面向21世纪高等院校规划教材

C程序设计

姚国清 夏军宝 何勇强 主编

附赠
精美教学课件
使用方法见
本书前言

航空工业出版社

面向21世纪高等院校规划教材

C程序设计

主　编　姚国清　夏军宝　何勇强

航空工业出版社
北京

内容提要

本书为高等院校规划教材，主要讲述了使用C语言设计程序的方法。本书不仅对C语言的基本概念和基本知识进行了深入浅出的讲解，同时还将作者多年来在C语言教学过程中积累的经验和体会告诉了读者。在每章的最后，我们为读者提供了有针对性的类型多样的习题，另外，在本书最后，还为读者安排了大量综合练习。希望读者学练结合，逐步培养阅读程序和编写程序的能力。

本书语言浅显易懂、实例和习题众多，可以作为高等院校计算机专业和非计算机专业通用教材，也可供大专院校学生和程序爱好者自学使用。

图书在版编目（CIP）数据

C程序设计 / 姚国清，夏军宝，何勇强主编. —北京：航空工业出版社，2008.3

ISBN 978-7-80243-081-5

Ⅰ.C… Ⅱ.①姚…②夏…③何… Ⅲ.C语言—程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2007）第195229号

C程序设计

C Chengxu Sheji

航空工业出版社出版发行

（北京市安定门外小关东里14号 100029）

发行部电话：010-64978486 010-64919539

北京市科星印刷有限责任公司印刷 全国各地新华书店经售

2008年3月第1版 2008年3月第1次印刷

开本：787×1092 1/16 印张：21.5 字数：537千字

印数：1—5000 定价：32.00元

编者的话

1972 年，Dennis Ritchie 在贝尔实验室开发出 C 语言，在过去的 30 多年中，它已经成为最重要和最流行的编程语言之一。C 语言是一种融合了控制特性的现代语言，其设计使得用户可以自然地采用结构化编程和模块化设计，而这种做法使编写出的程序更可靠和易懂。C 语言是一种高效语言，用它生成的程序很紧凑且运行速度快。C 语言也是一种功能强大和灵活性好的语言，它允许访问硬件，可以操纵内存中的特定位。同时，C 语言的可移植性极好，在一个系统上编写的 C 程序经过很少修改或不经修改就可以在其他系统上运行。

20 世纪 90 年代以来，很多软件开发商转向用功能更强大的 C++语言来开发大型项目。C++语言在 C 语言的基础上加入了面向对象的工具，其意图是使语言来适应问题而非让问题适应语言。基本上可以把 C++看作 C 的超集，所以学习 C 的同时也可以学习到 C++的很多知识。不管面向对象的语言如何流行，C 语言在软件产业中依然是一种最重要的工具，特别是在含有大量数值计算的领域和嵌入式系统的开发中。所以在未来的若干年中 C 仍将保持强劲的势头。

C 语言最初并没有官方标准，Brian Kernighan 和 Dennis Ritchie 在 1978 年出版的 The C Programming Language 成为被广泛接受的一个标准，称为 K&R C 或经典 C。美国国家标准局（ANSI）于 1989 年通过了一个新标准（ANSI C），国际标准化组织（ISO）于 1990 年采纳了这个标准，称为 ISO C。这两个标准的内容是一致的，也被称为 C89 或 C90。ANSI/ISO 委员会从 1994 年开始又对 C 标准进行修订，并于 1999 年通过了 C99 标准，是目前 C 语言的最新版本。在新标准中 C 语言在与 C++充分兼容的同时仍然保持了自己的特征，使它继续是一种简洁、清楚和高效的语言。

本书力图以简短的篇幅介绍 C 语言的基本概念和基本语法，使读者通过学习可以具有初步使用 C 来解决问题的能力。我们也注意到，学习编程语言最重要的不是其语法和标准，而是编程思想的建立和使用语言的能力。所以在内容上我们贯穿了引导学习者建立正确思维模式的训练。

本书由姚国清、何勇强和夏军宝共同完成编写。其中第 0 章至第 4 章由夏军宝编写，第 5 章至第 7 章、第 12 章和附录 A 由何勇强编写，第 8 章至第 11 章和附录 B、附录 C 由姚国清编写，姚国清负责最后的统稿。

