

# C 语言基础

通过本模块的学习,读者将对 C 语言和程序设计有一个初步的认识,熟悉 Visual C++ 6.0(VC++ 6.0)或 Devcpp 的集成开发环境。通过自己动手编写第一个 C 语言程序,读者可以了解 C 语言的基本语法和编写 C 语言程序的基本思路。



## 工作任务

- 输出“Hello World! Hello C!”——了解 C 语言的结构。
- 熟悉编写 C 语言程序的环境。
- 猜牌游戏的体验——C 语言的综合运用一。
- 打字游戏的体验——C 语言的综合运用二。



## 技能目标

- 了解 C 语言的结构和语法规则。
- 学会编写一个简单的 C 语言程序。
- 了解 C 语言简单的输入/输出语句。
- 了解 C 语言的实验环境及上机步骤。

## 任务 1.1 输出“Hello World! Hello C!” ——了解 C 语言的结构



### 任务描述

在屏幕上显示一行“Hello World! Hello C!”的文字。



### 任务分析

为了创建出第一个 C 语言程序,首先应了解 C 语言的程序结构和输出函数 printf() 等语句的用法,并熟悉相应开发工具的使用。

#### 1.1.1 计算机程序及其设计语言

“程序”这两个字从字面上理解是指一件事情进行的先后次序。计算机程序则是让计算机有步骤地完成某项工作。要让计算机完成某项工作,人们需要把事情描述出来让计

算机理解,这种人机之间交换信息的工具称为计算机程序设计语言。

自从世界上第一台计算机于1946年问世以来,用于编写计算机程序的程序设计语言由机器语言发展到汇编语言,又由汇编语言发展到高级语言。

机器语言是指计算机本身自带的指令系统。计算机的指令是由二进制数的序列组成的,用来控制计算机进行某种操作。用机器语言编写的程序不必通过任何翻译处理,计算机硬件就能够直接识别和接受。因此,用机器语言编写的程序,具有质量高、执行速度快和占用存储空间少等优点,但它缺乏直观性,并且难学、难记、难检查以及难修改。

为了解决机器语言的缺点,出现了汇编语言。汇编语言是一种面向机器的程序设计语言,它用助记符和符号地址代替机器指令,这使它变得易记,读起来更容易,检查及修改更方便。但是用汇编语言编写的程序并不能被计算机直接识别和接受,必须由一个起翻译作用的程序将其翻译成机器语言程序计算机才能执行,这个起翻译作用的程序通常称为汇编程序,这个翻译过程称为汇编。

汇编语言的缺点是依赖于具体的机器,不具有通用性和可移植性,且与人们习惯使用的自然语言和数学语言相差甚远,因此又出现了所谓的高级语言。

高级语言是一种更接近于人们习惯使用的自然语言和数学语言的程序设计语言,人们用它来编写计算机程序。比起使用机器语言和汇编语言,高级语言显然要方便很多。

C语言就是一种高级语言,它用比较接近人的思维和表达问题的方式来描述问题并规范计算机程序,然后以编译的方式进行翻译。

### 1.1.2 第一个C语言程序

**【例 1.1】** 在屏幕上显示两行文字“Hello World!”和“Hello C!”。

```
/* 打印两行语句 */
#include <stdio.h>
main()
{
    printf("Hello World!\n");          //打印第一条语句
    printf("Hello C!\n");              //打印第二条语句
}
```

### 1.1.3 第一个C语言程序的说明

程序的功能是在屏幕上显示两行文字信息。

第1行“/\* … \*/”为多行注释符号,其间的内容为注释,用来帮助读者了解程序的功能。从“/\*”开始,到“\*/”结束,注释不会被编译和运行。第5、6行的“//”称为行注释号,“//”的注释内容到行末自动结束。

第2行中的#include是C预处理程序的一条包含命令,stdio.h包含标准的输入/输出函数信息,放在源程序的前面。

第3行中的main()为主函数名。每一个程序必须有一个主函数,且只能有一个主函数。C语言程序总是从main()函数开始执行。

第4行与第7行的“{ }”为主函数的界定符。“{ }”必须成对出现;每个函数必须用

“{}”括起来。

第5、6行为打印输出语句。printf()函数的功能是把函数括号中“”内的内容输出到显示器显示。“\n”转义字符在此起回车换行的作用,每一条语句的结束必须加“;”符号。

