图书在版编目（CIP）数据

C 及 C++程序设计/张富编. —修订本. —北京：人民邮电出版社，2005.7
高等学校计算机教材
ISBN 7-115-13525-8

Ⅰ.C… Ⅱ.张… 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 064262 号

内 容 简 介

本书以 Turbo C++为依据，以C语言为起点，全面地介绍C++语言的程序设计基础和面向对象的程序设计方法。全书分为两大部分。第一部分介绍C语言基础；第二部分，介绍面向对象程序设计的概念和方法。

本书可作为高等学校的程序设计语言的教材或参考书，也可供从事计算机的技术人员参考。

高等学校计算机教材

C 及 C++程序设计

（修订本）

◆ 编　　张　富
责任编辑　滑　玉

◆ 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
北京铭成印刷有限公司印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：23.75
字数：573 千字　　2005 年 7 月第 2 版
印数：23 181 –26 150 册　　2005 年 7 月北京第 10 次印刷

ISBN 7-115-13525-8/TP・4720

定价：33.00 元

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

目　录

第1篇　C语言基础

第 1 篇　C 语言基础

C++是 C 语言的超集，或者说，C 语言是 C++的一个子集。作为 C++语言的基础，首先掌握好 C 语言，无疑会减少直接学习 C++程序设计中的许多困难。同时，C 语言还可以作为程序设计语言独立使用。因此，我们首先来学习 C 语言的程序设计。然后，在此基础上再学习 C++的面向对象的程序设计方法。

第 1 章　对 C 语言的初步认识

本章介绍程序设计及程序设计语言的有关概念，了解结构化程序设计的基本思想。介绍 C 语言的概况和 C 语言程序的基本结构和 C 语言程序的开发过程以及在 Turbo C++的集成开发环境下编译、连接和运行 C 程序的操作步骤。本章还介绍了 C 语言的基本词法。

通过本章的学习，读者可以建立起关于计算机程序、程序设计语言和程序设计等基本概念，并对 C 语言有一个初步了解，为进一步学习 C 和 C++语言程序设计打下基础。

1.1　程序与程序设计语言

1.1.1　程序、程序设计和程序设计语言

一般来说，程序是对解决或处理一个问题的方法步骤的描述。而计算机程序，则是用某种计算机能识别的语言工具所描述的解决问题的方法步骤。例如，有两个数据 *a* 和 *b*，它们的值分别为 1 和 2，求这两个量的和 *c*。此问题程序（方法步骤）可描述为：

```
a=1;
b=2;
c=a+b;
```

可以看到，这里是通过三个步骤，也叫做三条语句来完成的。它们的意义是：

（1）将数值 1 赋给 *a*；

（2）将数值 2 赋给 *b*；

（3）计算 *a*+*b* 的和并将结果赋给 *c*。

其中 a、b 和 c 在计算机的程序设计语言中，通常称做变量。

编制并记录解决问题的方法步骤的过程就是程序设计。在计算机技术中，将解决一个问题的方法和步骤叫做算法。进行程序设计时要使用计算机能识别的描述算法的工具，这个工具就是计算机程序设计语言。

人们最早使用的程序设计语言是二进制机器语言。二进制机器语言是由计算机硬件系统所决定的，每种计算机都有自己的一套用二进制数码表示的指令的集合，称为该计算机的指令系统，也称为计算机机器语言。显然，不同的计算机系统，其机器语言是不同的。

用机器语言设计程序是相当烦琐、费时和费力的工作。为了减轻程序设计人员的工作量和难度，提高程序设计的效率和质量，很快就出现了用简单的英文字母组合来代表烦琐的二进制指令的语言——符号语言。用符号编写的程序，计算机的硬件不能直接识别，需要把它翻译成机器语言，用来翻译的软件叫做汇编程序或汇编器，因此，符号语言又称为汇编语言。用汇编语言编写的程序叫做汇编语言源程序，简称源程序。源程序经过汇编后产生计算机能直接执行（运行）的机器语言程序。汇编语言与机器语言一样，也是依赖于机器的语言，不同的计算机系统，其汇编语言是不同的。因此，我们说二进制机器语言和汇编语言都是面向机器的程序设计语言。

计算机程序设计语言的进一步发展，出现了与具体的计算机硬件系统无关的，着重于描述解决问题的算法过程的语言。这种语言接近人类自然语言和数学公式的表达方式，且与机器的具体型号无关，称其为高级程序设计语言，简称高级语言。高级语言属于面向问题的语言，如 ALGOL，FORTRAN，Pascal 等。相对于高级语言，人们称二进制机器语言和汇编语言为低级计算机程序设计语言。