本书配有精美的教学课件，读者可到 **www.bjjqe.com** 下载，本书的下载号为 **802430815**。

编　者

2008 年 3 月

目 录

第 0 章　程序设计概论

用计算机解决问题包括两个概念上不同的步骤。首先应该构造一个算法或在解决该问题的已有算法中选择一个，这个过程称为算法设计。第二个步骤是用一种程序设计语言将该算法表达为程序。刚开始学习程序设计时将算法翻译成实际的编码过程会是较为困难的阶段。在学习了更多的有关程序设计的过程后，编码会很快变得简单，与此同时，随着要求解决的问题越来越复杂，算法设计会变得越来越复杂。所以在学习中要有意识的兼顾这两个过程。

本章介绍计算机程序设计的基本概念及发展。阅读本章读者将了解到程序设计中的一些基本概念以及了解 C 语言的概貌。

0.1　程序与程序设计语言

软件是计算机系统中不可缺少的部分。什么是软件？什么是程序？什么是程序设计语言？本节将给出解答。

0.1.1　程序

什么是软件？目前还没有一个精确的定义。但认为软件就是程序的概念肯定是错误的。目前比较公认的定义是由著名的软件工程专家 B.W.Boehm 提出的“软件是程序以及开发、使用和维护所需要的所有文档”，由此可见，程序只是完整软件产品的一个部分。

什么是程序？国标中规定“计算机程序是按照具体要求产生的适合于计算机处理的指令序列”。也就是说，程序是计算机可以识别和执行的操作表示的处理步骤。我国颁布的《计算机软件保护条例》更加明确提出“计算机程序是指为了得到某种结果而可以由计算机等具有信息处理能力的装置执行的代码化指令序列，或者可被自动地转换成代码化指令序列的符号化序列，或者符号化语句序列”。可以看出，程序既可以指能被计算机直接执行的指令序列（可执行文件），也可以是我们用程序设计语言编写的源程序，但源程序不能被计算机直接执行，必须通过程序设计语言翻译为可执行的指令序列。

0.1.2　程序设计语言

程序是由人来编写，程序的执行实际上反映了人的设计思想。随着计算机技术的不断发展，“编程”技术也在不断地发展，程序设计语言经历了几个不同的阶段。

1. 机器语言

众所周知，数据是以二进制形式存储在计算机中，我们所能感知的声音或图像进入计算机之前必须要经过编码转化为二进制。计算机能够执行的程序也不例外，也是以二进制代码表示指令序列。机器语言便是以二进制代码表示机器指令的一种语言，用机器语言写的程序能被计算机直接执行。

最早的计算机（如 ENIAC）便是采用机器语言来编写程序，“程序员”采用手工拨钮开

关和纸带打孔的方式向计算机输入程序，那时的程序员都是一流的科学家。下面的程序段便是使用 x86 计算机的机器语言编写的，功能是计算 1+1。

【例 0.1】使用机器语言计算 1+1。

```
10111000
00000001
00000000
00000101
00000001
00000000
```

很明显，用这样的方式编写程序非常麻烦，人类很难读懂，而且不同的计算机提供的机器语言不同，编写的代码很难通用。于是汇编语言应运而生。

2. 汇编语言

为了解决机器语言编程的困难，人们专门编写一种称为程序设计语言的程序，它提供一套语言规范，程序员可以采用该语言编写“程序”，然后由程序设计语言将“程序”翻译为计算机能够执行的程序。汇编语言便是最早的程序设计语言之一，用汇编语言编写 1+1 的代码如例 0.2 所示。

【例 0.2】使用汇编语言计算 1+1。

```
MOV AX,1
ADD AX,1
```

上面的两行符号，计算机并不能直接执行，需要借助汇编编译器将它翻译成例 0.1 所示的二进制机器代码。汇编语言屏蔽了枯燥的二进制代码，汇编语言用人类易懂的符号代替二进制代码，使得编程技术前进了一大步。但汇编语言与机器语言比较接近，多数符号基本上和一组二进制指令码直接对应，与计算机内部实现细节直接相关，编程的难度仍然较大。各种新的程序设计语言不断诞生发展，人们习惯把机器语言和汇编语言称为低级语言，而把之后的程序设计语言称为高级语言。

3. 高级语言

高级语言的出现是编程语言的一大进步，它屏蔽了机器的细节，语言更接近于自然语言和数学语言，给编程带来了极大的方便。据不完全统计，目前已经有超过 2500 种计算机语言，绝大多数是高级语言。利用 C 语言计算 1+1 的代码如例 0.3 所示。

【例 0.3】使用 C 语言计算 1+1

```
#inlcude <stdio.h>
void main()
{
    printf("%d\n",1+1);
}
```

