

■ 高 等 学 校 教 材

C/C++ 程序设计

C/C++ CHENGXU SHEJI

高俊文 主编



人民邮电出版社
POSTS & TELECOM PRESS

高等学校教材

C/C++程序设计

高俊文 主编

人民邮电出版社

图书在版编目 (CIP) 数据

C/C++程序设计 / 高俊文主编. —北京: 人民邮电出版社, 2005.9

高等学校教材

ISBN 7-115-13997-0

I. C... II. 高... III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 097217 号

内 容 简 介

本书以程序设计为主线, 系统介绍了 C/C++ 语言及其程序设计技术。全书共分为 13 章, 内容包括: C 语言知识、C++ 基础知识、Turbo C 2.0 及 Visual C++ 6.0 环境下的程序调试方法等。其中, C 语言的基本知识部分依据教育部考试中心最新发布的《全国计算机等级考试考试大纲》编写而成。

本书结构清晰, 通俗易懂, 应用大量的程序实例, 强化程序设计能力, 注重实际应用能力的培养。每章还设计了实训指导, 以供读者上机演练。

本书既可作为本科和高职高专程序设计入门课教材, 也适用于全国计算机等级考试的辅导教材及广大程序设计初学者的自学参考书。

高等学校教材

C/C++程序设计

◆ 主 编 高俊文

责任编辑 杨 莹

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 17.25

字数: 407 千字

印数: 1—3 000 册

2005 年 9 月第 1 版

2005 年 9 月北京第 1 次印刷

ISBN 7-115-13997-0/TP · 4967

定价: 24.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

前 言

C 语言是一种通用的程序设计语言。目前我国高等院校大部分以 C 语言作为学生程序设计入门的语言。在全国计算机的各类程序设计考试中也多为 C 语言内容,本书 C 语言的基本知识部分就是依据教育部考试中心最新发布的《全国计算机等级考试考试大纲》编写而成。

本书的编写目标是应用大量的程序实例,强化程序设计能力,注重实际应用能力的培养。书中对语法讲解不多,主要是使用大量的实例程序让读者通过阅读和上机实践来掌握相应的语法知识。除实例程序外,全书还安排了两章,专门介绍 50 多道应用程序。应用程序主要注重算法的分析和编程的实践。另外,为指导学生快速调试程序,掌握所学的内容,也为方便自学,每章还设计了实训指导,以供读者上机实践。

全书共分为 13 章。第 1 章介绍 C 语言基础,包括程序设计算法和结构化程序设计思想、C 语言程序开发及 Turbo C 集成开发环境。第 2 章介绍数据类型、运算符和表达式,包括 C 语言的语法、基本数据类型及各种运算符的功能、优先级及结合性。第 3 章介绍 C 语言程序控制结构,包括 if、switch、while、do-while 和 for 语句的使用。第 4 章介绍数组,包括一维、二维和字符数组的使用。第 5 章介绍函数与编译预处理,包括函数基本知识、函数的嵌套和递归调用、函数的变量和函数参数传递的用法。第 6 章介绍指针,包括指针变量和指针常量、指针运算、指针和数组的关系及指针和函数的关系。第 7 章介绍结构体、共用体和位运算,包括结构体和共用体类型定义及其使用方法、位运算的作用及其应用。第 8 章介绍文件,包括文件指针的概念及其正确使用方法、文件读写函数的使用和文件指针的定位方法。第 9 章介绍 C 程序设计应用实例。第 10 章介绍 C++ 初步认识,包括面向对象的基本思想和 Visual C++ 6.0 集成开发环境。第 11 章介绍类和对象,包括类与对象定义和使用方法、构造函数与析构函数的定义和使用、继承与派生的定义与使用。第 12 章介绍流与文件操作,包括流和文件的概念、C++ 中流类库中的流类层次结构、文件流类和 C++ 中的文件操作。第 13 章介绍 C++ 综合应用实例。

本书由高俊文主编,陈真明、欧阳国军和李法春为副主编。其中高俊文编写第 1 章内容,陈真明编写第 4 章和第 6 章内容,欧阳国军编写第 11 章部分内容、第 12 章和第 13 章内容,李法春编写第 5 和第 7 章内容,黄汉南编写第 8 章和第 9 章内容,张鹏飞编写第 2 章内容,廖福保编写第 10 章及第 11 章部分内容,张文梅编写第 3 章及附录内容,全书由高俊文负责策划、统稿和定稿。在编写中,编者参考了国内外有关 C/C++ 程序设计的书刊及文献资料,在此对文献的作者表示感谢。