 **注意:** 如果用 Devcpp 新版编程环境,主函数 main()之前不用加任何类型,默认返回值为整型;如果用 Visual C++ 6.0 编程环境,无返回值时需在 main()之前加 void。void 表示空,即无返回值的意思。

#### 1.1.4 C 语言程序结构

C 语言是一门函数化的语言,每个函数都完成一定的功能。用户既可以编写自己的函数(称为自定义函数),也可以根据需要调用库函数和自定义函数完成相应的任务。C 语言程序必须按规定的格式书写。

C 语言程序结构的语法格式如下,“[]”中的内容为可选部分。

```
[预处理命令]
[子函数类型说明]
[全程变量定义]
[返回值的类型] main()
{
    局部变量定义
    <程序体>
}
[返回值的类型 sub1()]
{
    局部变量定义
    <程序体>
}
...
[返回值的类型 subN()]
{
    局部变量定义
    <程序体>
}
```

总结起来,C 语言程序的特点如下。

(1) 一个 C 语言源程序可以由一个或多个源文件组成。源程序中可以有预处理命令(最常用的 include 命令仅为其中的一种,它包含需要调用库函数的头文件),预处理命令通常放在源文件或源程序的最前面,以“#”开头。

(2) 一个源程序无论由多少个文件组成,都有且只有一个 main()主函数。

(3) 每个源文件可由一个或多个函数组成。sub1()、…、subN()代表用户定义的子函数。

(4) 程序体指 C 语言提供的任何库函数调用语句、控制流程语句或其他子函数调用语句等。

(5) 每一行变量的声明、每一条语句都必须以分号结尾。但预处理命令、函数头和花括号“}”之后不能加分号。

(6) 标识符、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,也可不再加空格来间隔。

### 【课堂思考】

(1) 模仿例 1.1,在屏幕上显示两行中文文字信息,分别是“你好!”和“让我们一起学习 C 语言程序!”。

(2) 举一反三,编写程序在屏幕上输出显示信息。

① 编写一个小程序,在屏幕上输出以下内容。

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$           This is a C program           $
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

```

② 编写一个小程序,在屏幕上输出以下内容。

