

C YUYAN BIANCHENG SHILI

C 语言编程实例

主 编 王 琪
副主编 方向红 赵 骞



中国电力出版社
CHINA ELECTRIC POWER PRESS

C YUYAN BIANCHENG SHILI

C 语言编程实例

主 编 王 琪
副主编 方向红 赵 骞



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书将蓝牙小车作为教学组织引领,开展各章节、模块的教学,在教学的过程中把C语言编程、VC++软件操作、单片机外围电路搭建与调试、CAD机械制图、激光切割零件制作等知识与技能点串成知识、技能链,并以此涵盖产品的设计、安装和调试,突出“产品引领法”的基本特征:产品引领、知识系统、激发兴趣、内容实用、突出能力、边学边用。

本书可作为中职院校电类专业师生的教材,也可作为企业电子技术从业人员的培训用书,还可供电子技术初学者参考。

图书在版编目(CIP)数据

C语言编程实例 / 王琪主编. —北京:中国电力出版社, 2015.8

ISBN 978-7-5123-7497-3

I. ①C… II. ①王… III. ①单片微型计算机—C语言—程序设计 IV. ①TP368.1②TP312

中国版本图书馆CIP数据核字(2015)第062294号

中国电力出版社出版发行

北京市东城区北京站西街19号 100005 <http://www.cepp.sgcc.com.cn>

责任编辑:杨淑玲 责任印制:蔺义舟 责任校对:郝军燕

航远印刷有限公司印刷·各地新华书店经售

2015年8月第1版·第1次印刷

787mm×1092mm 1/16·8印张·192千字

定价:25.00元

敬告读者

本书封底贴有防伪标签,刮开涂层可查询真伪

本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究

前 言

近年来,随着大规模集成电路的发展,单片机的发展异常迅猛。单片机芯片的产量以每年 27% 以上的速度递增。实践证明单片机技术已经成为 20 世纪 90 年代后最活跃的新一代电子应用技术。随着我国关于产业结构调整政策的出台,传统工业向自动化、智能化、网络化方向发展已成为必然趋势。在这种情况下,本科、高职、中职、职业高中等不同层次的学校都开设了单片机课程,在职技术人员也由于工作需要,迫切希望掌握单片机的开发和应用技术。为此浙江省机电高级技工学校相关教师根据多年的教学经验,开发引进了这套 Arduino 单片机实验开发系统,重点面向教学,集实验和开发于一体,为提高单片机原理、C 语言编程课程的教学质量,提供一种经济、实用、方便的解决方案。

本书将蓝牙小车作为教学组织引领,开展各章节、模块的教学,在教学的过程中把 C 语言编程、VC++ 软件操作、单片机外围电路搭建与调试、CAD 机械制图、激光切割零件制作等知识、技能点串成知识、技能链,并以此涵盖产品的设计、安装、调试,从而突出“产品引领法”的基本特征:产品引领、知识系统、激发兴趣、内容实用、突出能力、边学边用。

需要说明的是在 Arduino 单片机教学平台的开发过程中,得到了学校、学院领导的大力支持以及课题组相关教师的大力帮助,在此,谨对他们致以衷心的感谢!同时在此对上海欧兰智能科技有限公司的鼎力支持表示感谢!由于编者水平有限,不妥之处在所难免,敬请读者批评指正。

编 者

目 录

前言

第1章 标准C语言程序与VC 6.0	1
第2章 标准C语言程序基础	6
2.1 标准C语言程序基本框架	6
2.2 库函数头文件调用	6
2.3 基本数据类型	7
2.4 常量与变量	8
2.4.1 常量	8
2.4.2 变量	9
2.5 主函数main()	9
2.6 顺序结构及其应用	10
2.6.1 格式输出函数	10
2.6.2 格式输入函数	12
2.7 运算符符号与表达式	12
2.7.1 表达式	12
2.7.2 赋值运算符及赋值表达式	12
2.7.3 算术运算符及算术表达式	13
2.7.4 关系运算符及关系表达式	13
2.7.5 逻辑运算符及逻辑表达式	13
2.7.6 自增自减运算符	13
2.7.7 符号优先级	14
2.8 分支结构及其应用	15
2.8.1 if语句	15
2.8.2 switch语句结构	17
2.8.3 分支结构程序应用实例	19
2.9 循环结构及其应用	22
2.9.1 for语句	23
2.9.2 while语句	23
2.9.3 do...while语句	24
2.9.4 循环结构应用实例	25
2.10 数组及其应用	30
2.10.1 一维数组	32
2.10.2 字符串数组及其处理	33