用高级语言编写的程序也叫做源程序。源程序需要有专门的翻译程序将其翻译成机器语言，计算机才能执行这个程序。高级语言的翻译过程有两种不同的方式：一种称为解释执行方式，其特点是翻译一句执行一句，完成这种翻译工作的程序叫做解释程序；第二种方式是将源程序全部翻译成机器语言程序后，才可以执行的编译方式。编译方式中的翻译软件叫做编译系统，或编译器。

1.1.2 结构化程序设计方法

计算机程序设计语言经历了由机器语言、汇编语言到高级语言的发展过程。在高级语言发展的初期，人们广泛使用的语言有 FORTRAN，ALGO，BASIC 等，这些语言的特点是以简单的语句序列构成程序。以语句序列为基础组织起来的程序，虽然简单，但对于大型软件来说，不便于管理和维护。为了解决这个问题，在 20 世纪 60 年代末开始提出结构化程序设计的概念，也就是将程序由语句序列结构转变为模块集合。在 20 世纪 70 年代这种结构化的程序设计语言（如 Pascal，C 等）得到了广泛地发展和广泛应用。

结构化程序设计方法的基本思想是，将任何复杂问题分解为若干较为简单的功能模块，每个模块中的任何逻辑问题再用少数几种基本结构（如顺序结构、选择结构、循环结构）加以描述。支持这种结构化的程序设计方法的语言称为结构化的程序设计语言。结构化的程序设计方法，主要是实现两个方面的问题：程序的模块化设计和结

构化编码。

模块化设计就是将一个复杂问题，自上往下逐步细化，形成多层次的多个相对比较简单的功能模块，这样就使问题的解决变得层次清晰简单容易了。所谓结构化编码就是对每个小模块中的每个逻辑功能用几种简单的基本结构进行描述。

结构化程序设计中采用的三种基本结构如图 1.1 所示，所有的程序代码都实现在这三种结构中。

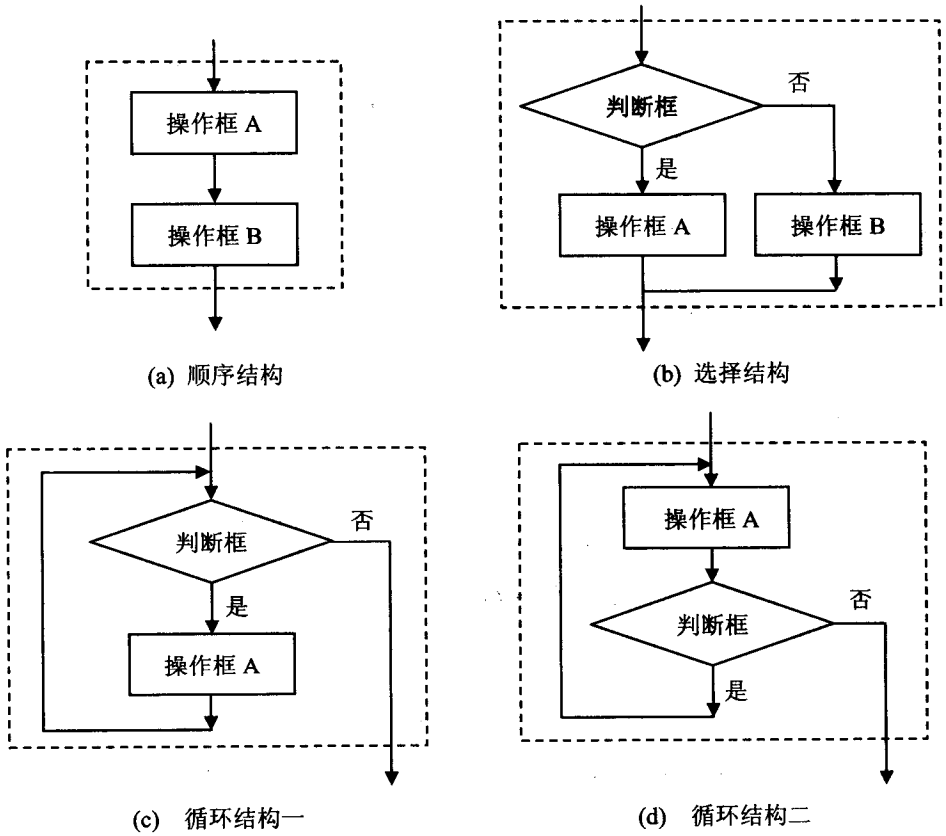

图 1.1　结构化程序设计的三种基本结构

图 1.1 中虚线框表示一种基本结构。矩形框表示一个简单操作或也是一个基本结构。菱形框表示对给定条件进行判断操作，它有两个可能的结果：条件成立(是)或条件不成立(否)，根据判断的结果执行不同的操作。图中的带箭头的实线表示流程的走向。