```

*****
*           学生成绩管理系统           *
*****

```

③ 编写一个程序,实现在屏幕上输出课程表的内容。

| 课 程 表      |             |             |
|------------|-------------|-------------|
| 8:00—10:00 | 10:00—12:00 | 14:00—16:00 |
| 一 高等数学     | 大学英语        | 程序设计        |
| 二 思想政治     | 高等数学        | 程序设计上机      |
| 三 大学语文     | 程序设计        | 大学英语        |
| 四 高等数学     | 网页设计        | 网页设计上机      |
| 五 计算机基础    | 体育          | 计算机基础上机     |

## 任务 1.2 熟悉编写 C 语言程序的环境



### 任务描述

熟悉 Visual C++ 6.0 或 Devcpp 的集成开发环境。



### 任务分析

在 Visual C++ 6.0 或 Devcpp 实际开发环境中完成指定操作。



VC++ 编程环境介绍



Devcpp 编程环境介绍

### 1.2.1 Visual C++ 6.0 编程环境介绍

Visual C++ 6.0(简称 VC++ 6.0)集成开发环境是一个将程序编辑器、编译器、调试工具和其他建立应用程序的工具集成在一起并用于开发应用程序的软件系统。这里简单介绍 C 语言程序在该开发环境中如何编辑、编译、连接和运行。具体操作步骤如下。