2.10.3 数组应用实例	35
2.11 子函数及其应用	38
2.11.1 函数的定义	39
2.11.2 函数的调用	40
第3章 Arduino 单片机开发套件简介	42
3.1 Arduino 单片机开发板硬件说明	42
3.2 Sensor Shield 扩展板硬件说明	42
3.3 数字量与模拟量	43
3.4 Arduino IDE 编程软件使用说明	44
第4章 Arduino C 语言程序应用	47
4.1 Arduino C 语言程序基本框架介绍	47
4.2 Arduino C 语言程序与标准 C 语言程序的异同	47
4.3 单片机外围电路及其元器件	48
4.4 常用库函数介绍	51
4.5 数字口输出控制	52
实验一: LED 闪烁实验	53
实验二: 流水灯实验	54
4.6 数字口输入控制	57
实验三: 按键控制 LED 灯	58
4.7 模拟量输出控制	60
实验四: 呼吸灯实验	60
4.8 模拟量输入控制	61
实验五: 环境光检测实验	62
实验六: 电位器调光实验	62
4.9 舵机的控制	63
实验七: 电位器控制舵机实验	63
第5章 机械设计与机械制图基础	66
5.1 AutoCAD 软件基本操作	66
5.1.1 AutoCAD 的界面	66
5.1.2 AutoCAD 的文件操作	67
5.1.3 AutoCAD 的绘图说明与设置	67
5.2 机械设计中常见几何特征的绘制	69
5.2.1 绘制直线	69
5.2.2 绘制矩形、正多边形	70
5.2.3 绘制圆、圆弧、椭圆、椭圆弧	71
5.2.4 绘制倒角、圆角	73
5.2.5 图形操作	74
5.2.6 绘制齿轮	80
5.3 常用标准件简介	81

5.4 机器人常用结构设计方案·····	82
第6章 激光切割机的使用 ·····	84
6.1 激光切割机软件操作说明·····	84
6.2 亚克力材料的切割·····	90
6.3 亚克力材料的雕刻·····	92
第7章 机器人综合实验——蓝牙小车制作 ·····	94
7.1 蓝牙小车机械结构设计与制作·····	94
7.2 蓝牙小车硬件搭建方案·····	95
7.3 蓝牙小车程序流程分析·····	96
7.4 蓝牙小车整体软件编写·····	103
附录1 ASCII 码表 ·····	108
附录2 字符串处理函数库 ·····	109
参考文献 ·····	122

第1章 标准C语言程序与VC 6.0

C语言是一种计算机程序设计语言,它既具有高级语言的特点,又具有汇编语言的特点。它由美国贝尔实验室的 Dennis M. Ritchie 于 1972 年推出,1978 年后,C 语言已先后被移植到大、中、小及微型机上,它可以作为工作系统设计语言编写系统应用程序,也可以作为应用程序设计语言编写不依赖计算机硬件的应用程序。它的应用范围广泛,具备很强的数据处理能力,不仅仅是在软件开发上,各类科研都需要用到 C 语言。它适于编写系统软件,三维、二维图形和动画,具体应用如单片机及嵌入式系统开发。

自 C 语言诞生以来,从大型主机到小型微机,C 语言被广泛应用于系统软件和应用软件的开发,随之也衍生了很多个版本。1989 年,美国国家标准局第一次为 C 语言制定标准,被称为 ANSI X3.159—1989“Programming Language C”,俗称 ANSI C 或 C89 标准。1990 年后,国际标准化组织(简称 ISO)为 C 语言多次修订标准,并于 2011 年 12 月 8 日最新修订并公布了 C 语言标准 ISO/IEC 9899:2011,俗称 C11 标准。本课程上机使用的 Microsoft Visual C++ 6.0 正是基于 C11 标准的 C 语言开发环境,因此本书将符合 C11 标准的 C 语言称为“标准 C 语言”,以便与 Arduino IDE 开发环境下的 C 语言做区分。

