

■ 高等院校信息技术应用型规划教材

C语言程序设计

主 编 张俊林

副主编 李清霞 林显宁 江涛 宗维玉



清华大学出版社

■ 高等院校信息技术应用型规划教材

C语言程序设计

主 编 张俊林

副主编 李清霞 林显宁 江涛 宗维玉

清华大学出版社
北 京

内 容 简 介

本书通过编程基础、程序界面设计、成绩等级判定程序、单科成绩统计程序、数组、函数、使用指针优化程序、结构体与共用体、文件、位操作和综合应用 11 个教学项目,循序渐进地介绍了利用 C 语言完成顺序结构程序设计、分支程序设计、循环程序设计及模块化程序设计的过程。每个项目由“知识学习”“技能训练”“课外提升”和“习题”4 个环节构成。在“技能训练”和“习题”部分融入了计算机等级考试(二级 C)的基本内容。为了便于教与学,本书配有丰富的教学资源,包括课件、课程标准、实训指导、习题解答和源代码等。

本书内容通俗易懂,实用性强,不仅可以用作应用型高校电子信息类各专业 C 语言程序设计课程的教材,还可以用作全国计算机等级考试(二级 C)复习教材和参考书。

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/张俊林主编. —北京:清华大学出版社,2017

(高等院校信息技术应用型规划教材)

ISBN 978-7-302-46153-1

I. ①C… II. ①张… 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 013766 号

责任编辑:王宏琴

封面设计:常雪影

责任校对:刘 静

责任印制:刘海龙

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

课件下载: <http://www.tup.com.cn>, 010-62770175-4278

印 装 者:北京泽宇印刷有限公司

经 销:全国新华书店

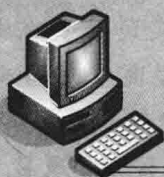
开 本:185mm×260mm 印 张:19.25 字 数:441 千字

版 次:2017 年 3 月第 1 版 印 次:2017 年 3 月第 1 次印刷

印 数:1~2000

定 价:39.80 元

产品编号:072059-01



前言

C语言是现代最流行的通用程序设计语言之一。它既具有高级程序设计语言的优点,又具有低级程序设计语言的特点;既可以用来编写系统程序,又可以用来编写应用程序。C语言是目前国内外广泛使用的计算机程序设计语言,是软件开发人员必须掌握的基础语言,也是理工科专业学习和研究活动的首选语言。

20世纪80年代以后,国际高等教育界逐渐形成一股潮流,那就是普遍重视实践教学,强化应用型人才培养。国内高校近年来纷纷地在教育教学改革的探索中注重强化实践环境,因为人们越来越清醒地认识到,实践教学是培养学生实践能力和创新能力的重要环节,也是提高学生社会职业素养和就业竞争力的重要途径。随着应用型高等教育的快速发展,对于C语言程序设计课程来说,迫切需要一套适合应用型高校教育教学改革,重视实践,强化应用,符合应用型高校学生学习特点的教材。

本书依据应用型高校教学改革和精品资源共享课程建设的需要,根据应用型高校学生学习特点和认知规律,由在教学一线多年从事程序设计课程教学和教育研究的教师编写而成。在编写过程中,课题组教师将长期积累的教学经验和体会融入知识系统中,力求做到通俗易懂。在内容编排上,采用案例驱动模式,通过由基础学习案例到技能训练案例的讲授、实现,将C语言基础知识和编程技能以及计算思维方法,根据程序开发的需要重新选取、序化,并将全国计算机等级考试大纲的内容融入其中,实现了将“知识学习”和“技能训练”融为一体,将知识传授转变为学生更愿意接受的做任务、项目的过程。

本书以培养学生的实际编程能力为基本目标。学生经过一学期的系统训练后,能基本掌握C程序设计语言的编程思想、思维方法;能应用C语言独立编写简单的程序;能应用C常用函数编写程序界面;能应用C语言中的数组、函数、指针、结构体、位运算和文件,编写高质量的程序;使学生遵循程序设计规范,养成良好的程序设计习惯。在整个教学过程中,采用VC++ 6.0作为实训环境,强调培养学生动手实践能力,掌握调试程序的方法,通过调试理解C语言程序运行的过程以及C语言的语法规则,为后续的课程学习、计算机等级考试及其他应用做好充分的准备。