也许你还不懂这些符号和单词，但很快就会学会这些内容。高级语言易学、易用、强大，具有一定的通用性。在例 0.3 中，如果想让计算机计算 20-4，你可能已经猜到，将程序中的 printf("%d\n",1+1);更改为 printf("%d\n",20-4); 就这么简单。回过头来，如果要用机器语言或

汇编语言来计算 20-4，恐怕你就不容易猜到了。

0.2　C 语言的历史和发展

高级语言门类繁多，C 语言是其中影响最大、寿命最长的语言之一。本书便是教授大家利用 C 语言编写计算机程序的。C 语言的应用非常广泛，普通计算机使用的 Windows 操作系统基本上是用 C 语言编写的，用来进行文档处理的 Word、Excel 也是基本上用 C 编写的。C 语言同时具有高级语言和低级语言的特点，因此也被称为“中级”语言，使得 C 语言具有其它语言无法相比的速度优势和灵活性，C 的语法规则简单、灵活，使得它屏蔽了汇编语言复杂、难以调试等缺点。下面介绍 C 语言产生发展的简单历史。

20 世纪 60 年代，贝尔实验室（Bell Laboratory）的 Ken Thompson 开始用汇编语言开发 UNIX 操作系统，当程序规模不断扩大时，他开始无法忍受汇编语言的难写、难读和难以维护，便开始在 BCPL 程序设计语言的基础上，将其改造成 B 语言，使得 B 语言更适合于开发 UNIX。1971 年，Dennis M.Ritchie 开始和 Thompson 合作开发 UNIX，M.Ritchie 主要负责改进 B 语言，后来便被命名为 C 语言。

1973 年 C 语言基本完成，Thompson 和 Ritchie 用 C 语言重写了 UNIX 系统，UNIX 操作获得了巨大的成功，C 语言也因此获得广泛的应用。

C 语言本身也在不断发展，1989 年，美国国家标准局（ANSI：American National Standards Institute）发布了一个完整的 C 语言标准，被称为 C89 或 ANSI C。1990 年，国际标准化组织（ISO: Internation Organization for Standardization）直接采用了 C89，有的资料上称为 C90，实际上 C90 和 C89 完全一样。1999 年，ISO 发布了最新的 C 语言规范，被称为 C99，是目前 C 语言标准的最高版本。

目前，有许多新程序设计语言正在兴起，如 C++、Java、C#等，它们借助面向对象程序设计思想，大大提高了开发效率，获得了程序员们的认可，称为程序设计语言中的三大主流阵营。在这样的形势下，仍然有一些程序员们在坚守 C 语言的阵营，他们追求着 C 语言的运行高效率和灵活性。

0.3　一个简单程序的诞生

在正式学习 C 语言编程规则之前，首先感受一下完整的 C 程序的概貌，以及程序设计语言的工作过程。

0.3.1　简单的 C 语言程序概貌

下面的示例 C 程序，完成计算两个整数中的最大值功能。

【例 0.4】计算两个整数的最大值

```
#include <stdio.h>
/* GetMax 函数求两个整数中的最大值，两个参数类型为
整型函数返回值类型为整数  */
int GetMax(int a, int b)
```

```
{
    if(a>b)                /* 如果a比b的值大，则返回a */
        return a;
    return b;
}
void main()
{
    int x, y, max;
    printf("Please input two integers: ");
    scanf("%d%d", &x, &y);                    /* 通过键盘输入两个整数 */
    max = GetMax(x, y);                       /* 求x、y中的最大值保存到max中 */
    printf("The max number is: %d\n", max);
}
```

例0.4就是一个简单但完整的C语言程序，现在你可能还读不懂其中每个符号所代表的含义，没关系，先对程序有一个轮廓的认识。运行程序时，屏幕上将显示一条提示信息：

```
Please input two integers:
```

要求用户通过键盘输入两个整数。假设用户输入33和-4，即

```
Please input two integers:33 -4↙
```

符号↙表示按下回车键，33和-4之间空一个空格，也可以每输入一个数后按一下回车（以后的示例均如此表示），输入后屏幕的提示结果为：

```
The max number is: 33
```

1. C语言程序由函数构成

函数（Function）是构成C语言程序的基本单位，例0.4由GetMax和main两个函数构成。每个函数由函数首部和函数体构成，GetMax的函数首部为“int GetMax(int a, int b)”，main函数的函数首部为“void main()”；函数体便是函数首部之后的一对花括号“{}”之间的内容。具体的有关函数的知识请参照第五章内容。