VC 6.0 软件基本操作如下:

1. 打开 VC 6.0 软件

打开“Microsoft Visual C++ 6.0”,将出现如图 1-1 所示窗口。

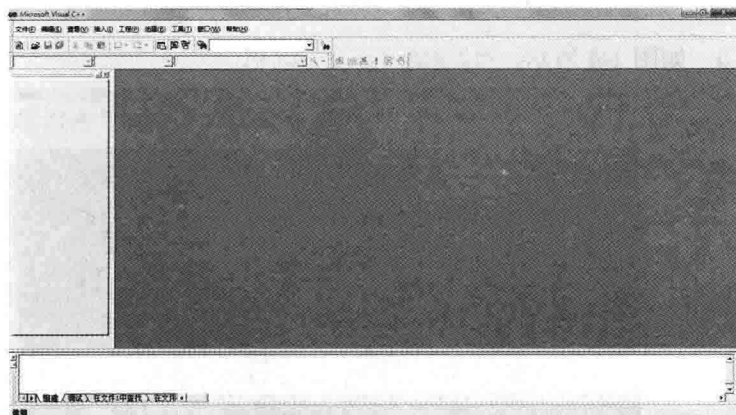


图 1-1 Microsoft Visual C++ 6.0 界面

2. 新建 win32 控制台工程

单击一级菜单“文件”→“新建”→选择弹出框的“工程”选项卡→选择“Win32 Console Application”→在“工程名称”对话框(图 1-2)内输入工程名称。