用上述框图表示流程的方法，是描述算法的一个很重要和常用的工具，今后还会在描述解决具体问题的算法时使用。

结构化的程序具有层次分明，易编、易读和易修改，从而有利于提高编程的效率和质量。这种方法在软件开发中至今仍占有重要的地位，支持这种结构化程序设计方法的语言，称为结构化程序设计语言。

进入 20 世纪 80 年代后，为了适应庞大而复杂程序的开发，出现了面向对象的程序设计方法和语言。然而它也吸收和继承了结构化程序设计的方法。

1.2 C语言及C语言源程序的基本结构

1.2.1 C语言

C语言是一种编译方式的结构化高级程序设计语言。1963年剑桥大学在ALGOL语言基础上开发出具有处理硬件能力的CPL（Combined Programming Language）语言。1967年该语言经简化演变为BCPL（Basic Combined Programming Language）语言。1970年美国贝尔实验室对BCPL进一步简化和对硬件处理能力的加强，推出了B语言（取自BCPL中的字母B）。1972年贝尔实验室在B语言的基础上，进一步完善和扩充，开发出C语言（取自BCPL中的字母C）。C语言最初用于实现UNIX操作系统。1977年Brain Kernighan和Dennis Ritchie编写了《The C Programming Language》一书，对C语言进行规范化，成为当时的标准。随着计算机的发展和普及，出现了一些不同的C语言版本。为统一标准，美国标准化协会（ANSI）制定了C语言标准，称为ANSI C。现在C语言已经成为广受欢迎的一种结构化程序设计语言。

目前，在微型计算机上使用的C编译系统有多种版本。本书将以“Turbo C++ 1.0”的编译系统为基础介绍C语言的程序设计。

C语言主要有下列一些特点。

（1）C语言是一种结构化的程序设计语言，它具有完整的程序控制语句。

（2）C语言是一种模块化程序设计语言，函数是组成程序的基本程序单位。

（3）C语言有丰富的数据类型和运算操作，使程序设计更为简单和方便。

（4）C语言提供了类似于汇编语言的低级语言动能，如直接访问内存的物理地址，对二进制数进行位操作等。使这种语言具有某些低级语言的特性，低级语言的优点，从而使这种语言更能接近硬件。

（5）语法结构简单，语句数目不多，但功能很强。所以，C语言简单易学且应用广泛。

1.2.2 C语言源程序的基本结构

C语言源程序，简称C程序，是建立在模块的基础上的，而基本的模块就是函数。因此，一个C程序是由一个或多个函数组成的。每个函数是完成一定功能的一段C语言程序。编写一个程序，首先是建立一个或若干个函数，然后把它们组织在一起，构成一个完整的C语言程序。在这些函数中必须有且只能有的一个函数就是main()函数，称为主函数。这就是说，一个C程序，至少要由一个函数组成，这个函数就是主函数。一个程序无论包含多少个函数，程序的运行总是从主函数开始，在主函数结束。

在C语言中，除了main()函数外，其他函数的函数名是用户选定的，称为自定义函数。每个函数都是包含一个或若干个语句并完成一定任务的独立程序单元。每个函数有一个写在函数名后面圆括号内的参数表。自定义函数不能像main()函数那样能独立运行，它们只能由主函数或其他函数激活（它也能激活其他函数），并开始运行。在计算机术语中，把这种激活叫做调用，所谓调用，简单地说，就是函数暂时中断本函数的执行，转去执行所调用的函数。前者称为主调用函数，后者称为被调用函数。被调用函数执行完自己的任务后，必

须返回原来的主调用函数，并从中断了的地方继续执行。这里发生了两个过程：调用和返回。可见，函数之间是互相调用和返回的关系。自定义函数可以互相调用，但不能调用主函数。