每个程序中有且只能有一个main函数，这个函数称为主函数，程序执行时，便从这个函数体“对应”的第一条指令开始执行。一个C程序中可以包含多个自定义函数，也可以一个没有，示例0.4中的GetMax函数便是自定义函数，main以外的函数均是自定义函数。

2. 对库函数的引用

编程时，如果所有的函数都要程序员自己编写，那工作效率一定很低，对于一些通用的功能，多数编译器均提供了一组标准库函数。库函数的使用极大提高了编程效率，程序员可以在自己的程序中调用库函数。

要调用系统提供的库函数，必须在程序中包含相应的头文件（Header files），包含头文件要用到编译预处理指令，示例0.4中所用的为：

```
#include <stdio.h>
```

stdio.h是最常用的需要包含的头文件，一般程序中均要包含这一头文件。要将计算结果

输出到屏幕上，需要调用库函数中的 printf 函数，从键盘让用户输入数据，需要调用库函数中的 scanf 函数，它们都需要包含 stdio.h；如果要调用数学函数，如计算正弦函数值的 sin 函数，则要包含 math.h（具体格式为：#include <math.h>）。

通常情况下，将头文件包含放在程序的最开始处。

3. 培养良好的书写习惯

C 程序的函数体部分由语句构成，语句都是以分号作为结束符（反过来不成立，出现分号的地方不一定是一条语句的结束），编译预处理指令（如头文件包含）不是 C 语句，所以不能在最后加分号，函数的定义前后也不能加分号。

C 的语句书写格式比较规范，可以在一行内书写多条语句，也可以将一条语句分开写在多行上。但为了保证程序的可读性和调试改错的方便，应该养成良好的代码书写规范和习惯。一般情况下，一行内只写一条语句，代码结构清楚，错落有致，希望大家在上机实践的时候，仿照书中的例题来书写程序，养成良好的习惯。

4. 注释也很重要

在例 0.4 中，/*和*/之间的内容便是注释，注释是程序员对自己所编写程序的注释性说明，可以采用中文也可以用英文，但一定要放在/*和*/之间，而且/和*之间不能有空格。编译器在将程序翻译为机器语言代码时，忽略所有的注释内容，也就是说注释对程序功能没有任何影响。但注释却是编写程序时不可缺少的内容，清晰的注释记录了程序员编写程序时的思路和设计思想，便于其他程序员或自己事后再来阅读这一段程序。规范化的程序设计提倡给程序加上必要的注释。

C 语言的注释可以有两种方式，上述是第一种方式，放在/*和*/之间，中间可以换行，也就是说可以有较多的注释说明。另一种方式最早是 C++语言引入的，目前大多数 C 编译器已经支持，ANSI C 标准也已经引入，也称为行注释，用 // 引导，// 之后到该行结束所有内容都是注释。如：

```
scanf("%d%d", &x, &y);              // 通过键盘输入两个整数
```

本教材中，根据需要两种注释都使用。

0.3.2　第一个程序的诞生

用高级语言编写的程序，计算机并不能直接运行，必须要借助编译程序将我们编写的代码翻译为用 0 和 1 表示的机器语言指令代码，才能真正在计算机中执行。用 C 语言开发软件都要经历“编辑→编译→链接→运行”的过程。

1. 编辑源代码

编辑阶段就是利用 C 语言编写实现特定功能的程序代码，通常将这一阶段的代码称为源代码（Source Code），源代码必须遵循高级语言的规范，同时也是人的智慧与思维的体现。本书的其余章节都是讲解如何写好 C 语言源代码。

一般将编写好的 C 语言源代码保存为文本文件，扩展名为.c，如例 0.4 的代码我们可以保存为 findmax.c。

2. 编译（Compile）

编译由编译器负责完成，编译器是一个事先开发好的软件，它专门用于将我们编写的源代码翻译为计算机能够理解的二进制目标代码。能将 C 语言源代码翻译为二进制码的的编译器有很多，常用的有 Turbo C、Microsoft C、Borland C 等。编译器是一种非常复杂的软件，不同的编译器在翻译代码时的策略也不完全相同，同样的代码经不同的编译器翻译后的可执行文件并不完全相同，所以在真正开始软件开发时，首先要了解自己所选择的编译器。本书选用 Turbo C 作为编译器，有关 Turbo C 的操作请参照附录的说明。

在编译阶段，编译器通常会发现源代码中不符合语言规范的语法错误，程序员需要修订这些错误并重新编译直到编译成功。如果你输入的 findmax.c 没有语法错误，编译后将生成 findmax.obj，我们称之为目标文件。