图 1-2 工程选择对话框

选择“一个空工程”→单击“完成”→单击“确定”(图 1-3)。

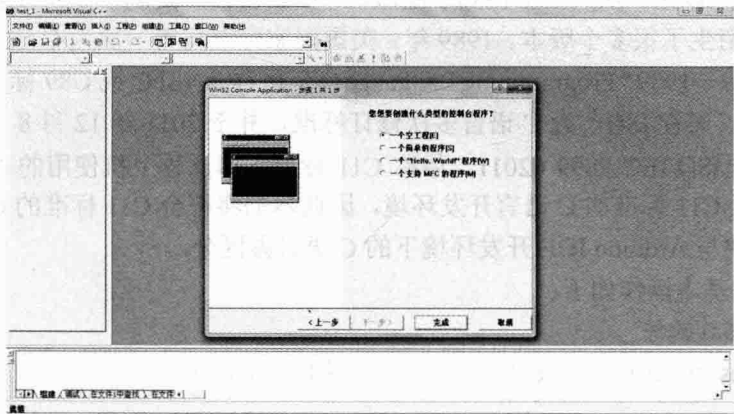


图 1-3 工程模板选择对话框

工程创建成功，如图 1-4 所示。



图 1-4 工程创建成功界面

3. 新建 C 程序文件

单击一级菜单“文件”→“新建”→选择弹出框的“文件”选项卡→选择“C++ Source

File” → 在“文件名称”对话框内（图 1-5）输入文件名称。



图 1-5 新建 C 程序文件

C 程序文件创建成功之后可以编写程序，如图 1-6 所示。

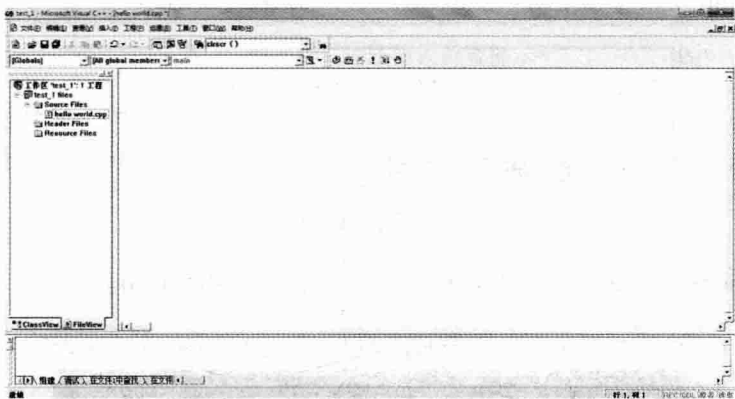


图 1-6 C 程序编辑界面

4. 在程序编辑框内（图 1-7）输入如下程序



图 1-7 简单的“Hello World!”程序

```
#include <stdio.h>

void main( )
{
    printf("hello, world!\n"); //打印输出"hello world"字样
}
```

5. 编译、生成、运行程序

编译程序 **Compile** ：单击之后，生成.obj 文件，一般用于检查程序中是否存在语法错误，如图 1-8 所示。

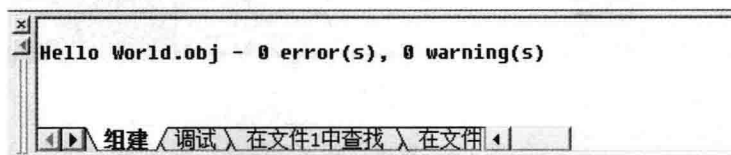


图 1-8 检查语法错误

生成程序 **Build** ：单击之后，生成.exe 文件，如图 1-9 所示。

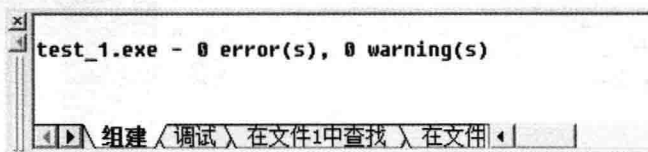


图 1-9 编译文件

运行程序 **BuildExecute** ：用于查看程序运行结果，如图 1-10 所示。

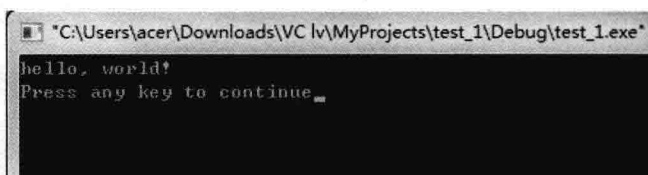


图 1-10 打印输出界面

6. 保存程序

选择“文件”菜单中的“保存”命令，将源程序保存到指定的文件夹中。

7. 调试程序

调试程序是程序设计中一个很重要的环节。程序中的错误一般分为程序语法错误和程序设计上的逻辑错误，VC 6.0 编译器只能找出语法错误，而逻辑错误则需要程序员手动查找。

如果程序中存在语法错误，那么编译时会在输出窗口中给出错误提示，如图 1-11 所示。

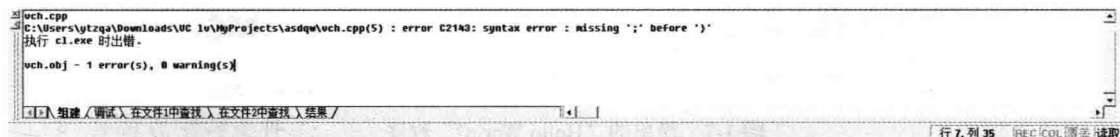


图 1-11 语法错误提示窗口

错误提示主要包括错误和警告及其个数、错误出现的行号及其原因。在输出窗口中双击错误提示信息或按 F4 键, 会出现一条醒目的蓝色条带突出提示信息, 同时通过一个箭头符号定位产生错误的语句, 如图 1-12 所示。