在最简单的情况下，C 函数有如下的格式：

```
函数名()
{
    函数体
}
```

函数名是用户为函数起的名字；函数名后跟圆括号，其内可含有参数，也可以没有参数；写在花括号内的是实现函数功能的程序语句，称为函数体。

例 1.2.1　一个最简单的 C 语言源程序。

程序由如下一个主函数组成：

```
main()
{
}
```

因为函数体不包含任何语句，所以该程序不执行任何功能，称它是空操作。

例 1.2.2　给例 1.2.1 的程序加入一个功能：在显示器上输出：hello!。

程序如下：

```
main()
{
    printf("hello!");
}
```

主函数的函数体由一个语句组成。C 语言规定每个语句必须以分号结束。分号表示一个语句的结束。语句

```
printf("hello!");
```

是一个输出语句。在这里它的功能是在显示器的屏幕上显示如下的字符串：

```
hello!
```

例 1.2.3　由两个函数组成的 C 程序。功能仍然是在显示器上输出：hello!。

主函数 main()调用另一个名为 hello()的自定义函数。hello()函数的功能是在显示器的屏幕上显示字符串："hello!"。

下面是源程序清单：

```
main()
{
    hello();
}
hello()
{
    printf("hello!");
}
```

程序运行后，在显示器的屏幕上显示如下的字符串：

```
hello!
```

主函数的函数体是C语言中的一个语句：

```
hello();
```

它的功能是调用函数hello()。执行调用的结果，使程序从主函数转去执行hello()函数。函数hello()的函数体是由一个C语句组成：

```
printf("hello!");
```

我们已经知道这个语句的功能是在显示器的屏幕上显示 hello!。该语句执行完毕后，返回主函数。由于主函数的所有语句已经执行完毕，于是程序结束运行。程序的执行从主函数开始，在主函数结束。

从以上的讨论，可以总结出以下几点。

（1）C 程序是由一个或多个函数组成的，其中必须有一个也只能有一个规定名为 main 的主函数。

（2）程序的执行总是从主函数开始，在主函数结束。其他函数是通过调用来执行的。

（3）主函数可以调用任何非主函数，非主函数之间可以互相调用，但不能调用主函数。

（4）函数体是完成函数功能的一组C语言的语句，每个语句完成一个小功能，并以分号作为一个语句的结束标志。

1.2.3 C语言的基本语句

C语言源程序是由函数组成的，其中包括主函数和自定义函数。函数的功能是靠C语句组成的函数体来实现的。因此，语句是C语言中最基本的成分，没有语句也就没有了程序。在例1.2.2和例1.2.3的简单程序中，出现了两个语句：调用函数语句和输出语句。学习C语言实现各种功能的语句，是本课程的重要内容之一，是设计程序的基本工具和基础。

C语言的基本语句主要有以下几种：

（1）数据定义语句；

（2）赋值语句；

（3）函数调用语句和返回语句；

（4）输入语句和输出语句；

（5）流程控制语句。

1.3 C语言的基本词法

构成C语言的最小单位是字符，由字符可以构成词类，再由词构成各种语句。有关C语言的字符和词类的规定，就形成了C语言的词法。这一节介绍词法方面的有关规定。

1.3.1 C语言的字符集

在C语言程序中允许使用的所有基本字符的集合，称为C语言的字符集。C语言的字符集采用的是ASCII（American Standard Code for Information Interchange）字符集。其中包括：

（1）52 个大写和小写的英文字母；

（2）10 个数字；

（3）如表 1.3.1 所示的 33 个键盘符号；

（4）如表 1.3.2 所示的转义字符集。转义字符总是由反斜杠开始，后跟一个或几个字符，用来表示控制代码或特殊符号。转义字符的具体应用，将在以后的章节中介绍。

附录 3 给出 ASCII 基本字符集的编码表。

表 1.3.1　　键盘符号

符　号	说　明	符　号	说　明	符　号	说　明
	空格符号	!	惊叹号	“	双引号
#	井号	$	美圆号	%	百分号
&	与符号	'	单引号	(	左圆括号
)	右圆括号	*	星号	+	加号
,	逗号	-	减号	.	小数点
/	正斜杠	:	冒号	;	分号
<	小于号	=	等号	>	大于号
?	问号	@	a 圈号	[	左方括号
\	反斜杠	]	右方括号	^	异或号
_	下划线号	`	重音号	{	左花括号
|	或符号	}	右花括号	~	波浪号

表 1.3.2　　转义字符集

符　号	说　明	符　号	说　明	符　号	说　明
\n	回车换行符号	\r	回车符号	\'	单引号
\t	tab 符号	\f	换页符号	\\	反斜杠
\v	垂直制表符号	\a	响铃符号	\ddd	1～3 八位进制数 ddd 对应的符号
\b	左退一格符号	\"	双引号	\xhh	1～2 位十六进制数 hh 对应的符号