(1) 启动 VC++ 6.0, 选择“文件”菜单中的“新建”命令, 如图 1.1 所示。

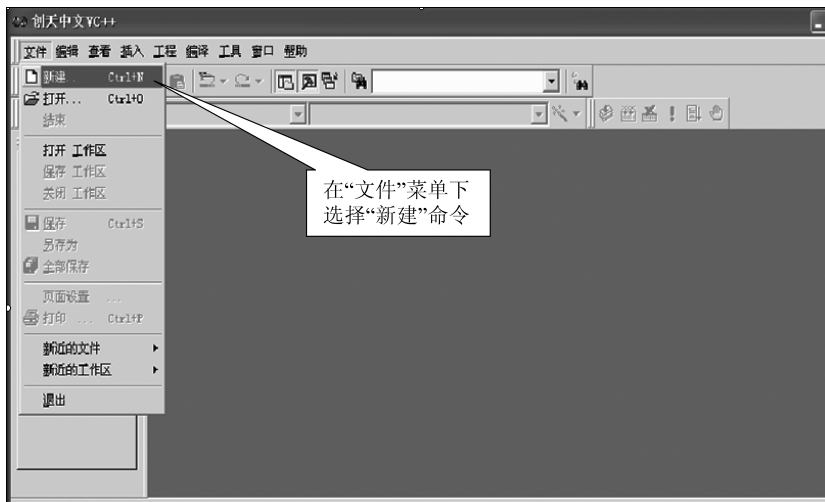


图 1.1 打开新建文件

(2) 在弹出的“新建”对话框中选择“文件”选项卡, 再选择 C++ Source File 选项; 选择一个想存放文件的目录, 如图 1.2 所示选择了 E:\TEMP 目录, 若没有该目录, 则新建一个; 新建的源文件命名为 1\_11。

❗注意: 尽管 C 语言源文件扩展名为“.c”, 但因为本书的教学实验环境用的文件的默认扩展名为“.cpp”, 所以请尽量不要输入扩展名, 以免出错。

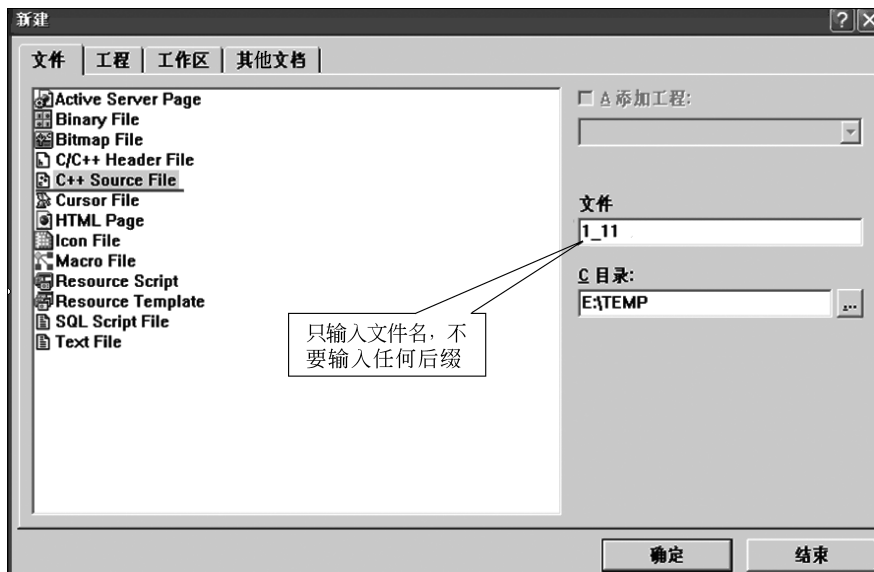


图 1.2 源文件的命名

(3) 为命名为 1\_11 的源文件输入程序的框架代码, 如图 1.3 所示。注意不要在中文状态下输入代码, 也不要使用全角标点。

(4) 编译源文件代码,检查是否正确,如图 1.4 所示。编译命令的快捷键是 Ctrl+F7。关闭两个对话框,并依次单击“是”按钮确认,如图 1.5 所示。

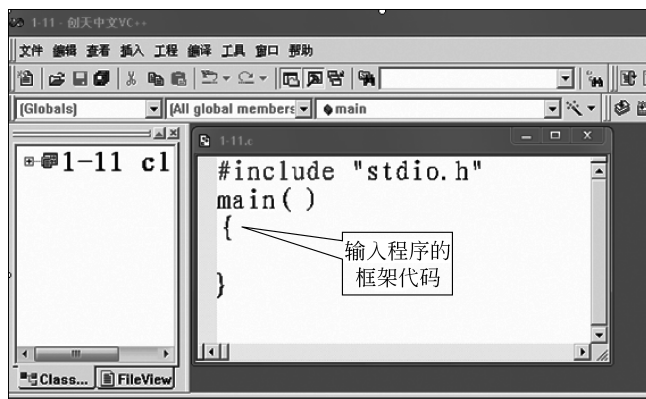


图 1.3 程序框架

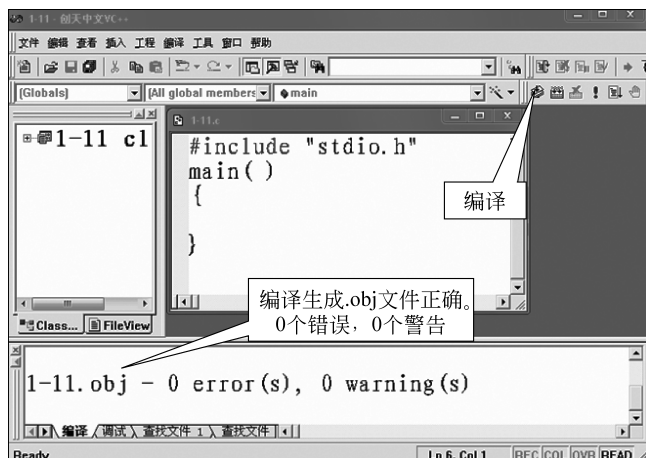


图 1.4 程序编译



图 1.5 编译确认

编译源文件后生成.obj文件,如果程序输入没有错误,就会在下面显示“0 error(s), 0 warning(s)”,表明程序有“0个错误,0个警告”,表示输入的程序通过了编译检查,无错误。如果两项不为0,则要修改错误的代码后再进行编译,直至编译显示正确为止。