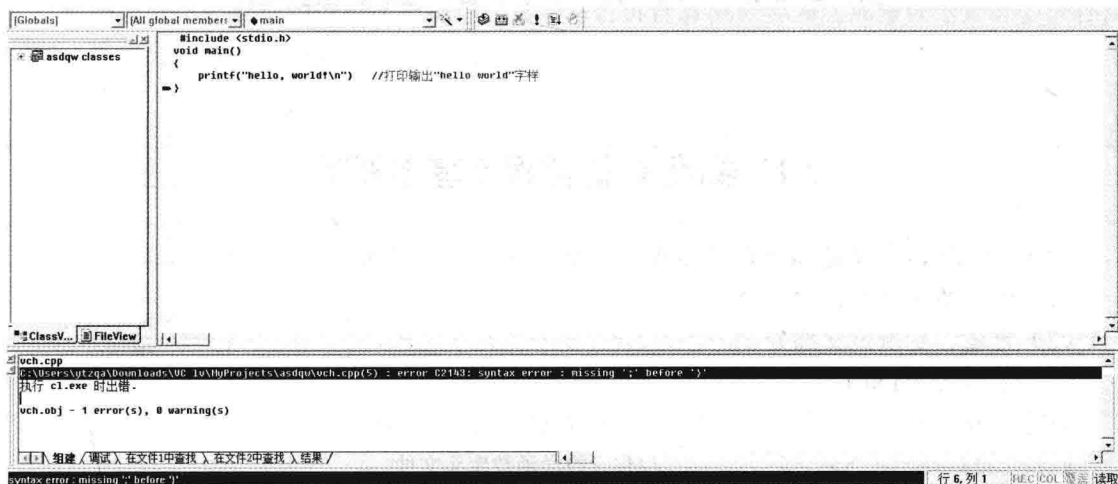


图 1-12 语法错误位置提示

提示: 程序中的一处错误往往会引出若干条错误提示信息, 因此修改一个错误后最好马上编译程序。通过反复的编译, 可使程序中的错误越来越少, 指导所有的语法错误都被修正。

第2章 标准C语言程序基础

2.1 标准C语言程序基本框架

VC6.0 中的 C 程序是标准 C 语言程序，基本包括三个部分：

(1) 库函数头文件包含部分。

(2) 常量、变量定义部分。

(3) 主函数 main()。

```
#include <stdio.h>           //包含标准输入输出头文件
#include <math.h>            //包含数学函数库头文件
#define PI 3.14              //定义常量PI，它的值是3.14
int a, b;                    //整型变量，存储矩形长度和宽度
float c, circum;             //浮点型变量，存储对角线长度和矩形外接圆周长
void main( )                 //主函数
{
    a=2;                     //长度赋值
    b=3;                     //宽度赋值
    c=sqrt((a*a+b*b));        //计算对角线长度
    circum=PI * c;           //计算外接圆周长
    printf("矩形对角线长度是%f\n", c);           //输出对角线长度
    printf("直径是%f，周长是%f\n", c, circum);    //输出外接圆直径和周长
}
```

2.2 库函数头文件调用

C 语言中有许多已经编写好的函数，可以直接调用，极大地减轻了程序员的工作量，我们编程时应该利用这部分资源来简化程序。

常用的库函数头文件有：

stdio.h：标准输入输出函数库。其中包含我们常用的 scanf() 输入函数，printf() 输出函数。

math.h：数学计算函数库。其中包含有三角函数、反三角函数、指数与对数、取绝对值等数学运算函数。

string.h：字符串处理函数库。其中包括字符串复制函数、拼接函数、查找函数、比较函数等，详情见附录 2。

2.3 基本数据类型

数据类型是对数据的一种抽象描述，目的是告诉计算机需要为这部分数据开辟多少存储空间。在定义变量时必须定义数据类型。常用数据类型见表 2-1。

表 2-1 常用数据类型

数据类型	类型标识符	占有字节	数据范围
整型	int	2	-32 768~32 767
长整型	long	4	-2 ³¹ ~2 ³¹ -1
浮点型	float	4	3.4×10 ⁻³⁸ ~3.4×10 ³⁸
双精度浮点型	double	8	1.7×10 ⁻³⁸ ~1.7×10 ³⁸
字符型	char	1	-128~127
字符串型	string	根据字符串长度	