1.3.2　标识符

用 C 字符集中的字符组合，可以为程序中的各种对象起名字。这些名字统称为标识符。例如常量名，变量名，函数名等都是标识符。例 1.2.3 中为函数起的名字“hello”就是一个标识符。C 语言对如何构成标识符做了如下的规定。

（1）标识符是由字母和下划线开头的数字、字母、下划线组成的一串符号。例如：

abc2，a2bc，a_2，_xq

都是合法的标识符，而

2a，a.2

则是不合法的。

（2）一个字母的大写和小写看作是不同的字符。例如：

abc，aBc

是两个不同的标识符。

（3）一个标识符可以由一个或多个字符组成，但 ANSI C 规定，标识符的长度不得超过 32 个字符。

1.3.3 保留字

保留字又称关键字，是 C 语言编译系统使用的、具有特定语法意义的一些标识符。这些标识符用户不能作为自己的标识符使用，所以把这些标识符称为保留字。C 语言中的保留字有：auto，break，case，char，continue，const，default，do，double，else，enum，extern，float，for，goto，int，if，long，register，return，short，signed，sizeof，static，struct，switch，typedef，union，unsigned，void，volatile 和 while 等。

1.3.4 C 语言的词类

C 语言的词类主要有以下几种。

（1）常量：在程序运行中值不发生改变的数据。

（2）变量：存放值可变化的数据。

（3）运算符：加工数据的运算符号。

（4）表达式：由常量、变量、函数及运算符组成的式子。

（5）保留字：编译系统使用的、具有特定语法意义的标识符。

上述词类将在以后的章节详细讲述。

1.4 源程序的编译和 C 语言的集成开发环境

1.4.1 C 程序的开发过程

开发一个 C 语言程序，要经过以下四个阶段：

（1）编辑源程序文件；

（2）编译源程序；

（3）程序连接；

（4）运行程序。

1. 编辑 C 语言源程序

编写源程序就是程序设计人员用 C 程序设计语言描述解决某问题的过程和具体实现的方法。这样写出的程序叫做 C 语言源程序。源程序以文件（File）的形式存储在计算机的软盘

或硬盘中，通常它是一种文本文件。所谓文本文件，就是以 ASCII 码存储的文件，它可以用任何文本编辑软件编写。

文件要有文件名，文件以其文件名在磁盘中存储和与其他文件相区别。文件名由两部分组成：（基本）文件名和扩展名。其书写格式为：

（基本）文件名.扩展名

按 C 语言编写的源程序，其文件扩展名通常为 c。例如，可以将例 1.2.1 中源程序以文件名 sample_1.c 存入计算机的硬盘。

2. 编译源程序

计算机系统只能认识和执行用机器语言编写的程序，不能理解用 C 语言或其他非机器语言编写的程序。所以，源程序必须翻译成机器语言程序。翻译是通过一个称为编译器（Compiler）的软件实现的。首先它对源程序进行语法检查，如果发现错误，就会显示错误的位置和错误的性质并终止编译。这时，用户需要对源程序进行再编辑，修改源程序文件中的错误。然后，重新进行编译。这个过程反复进行直到编译器认为没有发现错误为止。源程序通过编译后，产生一个目标文件。目标文件的文件名就是源程序文件的文件名，但扩展名为 obj。例如，源文件 sample_1.c 经编译后产生目标文件 sample_1.obj。目标文件由计算机的机器指令和其他一些二进制信息组成，它仍不能由计算机直接执行。

3. 连接程序

由编译系统中称为连接器（Linker）的软件，将目标文件和编译系统的系统函数库连接生成可执行的机器语言程序，这一过程称为连接。连接器在连接过程中也要对程序进行语法检查，如果发现错误，则给出相应的错误信息并终止连接。这时，程序设计人员要再次对源程序文件作相应修改，重新进行编译，重新进行连接，这个过程要一直进行到连接成功为止。目标文件经成功连接后，最终产生一个计算机直接可执行的程序文件。可执行程序文件以源程序文件名存储，但扩展名为 exe。例如，目标文件 sample_1.obj 经连接后产生可执行文件 sample_1.exe.