本书在编写和出版过程中得到我校电子与信息工程系教师和计算机科学系教师提供大力帮助,在此表示诚挚的谢意。

由于时间仓促和编者水平有限,书中错误或不妥之处在所难免,恳请读者批评指正。

编者的 E-mail 地址: gaojunwen@21cn.com 和 jwgao@gdaib.edu.cn。

编 者
2005 年 5 月

目 录

第 1 章 C 语言基础	1
1.1 程序设计方法.....	1
1.1.1 程序设计概述.....	1
1.1.2 程序设计的一般过程.....	2
1.1.3 结构化程序设计思想.....	3
1.2 C 语言简介.....	4
1.2.1 C 语言的发展.....	4
1.2.2 C 语言的特点.....	5
1.3 C 语言程序基本结构及书写风格.....	6
1.3.1 C 语言程序的基本结构.....	6
1.3.2 C 语言程序的书写风格.....	7
1.4 上机操作基础.....	7
1.4.1 C 语言程序的开发过程.....	7
1.4.2 Turbo C 集成开发环境.....	8
习题.....	11
第 2 章 数据类型、运算符和表达式	13
2.1 C 语言的基本语法单位.....	13
2.2 常量与变量.....	14
2.3 基本数据类型.....	15
2.3.1 整型数据.....	15
2.3.2 实型数据.....	17
2.3.3 字符型数据.....	19
2.4 变量的初始化.....	20
2.5 运算符和表达式.....	20
2.5.1 算术运算符和算术表达式.....	21
2.5.2 表达式计算中的数据类型转换.....	22
2.5.3 赋值运算符和赋值表达式.....	23
2.5.4 逗号运算符与逗号表达式.....	25
习题.....	25
第 3 章 C 语言程序控制结构	27
3.1 if 语句.....	28
3.1.1 if 语句的三种基本形式.....	28

3.1.2	If 语句使用注意事项	30
3.1.3	if 语句的嵌套	30
3.1.4	条件运算符和条件表达式	31
3.2	switch 语句	32
3.2.1	switch 语句(不带 break 语句)一般形式	32
3.2.2	switch 语句(带 break 语句)一般形式	33
3.2.3	switch 语句使用注意事项	34
3.3	for 语句	34
3.3.1	for 语句的一般形式	35
3.3.2	for 语句使用注意事项	36
3.4	while 语句	38
3.4.1	while 语句的一般形式	38
3.4.2	while 语句使用注意事项	38
3.5	do-while 语句	39
3.5.1	do-while 语句的一般形式	39
3.5.2	do-while 语句使用注意事项	40
3.6	程序转移控制语句	40
3.6.1	break 语句	41
3.6.2	continue 语句	41
3.6.3	goto 语句	42
	习题	43
第 4 章	数组	49
4.1	一维数组	49
4.1.1	一维数组定义和引用	49
4.1.2	一维数组的存储形式	50
4.1.3	一维数组的使用	50
4.1.4	一维数组程序举例	51
4.2	二维数组	52
4.2.1	二维数组的定义和引用	52
4.2.2	二维数组的存储形式	53
4.2.3	二维数组的使用	53
4.2.4	二维数组程序举例	54
4.3	字符数组	54
4.3.1	字符数组的定义和引用	55
4.3.2	字符串	55
4.3.3	字符数组的初始化及存储形式	55
4.3.4	字符数组的输入和输出	56
4.3.5	字符串处理函数	58