(5) 在源文件的程序框架中补充代码。如果对操作步骤比较熟悉,则可不要第(4)步,直接由第(3)步跳到第(5)步输入代码,如图1.6所示。

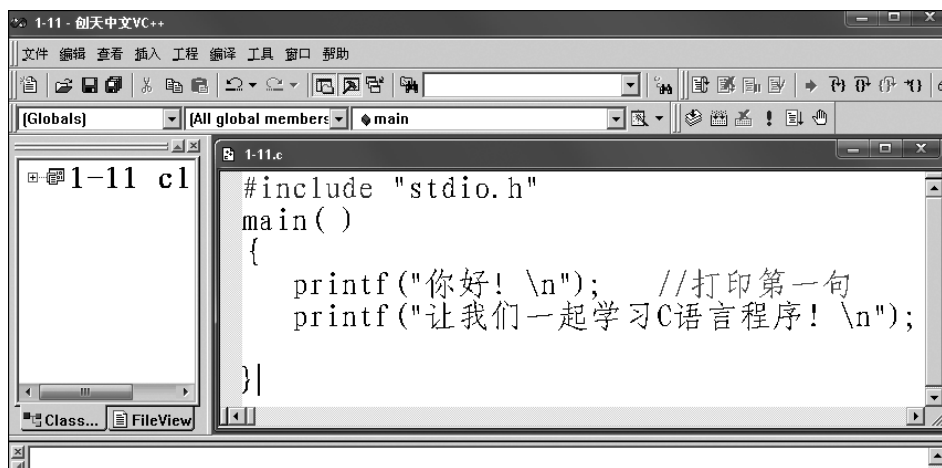


图 1.6 程序代码的输入

(6) 编译、连接程序代码,生成 EXE 文件,编译正确时的显示如图 1.7 所示。



图 1.7 连接生成 EXE 文件

(7) 运行程序。编译、连接程序的操作通过后,运行生成的 EXE 文件,如图 1.8 所示,程序运行结果如图 1.9 所示。

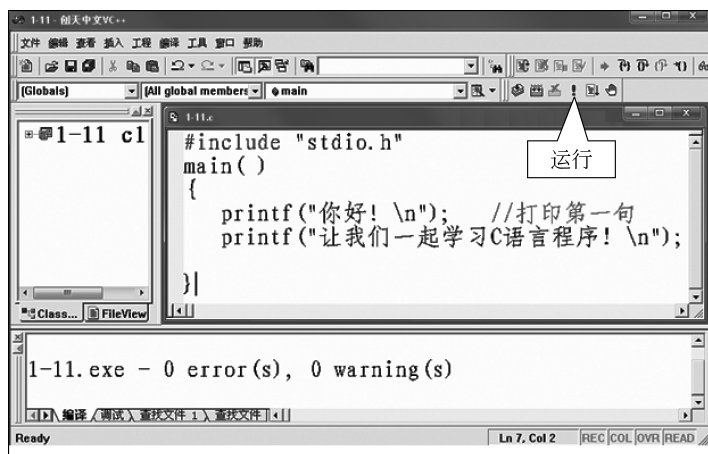


图 1.8 运行程序

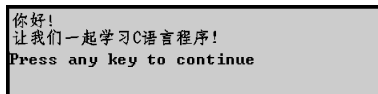


图 1.9 程序运行结果

## 1.2.2 Devcpp 编程环境介绍

Devcpp(标准名是 Dev-C++) 是一个 C 和 C++ 的开发工具,它是一款自由软件,遵循 C/C++ 标准。它的开发环境包括代码编辑窗口、工程编译器及调试器等,在工程编译器中集合了编译器、连接程序和执行程序。这款软件满足了初学者与编程高手的不同需求,是学习 C/C++ 的理想开发工具。

这里简单介绍一下 C 语言程序在该开发环境中如何编辑、编译、连接和运行。具体操作步骤如下。

(1) 启动 Devcpp,选择“文件”→“新建”→“源代码”命令(快捷键为 Ctrl+N),如图 1.10 所示。

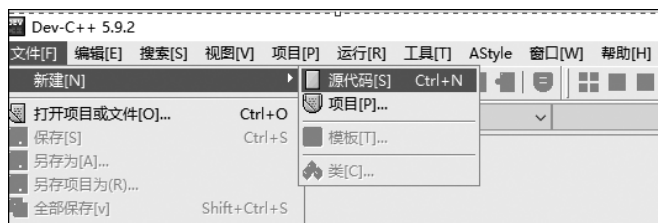


图 1.10 新建文件

新建文件后,出现的是未命名的文件,如图 1.11 所示。

(2) 给新建的文件取一个文件名。选择“文件”→“另存为”命令,如图 1.12 所示。把新建的文件存入指定的文件夹,并给新文件取一个具体的名字。