4. 运行程序

运行程序，即执行程序。通过运行程序，实现程序的功能。如果程序运行不正常或不能得到预期的结果，还需要从检查源程序开始，重复上述过程，直到程序运行正确为止。

运行可执行文件，既可以在 DOS 状态下进行，也可以在 C 语言系统提供的集成开发环境下进行。

图 1.2 说明了 C 程序的编译和连接两个过程。

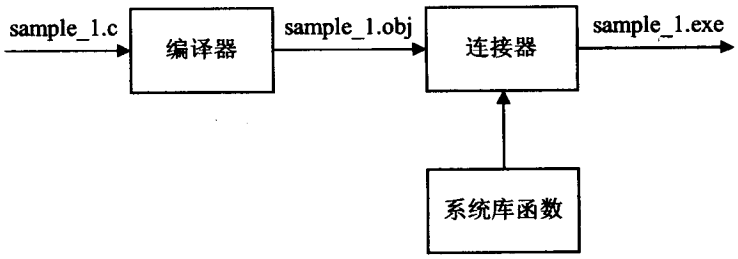

图 1.2　C 源程序的编译和连接过程

1.4.2 C语言的集成开发环境

为完成上述的程序开发过程，C语言系统提供两种独立的操作方式。一种是传统的命令行方式，就是在DOS状态下，用户直接通过键入命令进行工作；另一种是在集成开发环境（Integrated Development Environment，IDE）方式下操作。本节简要介绍IDE的使用。

IDE的优点，就在于它将源程序的编辑、编译、连接、调试和运行等多种操作集成在一个平台之上进行，从而为用户创造了一个非常方便的良好的工作环境。本节以Turbo C++的IDE为例，简要介绍在IDE下开发C程序的方法。更详细的内容可参考相关的手册。

1. IDE的启动

在启动之前，首先要安装Turbo C++系统软件。关于安装方面的细节，这里就不介绍了。系统是在操作系统DOS环境下运行的。在C++系统装入之后，为了进入IDE，只需在DOS状态下键入如下的命令行：

TC <回车>

系统被启动后，屏幕显示如图1.3所示的集成开发环境。

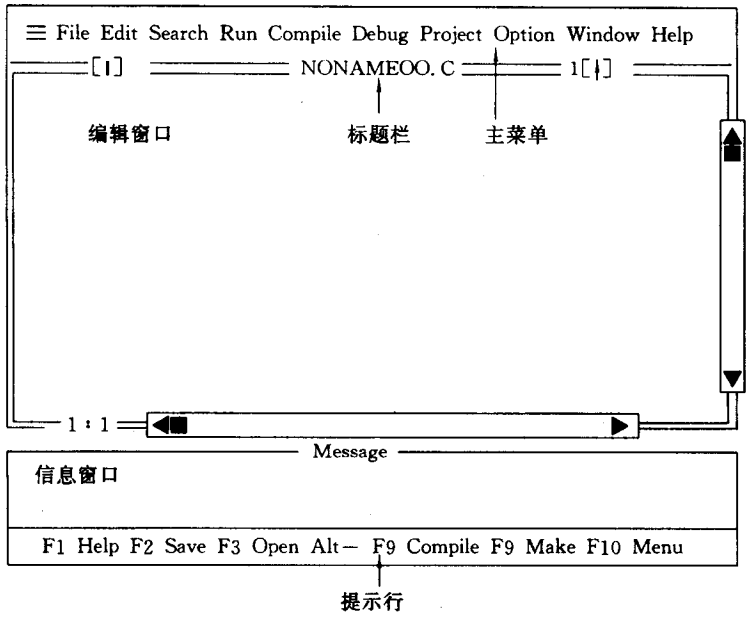

图1.3 Turbo C++的IDE

由图1.3可以看到，集成开发环境最上面的一行是主菜单。所有的操作命令都可以从这个菜单中或它的子菜单中找到。所以，即使读者记不住许多操作命令的快捷键，也没有关系，只需用鼠标选择所要执行的操作命令和输入必要的信息即可。

主菜单下面是程序的编辑区。用户在这里输入和编辑自己的C源程序。编辑区的顶部显示程序文件名，编辑区的底部显示当前编辑字符（光标处）所在的行号和列号。

编辑区的下方是信息区/输出区。在这里可以看到编译和连接过程中的信息，包括编译系统给出的错误信息。输出区则显示程序运行时输出的结果。

信息区的下面是一行提示信息，称为提示行。

2. 输入和编辑源程序

源程序的输入和编辑是在编辑区进行的。这里可能有两种情况：一是第一次输入程序，二是源程序文件已经在磁盘中。在第一种情况下，用户需要在编辑窗口直接用键盘逐个字符地输入源程序。如果想要编辑的源程序文件已在磁盘中，通过下面的操作，将文件读到编辑窗口：