4.3.6 字符串处理函数的应用	60
习题	61
第5章 函数与编译预处理	64
5.1 函数的定义和调用	64
5.1.1 函数的定义	64
5.1.2 函数的调用	67
5.1.3 函数的返回值与函数的类型	68
5.1.4 函数声明	69
5.2 形参与实参	70
5.2.1 函数的值调用和引用调用	70
5.2.2 函数的参数	71
5.3 函数的嵌套调用和递归调用	73
5.3.1 函数的嵌套调用	73
5.3.2 函数的递归调用	74
5.4 局部变量和全局变量	76
5.4.1 局部变量	76
5.4.2 全局变量	77
5.5 内部函数和外部函数	78
5.5.1 内部函数	78
5.5.2 外部函数	79
5.6 编译预处理	80
5.6.1 宏定义	80
5.6.2 文件包含	83
习题	84
第6章 指针	87
6.1 变量的指针	87
6.1.1 指针的概念	87
6.1.2 指针变量的定义和赋值	88
6.1.3 指针变量的引用	89
6.1.4 指针变量作函数参数	90
6.2 数组的指针	92
6.2.1 一维数组的指针	93
6.2.2 二维数组的指针	94
6.2.3 字符串的指针	96
6.2.4 数组名和指针变量做函数参数	98
6.3 返回指针值的函数	100
6.4 指针数组与多级指针	102

6.4.1 指针数组	102
6.4.2 多级指针	104
6.5 函数指针	105
习题	106
第7章 结构体、共用体和位运算	109
7.1 结构体类型	109
7.1.1 结构体定义、引用和初始化	109
7.1.2 结构体类型变量的定义	111
7.1.3 结构体变量的使用	112
7.1.4 结构体变量的初始化	113
7.2 结构体数组与结构体指针	114
7.2.1 结构体数组	114
7.2.2 结构体指针	116
7.3 共用体类型	118
7.3.1 共用体类型的定义	119
7.3.2 共用体变量的使用与初始化	119
7.3.3 共用体类型数据的特点	120
7.4 枚举类型	122
7.5 自定义类型	124
7.6 用指针处理链表	125
7.6.1 C语言动态分配函数	125
7.6.2 单链表及其基本运算的实现	126
7.7 位运算	129
7.7.1 位运算概念	129
7.7.2 位运算符	130
7.7.3 位段	133
习题	135
第8章 文件	139
8.1 C文件概述	139
8.2 标准文件的操作	141
8.2.1 标准文件读写函数的功能	141
8.2.2 标准文件读写函数的使用	142
8.3 一般文件的操作	145
8.3.1 文件类型指针	145
8.3.2 文件指针和读写指针	146
8.3.3 文件的打开与关闭	147
8.4 文件的读写	149

8.4.1	fputc()函数和 fgetc()函数	149
8.4.2	fread()函数和 fwrite()函数	152
8.4.3	fprintf()函数和 fscanf()函数	155
8.4.4	其他读写函数	156
8.5	文件读写指针的定位(函数)	157
8.5.1	rewind()函数	157
8.5.2	文件随机定位函数 fseek()	158
8.5.3	ftell()函数	159
8.6	出错的检测	159
8.6.1	ferror()函数	159
8.6.2	clearerr()函数	159
	习题	160
第9章	C程序设计应用实例	163
	习题	184
第10章	C++初步认识	185
10.1	面向对象的方法	185
10.1.1	关于面向对象	185
10.1.2	为什么要使用面向对象	186
10.1.3	面向对象的基本概念	187
10.2	C++程序的框架结构	190
10.3	Visual C++ 6.0 集成环境使用	196
	习题	199
第11章	类和对象	200
11.1	类的构成	200
11.1.1	从结构到类	200
11.1.2	类的定义	201
11.1.3	类和结构体的关系	204
11.2	对象的定义和使用	204
11.2.1	对象的定义格式	204
11.2.2	对象成员的代表方法	204
11.2.3	对象的初始化	205
11.3	继承与派生类	207
11.3.1	基类和派生类	207
11.3.2	单继承	208
	习题	212