本书的突出特点包括:通过典型案例和综合应用中的一个真实项目,实现C语言程序设计知识的学习和技能的训练;针对重视实践、强化应用的教学要求,以“学习案例”作为整个教学内容的编写线索,在重视学生技能训练的同时,兼顾其对理论知识的需求;通过“课外提升”部分的内容,既方便教师因材施教,也方便学生查阅相关内容。

本书始终以“重视实践、强化应用”为导向,在案例讲解过程中,突出问题的解决过程,引导学生遵循分析问题→设计算法→编写代码的步骤,学习程序设计的方法和技巧,突出

实际的编程能力。在习题的设计上,不仅满足等级考试的需求,而且强化实践动手的能力。本书内容通俗易懂,由浅入深,突出重点,重在应用,不仅可作为应用型高校 C 语言程序设计课程的教材,还可作为备考全国计算机等级考试(二级 C)的教材和参考书。

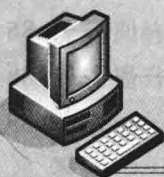
本书由张俊林主编,李清霞、林显宁、江涛、宗维玉任副主编。项目 1、项目 2 和项目 7 由张俊林编写,项目 3 和项目 4 由林显宁编写,项目 5 和项目 6 由江涛编写,项目 9、项目 10 和项目 11 由李清霞编写,项目 8 由宗维玉编写。由于编者水平有限,本书难免存在不当之处,恳请广大读者和专家批评指正。

本书在编写和出版的过程中,得到了清华大学出版社和编者所在学校的大力支持及帮助,在此表示诚挚的感谢。

本书在编写过程中参考了相关书籍和文献,还引用了一些网络资料,在此对所参考、引用资料的作者和机构表示衷心的感谢!

编 者

2017 年 1 月



目 录

项目 1 编程基础	1
知识学习	1
1.1 认识 C 语言	2
1.2 C 语言程序开发过程	7
1.3 基本数据类型	7
1.4 运算符和表达式	18
技能训练	25
课外提升	29
总结	37
习题	37
项目 2 程序界面设计——顺序结构程序设计	41
知识学习	41
2.1 算法基础	42
2.2 C 语言基本语句	45
2.3 格式输入/输出函数	48
2.4 字符输入/输出函数	57
2.5 顺序结构程序设计	59
技能训练	61
课外提升	65
总结	71
习题	72
项目 3 成绩等级判定程序——分支程序设计	77
知识学习	77
3.1 if 语句	78
3.2 switch 语句	82
技能训练	83
课外提升	86

3.3 分支程序应用	86
总结	88
习题	89
项目4 单科成绩统计程序——循环程序设计	93
知识学习	93
4.1 while 语句	94
4.2 do-while 语句	95
4.3 for 语句	96
4.4 循环嵌套	97
4.5 continue 语句和 break 语句	99
技能训练	100
课外提升	102
4.6 goto 语句	102
总结	103
习题	104
项目5 数组	109
知识学习	109
5.1 一维数组	110
5.2 二维数组	114
5.3 字符数组	119
5.4 字符串处理	124
技能训练	128
课外提升	137
总结	140
习题	141
项目6 函数——模块化程序设计	147
知识学习	147
6.1 函数概述	149
6.2 函数的定义	151
6.3 函数的调用	153
6.4 函数的参数	157
6.5 嵌套调用和递归调用	159
6.6 局部变量和全局变量	162
6.7 变量的存储类别	166
6.8 内部函数和外部函数	169