选择主菜单中的 File；

择 File 下拉菜单中的 Open；

选中或输入所需的文件名并单击 OK 按钮。

系统提供丰富的编辑命令，使用户能够轻松地编辑源程序。程序编辑完毕，一般要立即存盘，这个操作可以通过选择主菜单的 File 及其下面的 Save 或 Save as 来实现。

3. 编译源程序文件

编译源程序的步骤是：

（1）将待编译的源程序输入编辑区；

（2）选主菜单中的 Compile；

（3）选 Compile 下拉菜单中的 Compile to OBJ。

如果源程序文件中没有语法错误的话，编译的结果是在磁盘中产生一个扩展名为.obj 的中间代码文件（目标程序）。

4. 程序连接

产生了 OBJ 文件后，就可以进行程序的连接了。连接的操作方法是：使用主菜单 Compile 下面的 Make EXE File 命令。如果没有错误的话，将产生一个扩展名为.exe 的可执行文件。

5. 运行可执行文件

运行程序的命令是主菜单中 Run 下面的 Run 命令。程序运行中的输出，将显示在屏幕的输出区。如果程序运行的结果产生了不合逻辑的结果，这可能意味着程序有错误。应该检查程序，找出错误，再进行相应的编辑、编译、连接和运行。

在 C++的集成开发环境下，上述的编译、连接和运行三个过程，也可以通过一次操作完成。操作方法是：执行主菜单中 Run 下面的命令 Run。这时，三个过程依次完成。当然，在三个过程中的任何阶段，只要系统发现了程序中的语法错误，都会立即在信息区给出错误信息，并停止进一步的操作。

小　结

本章主要介绍了以下几个问题：

（1）程序与程序设计的基本概念，包括结构化程序设计的基本思想；

（2）C 语言源程序的基本结构；

（3）C 语言的基本词法；

（4）C 源程序的编译和集成开发环境及其使用。

通过本章的学习，应该了解什么是结构化的程序设计，C 语言程序的基本构成和编译连接的过程，能初步使用 C 语言系统的集成开发环境编辑源程序、编译和执行简单的 C 程序。

习　题

1-1　什么是计算机低级语言，什么是计算机高级语言？

1-2　编译系统的作用是什么？说明由源程序到产生可执行文件的过程。

1-3　什么是结构化的程序设计？

1-4　结构化程序设计中，有哪几种基本程序结构？

1-5　说明C语言的特点。

1-6　说明C语言程序的基本构成。

1-7　什么是标识符？C语言对标识符有哪些规定？

1-8　在C语言集成开发环境（IDE）中，运行例1.2.1、例1.2.2和例1.2.4中的3个简单C程序。

第 2 章　基本数据类型、操作符和表达式

计算机处理的基本对象是数据。变量和常量则是程序的最基本的数据形式，将它们用操作符（也称为运算符）连接起来，便构成了表达式。本章介绍 C 语言中关于变量、常量、操作符和表达式的语法规则。这些语法规则是 C 语言的基本要素。

2.1　数 据 类 型

C 语言有丰富的数据类型，数据类型是按数据的存储形式来分的。在 C 语言中，数据的类型分为基本数据类型、构造类型、指针类型和空值类型。对于每一种数据类型的数据，又有分成几种不同类型，如图 2.1 所示。

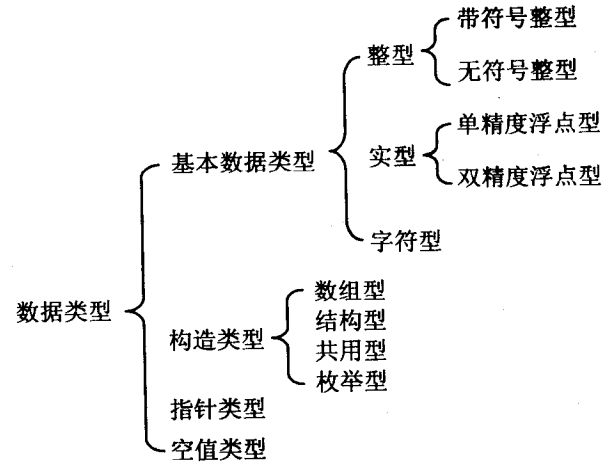

图 2.1　C 语言的数据类型

数据类型不仅决定了数据的存储形式，也确定了相应的取值范围和可进行的运算。通常将整型和实型统称为数值型。构造类型是由若干数据类型组合在一起构造成的复杂数据类型。指针类型可以表示数据的存储地址。空值类型表示没有数据值。

各种类型数据又可分为常量和变量。常量是程序运行中其值不能改变的数据。每个变量有自己的名字，叫变量名，变量用来存储在程序运行中其值可以发生变化的数据，一般用于存储原始数据、中间计算结果和最终计算结果等。