第 12 章 流与文件操作	216
12.1 流及其类库简介	216
12.1.1 iostream 类库的头文件	217
12.1.2 输入/输出流类和对象	217
12.1.3 文件和流	218
12.2 常用输入/输出函数	219
12.2.1 使用成员函数 put() 输出一个字符	219
12.2.2 使用成员函数 write() 输出一个字符串	219
12.2.3 使用成员函数 get() 获取一个字符	220
12.2.4 使用成员函数 read() 读取一个字符串	221
12.3 文件的打开和关闭	221
12.4 顺序访问文件	224
12.4.1 顺序文件的建立	224
12.4.2 读取顺序访问文件中的数据	226
12.5 随机访问文件	227
12.5.1 随机访问文件的结构	227
12.5.2 文件的随机读写方法	228
12.5.3 随机访问文件的建立与数据存取	229
习题	233
第 13 章 C++ 综合应用实例	236
13.1 公司人员管理系统	236
13.1.1 问题描述 (功能要求)	236
13.1.2 问题的解决方案	236
13.1.3 公司人员管理系统类的设计	237
13.2 公司人员管理系统的实现	237
实训	246
实训 1	246
实训 2	246
实训 3	247
实训 4	247
实训 5	248
实训 6	249
实训 7	250
实训 8	251
实训 9	252
实训 10	252

实训 11	252
实训 12	253
实训 13	254
附录	255
附录 1 ASC II 代码表	255
附录 2 C 运算符的优先级与结合性	256
附录 3 Turbo C2.0 常用的库函数及其标题文件	257
参考文献	261

第 1 章 C 语言基础

本章学习目的:

- 熟悉程序设计的一般过程、算法和结构化程序设计思想;
- 了解 C 语言程序的发展和特点;
- 掌握 C 语言程序的基本构成;
- 熟悉 C 语言程序的开发过程;
- 熟悉 Turbo C 集成开发环境。

本章学习重点:

- C 语言程序的基本构成;
- C 语言程序的开发过程。

随着计算机技术的迅速发展,软件开发领域出现多种程序设计语言。C 语言作为生命力最强的高级程序设计语言之一,非常适合编写系统程序,曾被誉为是真正的程序设计者的语言。C 语言流行广,影响大。从我们国家来说,自 1994 年由国家教育部考试中心推出二级 C 语言以来,备受社会关注,C 语言已成为讲解程序设计的标准语言。

由于面向对象技术的发展和广泛应用,C 语言作为结构化程序设计语言已逐渐被面向对象的程序设计语言 C++ 替代。C++ 是一种更好的 C 语言,是 C 语言的一个超集,是 C 语言的改良版本。因此,想要学好 C++ 语言,就首先要学习和掌握好 C 语言。在讲解 C 语言之前,我们先熟悉程序设计的一般过程、算法和结构化程序设计思想。

1.1 程序设计方法

1.1.1 程序设计概述

程序是为解决某一问题而编写的一组有序指令的集合。通常,将解决一个实际问题的具体操作步骤用某种程序设计语言描述出来,就形成了程序。计算机程序设计语言可以归纳为机器语言、汇编语言和高级语言三类。

1. 机器语言

机器语言是计算机硬件系统可识别的二进制指令构成的程序设计语言。机器语言是面向机器的语言,与特定的计算机硬件设计密切相关,因机器而异,可移植性差。它的优点是机器能够直接识别,执行速度快。缺点是记忆、书写、编程困难,可读性差且容易出错,因此就产生了汇编语言。

2. 汇编语言

汇编语言是一种用助记符号代表等价的二进制机器指令的程序设计语言。汇编语言也

是一种直接面向计算机所有硬件的低级语言，但计算机不能直接执行汇编语言程序，必须经过汇编程序翻译成机器语言程序后才能在计算机上执行。可以看出，从机器语言到汇编语言是计算机语言发展史上里程碑式的进步。

3. 高级语言

高级语言是一种用接近自然语言和数学语言的语法、符号描述基本操作的程序设计语言。它符合人的逻辑思维方式，简单易学。目前，常见的高级语言有 Visual Basic、Java、C、C++、C#、Delphi 等。用高级语言编写的程序通常称为“源程序”，而由二进制的“0”、“1”代码构成的程序称为“目标程序”。用高级语言编写的程序计算机同样不能直接执行，要用翻译程序将其转换成机器语言目标程序后才能执行。例如用 C 语言编写的程序，必须先经 C 编译系统翻译成目标程序，再连接成可执行文件后才能执行。