技能训练.....	171
课外提升.....	178
总结.....	185
习题.....	185
项目 7 使用指针优化程序	195
知识学习.....	195
7.1 指针	195
7.2 指针与数组	200
7.3 字符串与指针	207
7.4 函数与指针	209
技能训练.....	213
课外提升.....	217
总结.....	224
习题.....	224
项目 8 结构体与共用体	226
知识学习.....	226
8.1 结构体	227
8.2 结构体变量	228
8.3 结构体数组	231
8.4 结构体与指针	234
技能训练.....	236
课外提升.....	238
总结.....	240
习题.....	241
项目 9 文件	243
知识学习.....	243
9.1 文件概述	245
9.2 文件指针	246
9.3 文件的打开和关闭	247
9.4 文件的读/写.....	248
技能训练.....	253
课外提升.....	255
总结.....	257
习题.....	259

项目 10 位操作	263
知识学习	263
10.1 位运算符	263
技能训练	266
课外提升	268
10.2 位域	268
总结	270
习题	271
项目 11 综合应用	273
11.1 主菜单的显示	274
11.2 各功能模块的实现	276
附录 A ASCII 码对照表	284
附录 B 运算符优先级及结合性	285
附录 C C 语言常用库函数	287
附录 D C 语言编译错误信息	292
参考文献	298



项目 1

编程基础

➔ 学习目标

- (1) 掌握 C 语言程序的结构。
- (2) 掌握 C 语言程序的开发过程。
- (3) 熟悉 C 语言集成开发环境,并能熟练使用 VC++ 6.0 编写、运行简单的 C 语言程序。
- (4) 掌握 C 语言中的基本数据类型:整型、实型和字符型。
- (5) 熟练运用运算符和表达式,理解每种表达式的求值过程。

知识学习

项目任务: 用 C 语言编写程序,输出“这是我的第一个 C 语言程序!”。

源程序代码

```
/* 第一个 C 语言程序 */  
#include <stdio.h>  
void main()  
{  
    printf("这是我的第一个 C 语言程序!");    //输出“这是我的第一个 C 语言程序!”  
}
```

运行结果

这是我的第一个 C 语言程序!

程序分析: 这是一个简单的 C 语言程序。“/*”与“*/”及其之间的内容和“//”后面的内容是对程序的注释,用于对程序或语句进行说明,可以不输入,对程序的运行没有任何影响。

main()是主函数,所有的 C 语言程序都必须包含一个 main()函数。“{”和“}”之间是主函数的内容,是实现程序功能的主体,也称为函数体。C 语言程序从 main()函数的第一行语句开始执行,到 main()函数的最后一条语句结束。

“printf(“这是我的第一个 C 语言程序!”)”是用来在屏幕上显示“这是我的第一个

C 语言程序!”的 C 语言标准输出函数。printf() 函数是 C 语言提供的按指定格式进行标准输出的函数,其功能是输出由双引号括起来的字符序列,其中可以包括一些控制字符。

本项目主要学习: C 语言程序的结构及开发过程, C 语言中的基本数据类型、常量和变量的定义、运算符及其表达式等。

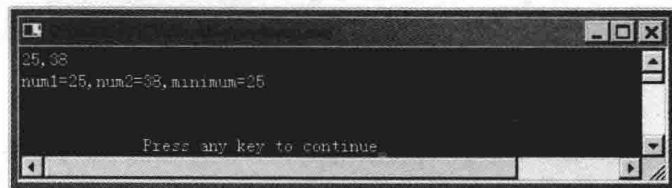
1.1 认识 C 语言

【学习案例 1-1】 键盘输入两个数,比较其大小,并输出两数中的较小值。

源程序代码

```
/* 求两个数中的较小值 */
#include <stdio.h>
int minimum(int x,int y);          /* 函数声明 */
void main()
{
    int num1,num2,min;
    scanf("%d,%d",&num1,&num2);
    min=minimum(num1,num2);        /* 自定义函数的调用 */
    printf("num1=%d,num2=%d,minimum=%d",num1,num2,min);
}
int minimum(int x,int y)           /* 用户自定义函数,求两个数中的最小值 */
{
    int m;
    if(x<y)
        m=x;
    else
        m=y;
    return m;
}
```