变量名应是合法的 C 标识符，用户在给变量起名字时要遵循标识符的规则。关于标识符的规则在上一章已经介绍。有一点要注意的是，用户定义变量名时，不要以下划线“_”开头，因为 C 语言系统本身使用的变量名多是用下划线开头的。

例如，下面是一些合法的变量名：

```
axb,  a_by, as2
```

本章对各种基本数据类型的常量和变量进行介绍。其他数据类型将在以后的章节中陆续讨论。

2.2 整型数据

2.2.1 整型常量

在C语言中使用三种不同进位制的整型常量，它们是：

（1）十进制数：例如，13，-15，0等；

（2）八进制数：八进制数的书写方法是在数字前加一个数字0，例如，015，-013，017，00等；

（3）十六进制数：十六进制数的书写方法是在数字前加一个0x，例如，0x0，-0x15，0xaf等。

整型常量在微机上一般占用两个字节的长度。所以，无论是十进制、八进制还是十六进制数，它们的数值的取值范围都是十进制的-32768～+32767。

为了扩大整型数据的取值范围，C语言还提供了一种长整型常量。长整型常量在计算机中占用4个字节，相应的取值范围扩大到-2147483648～+2147483647。长整型常量的书写方法是在数字的末尾加上一个字母L（或小写的l）。例如，15L，012L，0x1fL等都是长整型常量。

相对于长整型常量，占内存两个字节的整型常量叫做短整型常量，简称整型常量。

2.2.2 整型变量

整型变量在计算机中占两个字节（即16bit位），取值范围为从-32768～+32767(-2^{15}～$2^{15}-1$)。用以说明整型变量的关键字（也称为数据类型符）为int。

例如语句：

```
int a,b;
```

说明变量a和变量b是整型变量。在C语言中，这是一条定义变量数据类型的语句，并以分号结束。定义了变量a，意味着在内存中为变量a分配了两个字节的空间，用来存储变量a的数据。定义变量数据类型语句的一般格式为：

数据类型符　变量名1，变量名2，.....;

C语言规定，每个变量在使用前，都必须先定义。

在关键字int前加上修饰符，可改变整型变量的所占位数和取值范围。下列4种修饰符可以用来修饰整型变量：

（1）signed　　带符号的整型变量；

（2）unsigned　无符号的整型变量；

（3）long　　长型整型变量；

（4）short　　短型整型变量。

用这些修饰符修饰整型变量后，整型变量在计算机中所占位数和取值范围见表2.2.1。

表 2.2.1

数据类型符	所占位数(bit)	取 值 范 围
int	16（2 字节）	−32768～+32767
signed int	16（2 字节）	−32768～+32767
unsigned int	16（2 字节）	0～65535
short int	16（2 字节）	−32768～+32767
long int	32（4 字节）	−2147483648～+2147483647
unsigned long int	32（4 字节）	0～4294967925

以上含有修饰符的类型关键字中的“int” 是可以省略的。类型符 int 在没有修饰符的情况下，就是带符号的。因此，signed int 中的修饰符也是可以省略的。另外，short int 和 int 也是没有区别的。

signed int 和 unsigned int 的区别，就在于对数据的最高（二进制）位解释的不同。对于有符号的数据，其最高位是符号位，而对于无符号数据来说，最高位是可以存储数据的。因此，在存储位数不变的情况下，无符号数据的取值范围大了一倍。

C 语言的整数取值范围是很有限的，应用时要注意变量的取值不要超过允许的范围，即所谓的溢出问题。有符号的整型变量值超过其最大整数时，自动转到其负数的最大绝对值开始计数。如果是负数超过其最大绝对值，便从最大正数开始计数。无符号整型变量值超过其最大数时，从零开始计数。例如：定义了如下 3 个整型变量：

```
signed int      snum_1;
signed int      snum_2;
unsigned int    usnum_3;
```

3 个变量的值分别为（最大绝对值）：

```
snum_1=32767;
snum_2=-32768;
usnum=65535;
```

如果将上面的 3 个变量分别加 1、减 1 和加 1，便发生溢出，相应的结果分别为：

```
snum_1+1=-32768
snum_2-1=32767
usnum_3 +1=0
```

在程序设计中要避免发生数据的溢出，否则，将会得到不正确的计算结果。

2.3　实 型 数 据

实型数据也称做浮点数，是一种带小数点的数。现分别对浮点型常量和浮点型变量介绍如下。