1.1.2 程序设计的一般过程

一个程序应包括数据的输入、数据的加工处理和数据的输出三大部分。要设计出一个好的程序，必须了解计算机程序设计的过程，如图 1.1 所示。

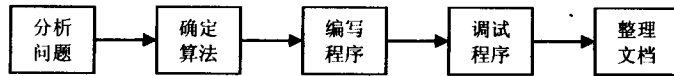


图 1.1 程序设计的一般过程

1. 分析问题

按照任务所提出的要求，对要处理的任务进行调查分析，明确要实现的功能，并选择合适的解决方案。要分析哪些是原始数据、从哪里来，要如何加工处理数据，结果如何输出等方面，从而确定和设计数据的组织方式，即数据结构。

2. 确定算法

算法是解决问题的方法和步骤。世界著名的计算机科学家 Wirth（沃思）曾提出一个用来表达程序设计实质的著名公式：程序=算法+数据结构。就是说：“程序是在数据的特定的组织方式的基础上，对抽象算法的具体描述”。因此，在分析问题时，必须认真考虑数据结构，然后针对具体的数据结构设计相应的解决问题的方法和步骤，即算法。

描述算法的方法有多种，常用的有自然语言、结构化流程图、N-S 图和伪码等。在这里主要介绍用自然语言描述和用流程图描述算法的方法。

(1) 用自然语言描述

用自然语言描述算法通俗易懂，但容易出现歧义，且表达的算法与计算机高级语言形式差距较大，通常只在很简单的问题中使用。

例 1.1 求整数 1 到 100 的和

首先我们分析这个问题，明确它有三项功能：输入 1 到 100 的整数、求和、输出结果；要处理的原始数据为：1 到 100 的整数，它可以根据表达式的规律自动生成，即 n 的初值为 1，它的后一项比前一项多 1，一直加到最后项为 100 止；要进行的处理为：累加求和，并显示。这里用 sum 代表和，用 n 代表 1 到 100 的某项值。因此，该问题的数据组织形式为： $sum=sum+n$ ， $n=n+1$ ，其中 $n \leq 100$ 。

用自然语言描述如下所示：

- ① sum 置 0、n 置 1;
- ② sum 进行累加 \rightarrow sum=sum+n;
- ③ n 增加 1 \rightarrow n++;
- ④ 如果 $n \leq 100$ ，表示数据还没处理结束，返回第②步，然后继续重复第②、③步；否则，计算结束，转下一步；
- ⑤ 输出结果 sum。

(2) 用流程图描述

流程图是用图形、箭头和文字说明来表示算法的框图。用流程图的优点是形象直观、通俗易懂，是描述算法的一种很好的工具，尤其是对于较复杂的问题，能将设计者的思路清楚地表达出来。

流程图一般由以下规定使用的基本框图组成，如图 1.2 所示。

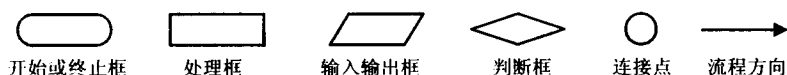


图 1.2 流程图基本框图

一般流程图由顺序、选择和循环三种基本结构组成，这三种基本结构可以相互嵌套，形成结构化的流程图。由结构化的流程图编出的程序也是结构化的程序。结构化程序的三种基本结构参见第 3 章 C 语言程序控制结构的相关内容。

用结构化的流程图的方式描述例 1.1 的算法，如图 1.3 所示。

3. 编写程序

根据以上确定的算法，用某种计算机语言实现这种算法，就是编写源程序。例如，用 C 语言编写源程序。编写源程序时要正确地使用语言，准确地描述算法，因此它是一个细致的工作，要经过不断的练习才能成为一个熟练的编程人员。

4. 调试程序

对于复杂的问题并不是编出来就能用的，要经过多次排错、调试及试运行，才可能得到正确运行的程序。

5. 整理文档

程序调试通过后，应该整理资料，编写程序使用说明书及程序所要求的软硬件环境等技术性文档。

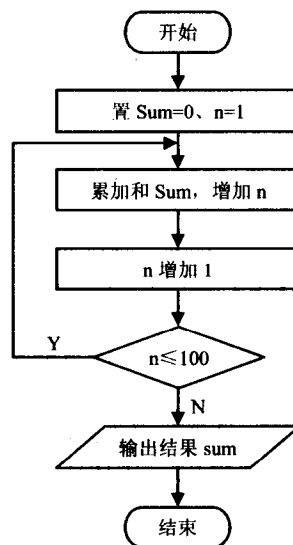


图 1.3 求 1 到 100 整数和的流程图

1.1.3 结构化程序设计思想

结构程序设计就是一种进行程序设计的原则和方法，按照这种原则和方法可设计出结构清晰、容易理解、容易修改、容易验证的程序。即结构化程序设计是按照一定的原则与原理，组织和编写正确且易读的程序的技术。结构化程序设计的目标在于使程序具有一个合理结构，以保证和验证程序的正确性，从而开发出正确、合理的程序。那么在编写结构化的程序时，应该采用结构化的程序设计思想。它包括两个方面的内容：模块化设计