运行结果



程序分析: 本程序中有两个函数 main() 和 minimum()。在主函数 main() 中,“int num1,num2,min;”是变量说明,后面的三行是功能实现(执行)部分;minimum() 是自定义函数,用来实现求两个整数中的最小值。

scanf() 是 C 语言中用来从键盘输入数据的标准格式输入函数。

从学习案例 1-1 中,可以看出 C 语言程序的特点和书写规则。

1. C 语言程序的特点

(1) 在 main() 之前的行称为预处理命令(详见项目 6)。预处理命令还有其他几种。

这里的“include”称为文件包含命令,其意义是把尖括号“<>”或引号“”内指定的文件包含到本程序中,使之成为本程序的一部分。被包含的文件通常由系统提供,其扩展名为.h,因此也称为头文件或首部文件。C语言的头文件中包括各个标准库函数的函数原型。因此,在程序中调用一个库函数时,都应该包含该函数原型所在的头文件。本例使用了两个库函数:格式输入函数 scanf()和格式输出函数 printf()。

(2) 每个源程序可由一个或多个函数组成。

(3) 一个源程序不论由多少个文件组成,都有一个且只能有一个 main()函数,即主函数。

(4) 在C语言中,大、小写字母是有区别的。

(5) 每一个说明、每一条语句都必须以分号“;”结尾。但对于预处理命令,函数头和花括号“{}”之后不能加分号。

(6) 标识符和关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符,可不再加空格。

(7) 程序中可以加注释部分,有块注释和行注释两种方式。“/*”和“*/”之间的是块注释,块注释可以不在同一行上;“//”后面的是行注释。顾名思义,行注释只能在同一行上。

2. 书写程序时应遵循的规则

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) 在C语言程序中,每行可写一条语句或一个说明,也可写多条语句或多个说明。一般情况下,一行写一条语句或一个说明。

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

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便更加清晰,增强程序的可读性。

(4) C语言程序中的花括号必须成对出现。

(5) 在书写程序时,要习惯使用注释。

初学者在编程时应力求遵循上述规则,逐步养成良好的编程习惯。

1.1.1 C语言程序结构

通过项目任务和学习案例 1-1 的学习,可以看出,C语言程序的结构一般包括以下内容。

编译预处理行

全局变量说明

void main()

{

 局部变量说明

 程序段

}

返回类型 f1(形式参数说明)

//用户自定义函数 f1

```
{
    局部变量说明
    程序段
}
返回类型 f2(形式参数说明)           //用户自定义函数 f2
{
    局部变量说明
    程序段
}
:
返回类型 fn(形式参数说明)           //用户自定义函数 fn
{
    局部变量说明
    程序段
}
```

C 语言程序结构说明如下。

(1) C 语言程序中的变量在使用之前必须先定义其数据类型。未经定义的变量不能使用,且定义变量的语句必须放在可执行语句的前面。

(2) 程序段由多行语句组成。

(3) 函数定义时使用的参数为形式参数,其说明用于传值。

(4) 区分局部变量(函数内部定义的变量)、全局变量(函数外定义的变量)的用法含义。

(5) 以上示例中花括号内的内容统称为函数体。

(6) 用户定义的函数名由用户命名。

(7) 用户自定义函数可置于主函数之前,也可置于主函数之后。若在主函数之后,要在主函数之前对被调用函数进行函数声明。

1.1.2 C 语言的字符集

字符是组成程序设计语言的最基本的元素。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) 字母: 小写字母 a~z 共 26 个,大写字母 A~Z 共 26 个。

(2) 数字: 0~9 共 10 个。

(3) 空白符: 空格符、制表符、换行符等统称为空白符。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,编译程序时将被忽略。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,将增强程序的清晰性和可读性。