3. 链接（Link）

程序中除了自己编写的代码外，往往还需要调用库函数中提供的函数或者其他人编写的目标函数，链接过程便是将这些目标函数合并成有关完整的可执行文件。对于例 0.4 的 findmax.c，将生成 findmax.exe，findmax.exe 便可以在计算机上直接运行。

目前大多数编译器都提供了集成的开发环境（IDE：Integrated Development Enviroment），只要点击一下菜单，便可以连续完成编译、链接过程（编译出错则不能完成链接），最后生成一个可执行文件，一般不会觉察到链接过程的存在。

4. 运行

链接成功后得到的便是一个可执行程序，可执行程序便可以复制给别人去执行应用。可执行程序生成后便和源代码脱离关系，可执行程序的执行并不需要源代码的存在。很多初学 C 语言程序设计的人，总是认为是自己编写的源代码再执行（一点 Turbo C 中的 Run 菜单项，程序便执行），实际上应该是你编写的源代码被编译器翻译成的可执行程序在运行，当运行有错误时，重新修改代码，重新翻译，重新运行。

程序在运行时，将二进制指令代码和要处理的数据加载到内存中才能执行，要想了解可执行文件的加载过程还需要进一步的学习，而源代码只是存储在磁盘上的一个普通文件。由此可见，真正执行的不是源代码，而是编译链接结果加载到内存后的二进制代码。

5. 小结：程序员、编译器、测试人员、用户

在软件开发过程中，有非常重要的四个角色：程序员、编译器、测试人员和用户。程序员是 C 语言源代码的撰写者，根据功能需求，按 C 语言语言规范编写源代码。源代码需要借助编译器进行翻译，我们需要购买一个 C 语言编译器（如 Turbo C）安装到自己的机器上，执行编译器提供的翻译功能，生成一个可执行程序。生成的可执行程序需要经过测试人员进行运行测试。测试无误后，将软件打包分发、销售给广大用户使用，用户是软件的使用者和受益者，用户通过软件解决实际的问题。

当然在学习 C 语言编程的时候，我们既是程序员，同时也是测试人员和用户，但我们不能混淆这几个截然不同的角色。编写源代码时我们是程序员；编译时需要借助编译软件完成；程序编写完成后要反复运行，查看有无功能上的缺陷或错误，此时我们便是测试员和用户。

程序员、测试员和用户应该是在不同的环境和场合下编写或测试程序，但在实际学习时，多个环节几乎都是在同一台机器的 Turbo C 环境下完成的，容易造成错觉。

大家在编写程序时，要养成良好的编码习惯，编写的程序要界面友好，输入输出要有提示说明，用户输入错误要有检测和提示。你编写的程序不一定只是你在运行，你的成果可能会拷贝到别人的机器上执行，如果在输入数据时没有提示，用户很难知道程序到底要干什么？在例 0.4 中，在要求用户输入两个整数之前，通过 printf("Please input two integers: "); 向屏幕上打印一行提示，非常直观，在上机实践时一定要遵循这一编码习惯。

本章小结

C 语言是目前应用比较广泛的编程语言之一，本章简单介绍了程序及程序设计语言的基本概念，介绍了 C 程序设计语言的发展历史。

C 语言编写的源程序不能被计算机直接执行，必须借助编译器将源代码编译链接为可执行程序，才能被计算机识别执行。通过一个简单的示例，展现 C 语言源代码的概貌和编写 C 程序的基本流程。

复习题

一、选择题

1.【　】是构成 C 语言程序的基本单位。

A）函数　　B）过程　　C）子程序　　D）子例程

2. C 语言规定，必须用【　】作为主函数名。

A）function　　B）include

C）main　　D）stdio

3. 在 C 语言中，每个语句和数据定义是用【　】结束。

A）句号　　B）逗号

C）分号　　D）括号

二、简答题

1. 指令、程序和软件有什么区别？
2. 机器语言、汇编语言、高级语言各有什么特点？
3. 简述 C 语言编程的基本过程。
4. 简述 C 程序的基本结构。
5. 程序员、编译器、测试人员、用户在软件开发过程中各自的职责是什么？

第1章　C程序设计基础知识

程序的最后执行需要将可执行代码及待处理的数据加载到内存中，计算机内存中要为可执行代码留出保存空间，还要为待处理的数据留出相应的空间，很多场合下，要处理的数据占用的空间要比可执行代码占用的空间要多。现实生活中有自然数、整数、小数等概念，不同类型数的精度是不一样的。在计算机内存中，也要区分所保存数据的类型，如整数、浮点数、字符数据等，不同类型数据所占用的内存空间（即占用的字节数）是不一样的。