和结构化编码。

1. 模块化设计

程序设计的第一步是分析问题，但是人脑的思维能力的局限性与处理问题的复杂性之间存在如下矛盾：当我们分析一个较小的问题时，可以考虑到问题的每一个细节；当我们分析一个庞大而又复杂的问题时，要想直接考虑到问题的每一步具体操作几乎是不可能的。但是，通常一个完整的系统，往往由成千上万个具体操作组成，即要实现几百行、几千行乃至几十万行的程序，该如何处理呢？

解决办法是采用“自顶向下，逐步细化和精化”的模块化设计方法。

这种模块化设计方法的基本思想是，将一个复杂的大任务逐层分解出许多易于理解和实现的、逻辑上相对独立的子任务，形成一棵功能树。然后分别实现各个子任务。这样，对一个大任务的处理就变成了对一个个小的子任务的实现，使问题变得简单了。

功能树上的每一个子任务在 C 语言中对应一个函数，也就是说每个函数是一个功能模块。一般的划分原则是每个功能模块完成一项相对独立的功能，模块之间的联系尽量少，以保证各模块的独立性。

2. 结构化编码

在设计好结构化的算法之后，还需进行结构化的编码，才能得到结构化的程序。用结构化的程序设计语言正确地描述算法，就是结构化编码。在 C 语言中有直接描述三种基本结构和基本操作的语句，用这些语句将算法表示出来，就是 C 的源程序。

我们学习 C 语言，不仅仅是学习 C 语言的语法，更重要的是学会设计算法和用 C 语言编码，从而掌握程序设计的一般方法。

1.2 C 语言简介

1.2.1 C 语言的发展

C 语言是在 B 语言的基础上发展起来的，最早可追溯到 ALGOL 60 语言，它是 1960 年由国际委员会设计和开发的。由于它是一种面向问题的高级语言，离计算机硬件太远，太抽象，不宜用来编写系统程序。1963 年剑桥大学和伦敦大学推出了 CPL 语言。CPL 语言在 ALGOL 60 的基础上接近硬件一些，但规模比较大，不易实现。1967 年剑桥大学推出 BCPL 语言。它具有结构化好，具有指针处理方式和直接访问内存地址等功能。1970 贝尔实验室对 BCPL 进行简化并设计出 B 语言，用 B 语言编写了第一个 UNIX 操作系统，在 PDP-11 上实现。但 B 语言过于简单，功能有限。1972 贝尔实验室又在 B 语言的基础上设计出 C 语言。C 语言既保持了 B 语言精练和接近计算机硬件的优点，又克服了 B 语言过于简单和数据无类型的缺点。

最初的 C 语言只是为描述和实现 UNIX 操作系统而设计开发的。但随着 C 语言的不断发展和应用的普及，C 语言可以在多种操作系统下运行，并且产生了各种不同版本的 C 语言系统。1983 年美国国家标准化协会（ANSI）根据 C 语言问世以来的各种版本，对 C 语言进行发展和扩充，制定了新的标准，称为 ANSI C。1987 年 ANSI 又公布了新标准

87 ANSI C。目前流行的C编译系统都是以它为基础的。现在微机上常使用的C语言系统有Microsoft C、Turbo C、Quick C等，本书C语言部分的实例和习题都是在Turbo C 2.0集成环境下运行的。

随着面向对象技术的发展，在C语言的基础上增加了面向对象的程序设计功能，于1983年由贝尔实验室设计了C++。C++语言的主要特点是全面兼容C语言和支持面向对象的编程方法。目前流行的软件开发工具以Visual C++6.0和C++Builder 5.0为代表。由于Windows软件的广泛使用，Visual C++6.0是程序员选择的主要编程工具，本书C++部分的实例和习题都是在Visual C++ 6.0集成环境下运行的。