注：以上数据类型的占有字节和数据范围会随计算机硬件的不同而有所变化。

1. 整数类型

整数是如-2，-1，0，1，2，…的无小数部分的数。C 语言提供了多种整数类型数据，以适应不同场合的需求，其中经常用到的是整型（int）和长整型（long）这两种数据类型。

整型（int）在计算机内存中占据 2 字节的存储空间，用 16 位二进制数表示，数值范围为-2¹⁵~2¹⁵-1（-32 768~32 767）。

长整型（long）在计算机内存中占据 4 字节的存储空间，用 32 位二进制数表示，数值范围为-2³¹~2³¹-1（-2 147 483 648~2 147 483 647）。

提示：当整型（int）或长整型（long）定义的数据超出允许数值范围时，会发生溢出错误。同时，溢出错误不是语法错误，不会被编译器指出，所以程序员在运算时应注意数据的数值范围，特别是在乘法运算和指数运算时。

2. 实数类型

实型也叫浮点型，指有小数部分的非整数数值，比如像 123.45 和 1.2×10⁹。它们在计算机内部也是以二进制的形式存储和表示的。C 语言提供了两种表示实数的类型：单精度型和双精度型。

单精度型（float）在计算机内存中占据 4 字节，有效位为 7 位。

双精度型（double）在计算机内存中占据 8 字节，有效位为 16 位。

3. 字符类型

字符型数据包括两种：单个字符和字符串。如：'A'是字符，"hello"是字符串。

单个字符用 char 定义，在计算机中字符是以 ASCII 码的形式存储的，一个字符占用 1 字节的存储空间。

字符串用 string 定义，在计算机中可以看作是以字符数组的形式存储的，其存储空间依据字符串长度而定。

提示：字符与 ASCII 码的对应关系可查阅附录 1。

2.4 常量与变量

2.4.1 常量

常量是不随时间和程序变化而变化的值，如上述程序中 PI，它的值是 3.14，程序中所有出现 PI 的地方，都相当于 3.14，在这种情况下 PI 不能被赋值，即不能出现“PI=...”这样的赋值语句。

常量的定义有两种方式：

(1) #define <常量名><学量值>，如上述程序中的“#define PI 3.14”，这种定义方法，PI 没有数据类型，也不占有存储单元，只是在编译时，所有的字符串 PI 都会被替换为 3.15。语句结尾不需要加分号“;”。

(2) const<常量数据类型><常量名>=<常量值>，如“const float PI=3.14;”，这种定义方法与变量定义类似，有数据类型，也占有存储单元，在语句开头加上“const”字样来表示常量定义。语句结尾必须加分号“;”。

定义常量时，必须给常量赋值，如：

```
#define PI 3.14
const float PI=3.14;
```

1. 整型常量

整型常量即整数。虽然 C 语言允许整数采用十进制、八进制和十六进制书写，但一般还是采用比较熟悉的十进制形式。注意，C 语言不支持二进制形式。

2. 实型常量

实型常量即实数。实数可以使用两种方式书写：一种是小数形式，例如 123.45，-2.0，0.5；另一种是指数形式，其中用字母 e 或者 E 表示 10 的幂次，例如 1.2345E2 和 1.2E-9 分别表示 123.45 和 1.2×10^{-9} 。

3. 字符常量

字符常量即单个字符，书写时要用单引号将这个字符括起来，例如 'A'、'2'、'#'等，它们属于常规字符。另外，还有一些字符比较特殊，不可视或无法通过键盘输入，例如换行符、回车符，C 语言的解决方法是用转义字符表示它们。

转义字符由一个反斜杠“\”后跟规定字符构成。常用转义字符的定义见表 2-2。

表 2-2 常用转义字符

转义字符	含 义	转义字符	含 义
\n	换行符	\\	反斜杠
\t	横向跳格符	\'	单引号
\0	空字符	\"	双引号

提示：转义字符从书写上看是一个字符序列，实际上是作为 1 个字符对待的，存储时只占 1 字节。

由于字符在计算机内是以 ASCII 码形式存储的, 实际上 ASCII 码值是一个 0~127 之间的整数, 因此字符常量也是可以参加运算, 例如:

'a'+1	字符'a'的 ASCII 码值加 1
'a'+32	字符'a'的 ASCII 码值减 32, 可用于大小写字母的转换
'a'<'b'	实际是比较两个字符的 ASCII 码值

4. 字符串常量

字符串常量简称字符串, 是用双引号括起来的一串字符。例如: "china"、"x"等。这里的双引号只起定界的作用, 它不属于字符串中的字符, 因此双引号之间的字符个数才是字符串的长度。但是, 字符串在内存中占用的存储字节数要比字符串长度多 1, 因为 C 语言总是自动地在字符串尾部加上一个转义字符 '\0' (空字符)。

2.4.2 变量

变量是程序设计语言的一个重要概念, 它是指程序运行期间值可以发生变化的量。

可以认为, 变量是一个存储数据的容器, 即存储单元, 它的功能就是用来存放程序中需要处理的数据, 这些数据可以是原始数据、中间结果或最终结果。

变量具有保持值的性质, 但是当给变量赋值时, 新值会取代旧值, 这就是变量的值发生变化的原因。

变量定义时, 可以在定义数据类型时同时赋值, 也可以只定义数据类型在后续的程序中赋值, 如果不赋值, 默认为变量值为 0。如下述两种方式都是正确的:

```
(1) int i;
```

```
    i=3;
```

```
(2) int i=3;
```

出于对程序稳定性的考虑, 不推荐使用默认值 0 来进行运算操作。如:

错误写法:

```
int a, b;
```

```
b=a+1;
```

正确写法:

```
int a=0;
```

```
int b=0;
```

```
b=a+1;
```

关键字: 指 C 语言中已有明确含义的词汇, 如 main, int, void 等。在常量名和变量名的选取时, 不能使用这些词汇。

2.5 主函数 main()

一个程序, 无论复杂或简单, 总体上都是一个“函数”, 这个函数就是主函数 main(), 它是程序运行的入口, 一个 C 程序中有且只能有一个 main 函数。程序的数据处理全部都在 main 函数中进行。

2.6 顺序结构及其应用

顺序结构程序的执行逻辑是按照先后顺序，是三种基本结构中最简单的一种。

组成顺序结构程序的语句，可以是表达式也可以是函数，这里将标准 C 语言中最常用的输入输出函数做个介绍。

2.6.1 格式输出函数

printf()是标准 C 语言中的格式输出函数，其功能是把计算机中的数据输出到显示屏上，并且可以指定数据的格式。在使用前必须调用库函数“stdio.h”，示例程序如下：

```
#include "stdio.h"
{
    printf("Hello!\n");
}
```

printf() 函数调用的格式为：

printf(格式控制字符串，输出项表)；

例如：printf("a=%d, b=%d\n", a, b)；

函数说明：

(1) 格式控制字符串可以包含三类字符。

格式字符：由%打头后跟格式符。其中格式符有 C 语言约定，作用是指定数据输出时的格式。在打印结果中“%格式符”会被输出项表的值替换。格式符及其功能说明见表 2-3。

表 2-3 常用 格 式 符

格式符	printf()	scanf()
d	输出一个十进制整数 (int 型)	输入一个十进制整数 (int 型)
f	输出一个单精度实数 (float 型)	输入一个单精度实数 (float 型)
c	输出一个字符 (char 型)	输入一个字符 (char 型)
s	输出字符串	输入字符串
ld	输出一个十进制整数 (long 型)	输入一个十进制整数 (long 型)
lf	输出一个双精度实数 (double 型)	输入一个双精度实数 (double 型)

转义字符：'\n' 是输出函数中做常用的转义字符，起回车换行的作用。转义字符含义详见“字符常量”一节。

普通字符：格式控制字符串中除了格式字符和转义字符以外，其余都是普通字符，普通字符将原样输出。

(2) 输出项表。输出项表列出要输出的数据项，数据项可以是常量、变量或表达式，各输出项之间用逗号分隔。

通过以下 5 个程序段，观察 printf() 的输出效果。

```
1) printf ("Good morning!");
```

输出结果为：