大多数高级语言在源代码中都要用声明语句对程序中将要处理的数据进行变量定义，其目的就是为不同类型的数据开辟不同的存储空间，便于进行处理与存储。

本章从数据基本类型开始，介绍变量定义、基本运算符及表达式语法，是C语言编程中的基础知识。

1.1　变量

C语言的数据类型比较丰富，整体上可划分为基本类型、构造类型以及空类型，基本类型包括整型、浮点型（进一步区分为单精度和双精度类型）、字符型、枚举类型和指针类型，构造类型有数组、结构体和共用体。本节将重点讨论最基本的整型数和实型数以及定义整型和实型变量的方法。

1.1.1　有关内存的基础概念

内存由能存储信息的介质组成，它们划分为众多的存储单元，每个存储单元可以存储一个字节或8个二进制位，可以将存储单元理解为由8个微小开关构成，每个开关可以表示一个二进制位，将指令和数据送入存储器，只不过是打开一些开关而合上另外一些开关。内存包含众多的存储单元，为了识别它们，从0开始顺序为它们编号，编号称为某个存储单元的地址，表示这一存储单元在内存中的位置（很多时候用相对位置的概念）。存储单元的地址反映的是位置信息，存储单元的内容是8个二进制数。

磁盘、光盘、CPU中的寄存器等存储器的基本原理与内存相似，只不过各自的存储介质不同而已。

1.1.2　变量的引入

程序运行过程中，经常需要保存用户输入的数据、计算的中间结果以及最终计算结果，因此在C语言中引入了变量的概念，用来存放值可以变化的数据。顾名思义，变量对应的数据可以被读出来参与计算，也可以将计算结果写入某个变量对应的存储空间。

C语言中每个变量要具备3个要素：变量的名称、变量的数据类型、变量的值。

1. 标识符命名规则

C程序中经常要为函数、变量等起一个符号化的名称，这些符号称为标识符，标识符的

命名必须遵循一定的规则。

（1）标识符只能由英文的字母、数字和下划线构成，标识符不能以数字打头，第一个字符只能是字母或下划线。

（2）C 语言的关键字不能作为标识符，C 语言总共有 32 个关键字，见表 1-1 所示。需要注意的是，不同的 C 编译器，支持的关键字略有不同。

（3）C 语言中的标识符区分大小写，如 Sum 和 sum 是两个不同的标识符。

（4）在 C 语言中，标识符长度可以不受限制，但旧的 ANSI C 规定，只有前 6 个字符有意义，也就是说编译器只看标识符的前 6 个字符，新的 C 和 C++编译器打破了这一限制，但不同的编译器还是有一个长度限制。

（5）标识符命名要养成良好的习惯和一致的风格，尽量做到“见名知义”，尤其当程序代码增多时，不好的标识符命名将使得代码难以阅读，也很难维护。这一条不是硬性规定，但应该从一开始就要养成良好的习惯。

表 1-1　C 语言的关键字表

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

2. 变量的名字

变量的名字可以根据需要由程序员指定，除了要遵循标识符的规定，还应该注意以下的规则。

（1）在同一个函数体内，不要定义同名的两个变量。在 Turbo C 中，所有变量的定义语句必须放在函数体的最开始处，变量定义要遵循“先定义后使用”的原则，否则会出现编译错误。

（2）尽量避免定义与库函数中函数名同名的变量，如 printf、scanf 等，虽然不会出现语法错误，但容易造成混乱。

下面的变量名是合法的变量名：total_money、_MyCar、AverageScore、class_1，也是比较好的变量命名，变量名中最好用英文单词表达变量的含义，有的系统推荐使用下划线分隔单词，有的推荐使用单词首字母大写来区分，尽量使用统一的风格，避免多种方法混杂在一起。下面的变量命名虽然合法，却不是很好的命名习惯：a1、a2、xyz 等。而下面的变量命名则是不合法的：1_class、total+moneny、My=Car、int。

3. 变量的定义方法

变量定义的基本语法格式为：

变量类型　变量名列表；

其中变量类型确定了变量所要存储数据的类型，如整型、实型等，变量名列表列出多个需要定义的变量的名称（也可以一次只定义一个变量），变量名之间用逗号（一定不要使用中文逗号）分隔。