1.2.2 C语言的特点

多年来，C语言经久不衰并不断发展，主要是由于它具有以下特点。

1. 结构化程序设计语言

C语言支持结构化程序设计的三类基本控制结构：顺序、选择和循环结构，如if……else语句、switch语句、while语句、do……while、for语句等。用函数作为程序模块以实现程序的模块化。而且C语言源程序是由多个函数模块组成的，每个函数功能独立，可以单独进行编译，生成的目标代码质量高，而且可以与其他语言连接生成可执行文件，因而它的可读性及可移植性好，调试和维护方便。

2. 语言简洁，使用方便、灵活

C语言只有32个关键字、45个标准运算符和9种控制语句。程序书写形式自由，区分大小写，主要用小写字母表示。语言的许多成分都是通过调用函数来实现的，使其程序简练而功能强大。

3. 运算符丰富

C语言把括号、赋值、强制类型转换等都作为运算符处理，使C的运算类型极其丰富，表达式类型多样化，从而实现在其他高级语言中难以实现的运算，并且简练易读。

4. 数据类型丰富

C语言数据类型有：整型、实型、字符型、数组、指针、结构体及共用体等。这些数据类型能实现各种复杂的数据结构运算。

5. 表达能力强

C语言允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。可见，C语言同时兼有高级语言和低级语言的功能。因此，C语言又被认为是中级语言，广泛应用于编写系统程序和应用程序。

6. 设有预处理功能

C语言的预处理是一种正文处理，是在编译之前对正文文件即源程序文件的再安排。它的预处理命令如#define和#include等，在编译源程序时首先被处理，提高程序的可读性和可移植性，并为调试程序提供方便。

当然，C语言也有不足。例如，C语言对数组下标越界不做检查，由程序员自己保证程序的正确性；对变量的类型使用比较灵活，如整型与字符型以及逻辑型数据可以通用，这降低了安全性；运算符的优先级较多，不易记忆和掌握。因此，学习C语言要先学基本知识，再灵活运用编程技巧，最后熟练使用C语言。

1.3 C 语言程序基本结构及书写风格

1.3.1 C 语言程序的基本结构

为了说明 C 语言的基本结构，先给出一个简单的 C 程序实例。

例 1.2 编写一个 C 语言程序，在屏幕上显示 “Let's learn C program.”

```
/* The C example */  
#include <stdio.h>  
main()  
{  
    printf("Let's learn C program.\n");  
}
```

运行结果是在屏幕上显示一行文字：

Let's learn C program.

以上程序非常简单，但可以说明 C 语言程序的基本结构。

1. C 语言程序是由函数构成的

(1) 一个 C 语言程序是由一个或多个函数组成。

其中必须有一个用 `main()` 命名的主函数，一个 C 语言程序总是从 `main()` 函数开始执行。该函数在程序中的位置任意，可以放在前，也可以放在后或放在各函数之间。

(2) C 语言函数的分类。

C 语言函数分两类：用户定义的函数和库函数。本实例中所用到的 `main()` 是用户定义的函数；`printf()` 函数是库函数，即系统提供的标准函数，该函数是面向标准输出设备的格式化输出函数。

(3) C 语言函数的组成。

一个函数的结构是由函数说明部分和函数体两部分组成。

① 函数说明部分

包括函数名、函数类型、函数参数表，其中函数名和一对圆括号是必不可省的。本实例中 `main()` 函数的函数类型省略，参数表为空。

② 函数体

函数体是函数说明下面花括号 `{.....}` 内的内容。函数体一般包括变量定义和执行部分。本例中 `main()` 函数的函数体只有 `printf("Let's learn C program.\n");` 一个语句，其功能是输出字符串 “Let's learn C program.”，并将光标移到屏幕的下一行上。

一个函数体如果既没有变量定义，又无执行部分，则该函数为空函数。空函数的花括号不能省略。例如：

```
delay()  
{  
}
```

它是一个空函数，不执行任何操作，但可延迟整个程序的执行时间，起延时作用。

2. C 语言程序语句

一个 C 语言函数由若干条 C 语句组成，C 语句是完成某种程序功能的最小单位。每个