdio.h>
int main()
{
    int x,y;
    float area;
    printf("please input x,y:");
    scanf("%d%d",&x,&y);
    area=x*y;
    printf("area is %f\n",area);
    return 0;
}

```

(3) “学号姓名实验 1”的文件包含以上 3 个 C 源程序。当编辑完成程序 1 后,对其进行编译、连接和运行,分析其结果是否符合题目要求,符合后则对 C 语言源程序加上注释。编辑程序 2 并对其进行编译、连接和运行。编辑程序 3 并对其进行编译、连接和运行。

5. 实验报告

(1) 写出创建“学生姓名实验 1”项目的步骤和建立健全.CPP 文件并添加到项目的步骤;写出直接创建“学生姓名实验 1”C 程序的步骤。

(2) 实验过程中使用到的变量的数据类型、使用的相关格式符的含义。

(3) 通过 3 个 C 程序的上机实验,总结出 C 源程序的构成要素有哪些。

(4) 在上机实验过程中遇到了哪些问题,对这些问题是如何解决的?例如,程序 2 中输入 3 个数,你是怎样输入的;程序 3 中输入 2 个数,你是怎样输入的?