(4) 标点和特殊字符: 标点在程序中起分隔作用,一般称作普通分隔符。特殊字符包括注释符、转义字符(后面会专门介绍)等。

① 普通分隔符有如下几个。

“{”和“}”：大括号/花括号，成对出现在程序中，用来定义复合语句、函数体及数组的初始化。

“[”和“]”：方括号，成对出现在程序中，用来定义数组及引用数组元素。

“(”和“)”：小括号或圆括号，成对出现在程序中，用来分隔函数名与其后的实参或形参；用在表达式中，限定运算顺序。

“;”：分号，语句结束标志。

“:”：冒号，用来给语句添加标号，或与问号一起构成条件表达式。

“.”：圆点，用来分隔复合类型变量与其成员变量。

“,”：逗号，用来构成逗号表达式。

“?”：问号，与冒号一起构成问号表达式。

② 注释符有行注释符“//……”和块注释符“/*……*/”之分。其中，省略号“……”代表注释内容。

1.1.3 C 语言词汇

在 C 语言中使用的词汇分为以下六类。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。

C 语言标识符命名规则如下所述。

(1) 标识符只能是由字母(A~Z, a~z)、数字(0~9)、下划线“_”组成的字符串。

(2) 标识符的第一个字符必须是字母或下划线(即数字不能出现在标识符的第一个字符处)。

(3) 标识符不能使用 C 语言的关键字(关键字的说明见下文)。

例如，以下标识符是合法的：a、x、x3、BOOK_1、sum5；以下标识符是非法的：3s(以数字开头)、s * T(出现非法字符*)、-3x(以减号开头)、bowy-1(出现非法字符-(减号))、if(关键字不能作为标识符)。

在使用标识符时，还应该注意以下几点。

(1) 标准 C 语言不限制标识符的长度，但它受不同版本 C 语言编译系统限制，也受到不同机器的限制。例如，在某版本 C 中规定标识符前 8 位有效，当两个标识符的前 8 位相同时，则被认为是同一个标识符。

(2) 在标识符中，大、小写是有区别的。例如，TEACHER 和 teacher 是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义，但它是用于标识某个量的符号。因此，命名时应尽量有相应的含义，以便阅读理解，做到“见名知义”。

2. 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 语言的关键字共有 32 个，根据不同的作用，分为数据类型

说明符、控制语句关键字、存储类型关键字和其他关键字 4 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。如学习案例 1-1 中用到的 int, 还有后面要学到的 double、char 和 float 等, 共有 12 个, 分述如下。

char: 声明字符型变量或函数; double: 声明双精度变量或函数; enum: 声明枚举类型; float: 声明浮点型变量或函数; int: 声明整型变量或函数; long: 声明长整型变量或函数; short: 声明短整型变量或函数; signed: 声明有符号类型变量或函数; struct: 声明结构体类型、变量或函数; union: 声明共用体(联合)数据类型; unsigned: 声明无符号类型变量或函数; void: 声明函数无返回值或无参数, 声明无类型指针。

2) 控制语句关键字

控制语句关键字用于表示一条语句的功能, 如学习案例 1-1 中用到的 if else 就是条件语句的语句定义符。针对循环语句、条件语句、开关语句和返回语句, 共有 12 个关键字, 分述如下。

循环语句: for、do、while(用来构成 while 循环, 或与 do 一起构成 do-while 循环)、break、continue; 条件语句: if、else、goto; 开关语句: switch、case、default; 返回语句: return。

3) 存储类型关键字

存储类型关键字用于表示变量的存储类型, 共有 4 个, 分述如下。

auto: 声明自动变量, 一般不使用; extern: 声明变量是在其他文件中声明; register: 声明寄存器变量; static: 声明静态变量。

4) 其他关键字

const: 声明只读变量; sizeof: 计算数据类型长度; typedef: 用于给数据类型起别名(当然还有其他作用); volatile: 说明变量在程序执行中可被隐含地改变。

3. 运算符

C 语言中含有相当丰富的运算符(如 1.4 节所述)。运算符与变量、常量、函数一起组成表达式, 表示各种运算功能。运算符由一个或多个字符组成。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中, 分隔各个变量。空格多用于语句各单词之间, 作为分隔符。在关键字与标识符之间必须有一个以上的空格符作为间隔, 否则将出现语法错误。例如, 把“int a;”写成“inta;”, 编译器会把 inta 当成一个标识符处理, 其结果必然出错。

5. 常量

C 语言中使用的常量分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。后面将详细介绍。

6. 注释符

C 语言的注释符是以“/*”开头并以“*/”结尾的串; 或者是以“//”开头, 用在某一程序行的后面的串。在“/*”和“*/”之间或“//”后面的内容即为注释。程序编译时, 不对