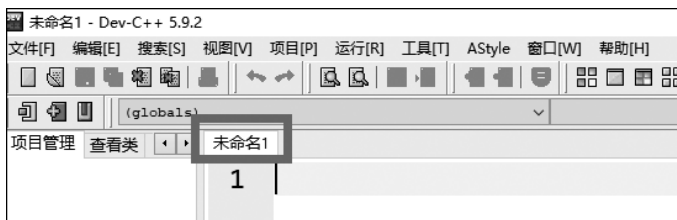


图 1.11 未命名的文件

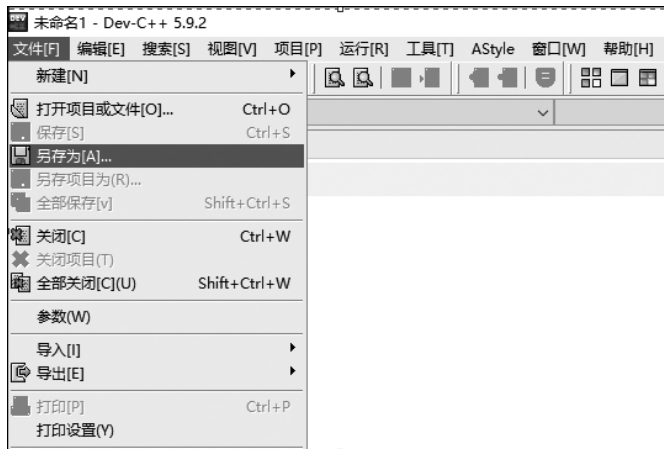


图 1.12 保存新文件

此处给第一个文件取名为 1\_1。由于 Devcpp 文件的默认扩展名为“.cpp”，所以在给文件取名时，尽量不要输入或修改扩展名，保持默认的扩展名“.cpp”，以免出错，如图 1.13 所示。

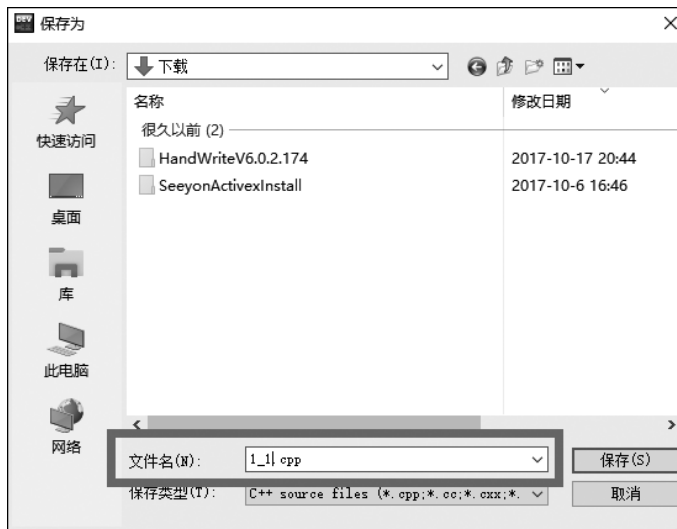


图 1.13 在指定目录下给新文件取一个名字

(3) 1-1.cpp 源文件的代码如图 1.14 所示,这个程序的功能是在屏幕上显示一行文字“Hello World! Hello C!”。注意不要在中文状态下,也不要使用全角标点符号。

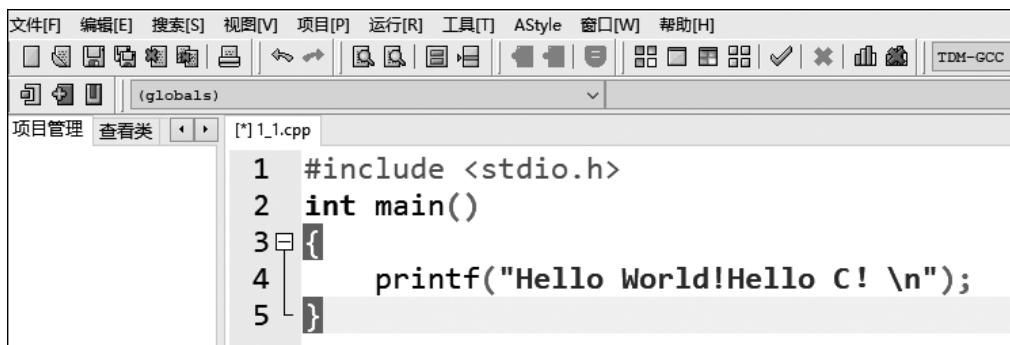


图 1.14 第一个 C 语言程序

(4) 编译源文件代码,检查程序语法是否正确。选择“运行”→“编译”命令(快捷键是 F9)可编译源程序,如图 1.15 所示。



图 1.15 编译源程序

如果没有语法错误,源程序顺利通过编译后,显示如图 1.16 所示。

(5) 运行程序。生成可执行文件(.exe)后可直接运行程序。选择“运行”菜单下的“运行”命令(快捷键是 F10)可运行程序,如图 1.17 所示。

程序运行结果如图 1.18 所示。

**注意：**本书所用的 Devcpp 5.9.2 这个版本可以把第(4)步的编译操作和第(5)步的运行程序操作整合在一起,选择“编译运行”命令即可(该命令的快捷键是 F11),如图 1.19 所示,同样能得到所要的结果。