注释做任何处理。注释可出现在程序中的任何位置,用来向用户提示或解释程序的含义。在调试程序时,对于暂不使用的语句,可用注释符将其暂时标注起来,跳过它,不做编译;待调试结束后,再去掉注释符。

1.2 C 语言程序开发过程

在 C 语言集成开发环境下,C 语言程序的开发过程分成四个步骤:编辑→编译→链接→执行,如图 1-1 所示。

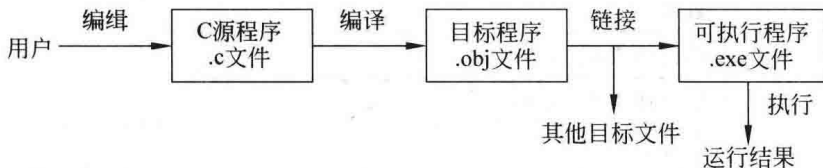


图 1-1 C 语言程序开发过程

1. 编辑

编辑即编写源程序代码。这是最基础的一步,即生成 C 语言的源文件(.c,必需)和可能的头文件(.h,非必需)。

2. 编译

通过编译工具,将编写好的代码转换为目标文件,即 .obj 文件。通过这一步,对文件内部及包含的头文件进行语法、语义的分析、检查。如果出错,必须返回到“编辑”步,对代码进行修改,直到没有错误为止。

3. 链接

将目标文件链接成可执行文件,即 .exe 文件。通过这一步,对文件直接的关联进行检查。如果出错,返回到“编辑”步,修改代码,直到没有错误。

4. 运行

这是最后一步,也是 C 语言程序的最终目的。

当运行结果与所期望的不符时,需要查明原因,修改代码,然后重新执行上述 3 步,直到程序运行结果正确。

1.3 基本数据类型

如前所述,程序中的各种变量都应先说明,后使用。对变量的说明包括三个方面,即数据类型、存储类型和作用域。

本节只介绍数据类型说明。其他说明在后续章节中陆续介绍。

所谓数据类型,是按照被说明量的性质、表示形式、占据存储空间的多及构造特点来划分的。在 C 语言中,数据类型分为基本数据类型、构造数据类型、指针类型和空类型四大类,如表 1-1 所示。

表 1-1 数据类型

基本类型	数值型	整型	短整型(short)
			基本整型(int)
			长整型(long)
		实型(浮点型)	单精度实型(float)
			双精度实型(double)
	字符型(char)		
	枚举型(enum)		
构造类型	数组		
	结构体类型(struct)		
	联合体类型(union)		
	文件类型(FILE)		
指针类型			
空类型(void)			

1. 基本数据类型

基本数据类型最主要的特点是：其值不可以再分解为其他类型。也就是说，基本数据类型是自我说明的。

2. 构造数据类型

构造数据类型是根据已定义的一个或多个数据类型用构造的方法来定义的。也就是说，一个构造类型的值可以分解成若干个“成员”或“元素”。每个“成员”都是一个基本数据类型，或又是一个构造类型。在 C 语言中，构造类型有数组类型、结构体类型及联合体（共用）类型等几种。

3. 指针类型

指针是一种特殊的，具有重要作用的数据类型，其值用来表示某个变量在内存存储器中的地址。虽然指针变量的取值类似于整型量，但它们是两种类型完全不同的量，不能混为一谈。

4. 空类型

在调用函数时，通常应向调用者返回一个函数值。这个返回的函数值具有一定的数据类型，应在函数定义及函数说明中给予说明。如果该函数没有返回值，则其返回类型为 void(空类型)。

1.3.1 常量和变量

对于基本数据类型的量，按其取值是否可改变，又分为常量和变量两种。在程序执行过程中，其值不发生改变的量称为常量，其值可变的量称为变量。它们可与数据类型结合起来分类。例如，分为整型常量、整型变量，浮点常量、浮点变量，字符常量、字符变量等。在程序中，常量可以不经说明而直接引用，变量则必须先定义后使用。