



计算机“十三五”规划教材  
互联网+计算机教育立体化教材

# C语言程序设计

主编 张其文

含微课

## 由浅入深，循序渐进

基础知识 > 核心技术 > 高级应用

## 案例典型，针对性强

采用“一个知识点，一个例子”的模式，难易适中，针对性强

## 配套微课，扫码即得

采用二维码技术，轻松扫码即可获取相关微课视频

## 资源丰富，易教易学

提供丰富的教学资源，包括微课视频、例题源代码、习题答案等

航空工业出版社

内容简介

计算机“十三五”规划教材

互联网+计算机教育立体化教材

# C 语言程序设计

主编 张其文

目录页插图

航空工业出版社

北京

## 内 容 提 要

本书从初学者的角度出发,以通俗易懂的语言、丰富多彩的实例,详细地介绍了使用C语言进行程序开发所需掌握的知识和技术。本书共分为12章,内容包括绪论,数据类型、运算符及表达式,选择结构程序设计,循环结构程序设计,函数,数组,指针,结构体及其他构造数据类型,文件,预处理,程序设计,C++简介。

本书可作为各类院校和培训机构C语言程序设计课程的专用教材,也可供程序设计爱好者自学使用。

## 图书在版编目(CIP)数据

C语言程序设计 / 张其文主编. — 北京: 航空工业出版社, 2018.9

ISBN 978-7-5165-1721-5

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(2018)第207716号

## C语言程序设计 C Yuyan Chengxu Sheji

航空工业出版社出版发行

(北京市朝阳区北苑2号院 100012)

发行部电话: 010-84936597 010-84936343

三河市祥达印刷包装有限公司印刷

全国各地新华书店经售

2018年9月第1版

2018年9月第1次印刷

开本: 787×1092

1/16

印张: 19.75

字数: 456千字

印数: 1—2500

定价: 49.80元



# 前言

C 语言作为一种计算机编程语言，具有功能强大、使用灵活、可移植性好等特点，深受广大编程人员的喜爱。另外，由于 C 语言的概念简洁、语句紧凑、表达能力强、程序结构性和可读性好，因此很多院校开设了 C 语言程序设计课程作为第一门计算机语言课程。同时，全国计算机等级考试（NCRE）、全国计算机应用水平考试（NIT），以及全国各地组织的大学生计算机统一考试都将 C 语言列入了考试范围。

随着社会的发展，传统的教学模式已难以满足就业的需要。本书根据应用型人才培养的需要，结合作者多年讲授 C 语言程序设计课程的教学经验编写而成。

## 一、本书特色

总体而言，本书主要具有以下几个方面的特色：

- **由浅入深，循序渐进。**本书首先介绍了 C 语言的基础知识，然后介绍了 C 语言的基本程序结构，接着介绍了 C 语言的核心技术，最后介绍了 C 语言的高级应用，由浅入深，符合读者的认知规律和学习流程，帮助读者循序渐进地学习相关知识。
- **案例典型，针对性强。**本书采用“一个知识点，一个例子”的模式，安排了大量例题，难易适中，针对性强。
- **配套微课，扫码即得。**本书采用最新的二维码技术，读者借助手机或其他移动设备扫码即可获取相关微课视频，从而更方便地理解和掌握相关内容。
- **资源丰富，易教易学。**本书配有丰富的教学资源，包括微课视频、例题源代码、习题答案等，方便老师教学和学生学习。

## 二、本书作者团队

本书由张其文任主编，侯亮、张艳任副主编，施秋霞参与编写。其中，第 1~4 章由侯亮编写，第 5~8 章由张其文编写，第 9 章由张艳编写，第 10~12 章由施秋霞编写。全书由张其文、张艳负责统稿，任旭鹏、胡颖杰负责校对和程序验证，马威阅读了本书的初稿并提出了许多具有建设性的意见，在此表示感谢。另外，本书在编写过程中也参阅了大量的国内外相关文献，在此一并向同行和相关作者表示诚挚的谢意。



### 三、教学资源下载

本书配有丰富的教学资源，读者可从网站（[www.bjjqe.com](http://www.bjjqe.com)）下载。如果读者在学习过程中有什么疑问，也可登录该网站寻求帮助，我们将会及时解答。

尽管我们在编写本书时已竭尽全力，但书中存在的疏漏及错误之处在所难免，敬请广大读者批评指正。

编者

2018年8月

编者

编者



第1章 绪论	1	2.4.3 关系运算符及关系表达式	38
1.1 C语言的发展历史	1	2.4.4 逻辑运算符及逻辑表达式	39
1.2 C语言的特点	2	2.4.5 逗号运算符及逗号表达式	41
1.3 C语言的结构	2	2.4.6 位运算符和位运算	42
1.4 Visual Studio 2010 简介	6	2.4.7 sizeof 运算符及其表达式	45
1.4.1 Visual Studio 2010 集成开发环境	7	2.5 不同数据类型之间的转换	45
1.4.2 用 Visual Studio 2010 运行 C 语言程序的一般步骤	8	2.6 数据的输入与输出	48
1.5 算法	14	2.6.1 数据的输出	48
1.5.1 算法的特征	14	2.6.2 数据的输入	53
1.5.2 算法设计的原则	15	2.7 C语言的基本语句简介	56
1.5.3 算法的表示形式	15	本章小结	57
1.5.4 算法的基本结构	16	习 题	58
1.5.5 算法复杂性	18	第3章 选择结构程序设计	62
本章小结	18	3.1 if 语句	62
习 题	19	3.1.1 基本的 if 语句	62
第2章 数据类型、运算符及表达式	21	3.1.2 if-else 语句	63
2.1 基本概念	22	3.1.3 if-else-if 语句	64
2.1.1 标识符	22	3.2 if 语句的嵌套	67
2.1.2 关键字	22	3.3 条件运算符与条件表达式	68
2.1.3 常量与变量	23	3.4 switch 语句	69
2.2 基本数据类型	24	本章小结	73
2.2.1 整型数据	24	习 题	73
2.2.2 实型数据	27	第4章 循环结构程序设计	77
2.2.3 字符型数据	30	4.1 while 循环	77
2.3 变量的初始化	33	4.2 do-while 循环	80
2.4 运算符及表达式	33	4.3 for 循环	81
2.4.1 算术运算符及算术表达式	34	4.4 循环嵌套	85
2.4.2 赋值运算符及赋值表达式	37	4.5 goto 语句	86



4.6 其他循环控制语句	87	6.3.4 字符串处理函数	135
4.6.1 break 语句	87	6.3.5 字符数组及字符串操作	
4.6.2 continue 语句	90	应用举例	138
本章小结	91	6.4 数组作为函数的参数	141
习 题	91	6.4.1 数组元素作为函数参数	141
<b>第 5 章 函 数</b>	<b>95</b>	6.4.2 数组名作为函数参数	142
5.1 函数的定义和调用	95	本章小结	146
5.1.1 函数的定义	95	习 题	147
5.1.2 函数的调用	96	<b>第 7 章 指 针</b>	<b>150</b>
5.2 函数的参数传递和返回值	99	7.1 指针和地址的概念	150
5.2.1 函数的形参与实参	99	7.2 指针变量的定义和使用	151
5.2.2 函数的返回值	101	7.2.1 指针变量的定义	151
5.3 函数的嵌套调用	102	7.2.2 指针变量的初始化	151
5.4 函数的递归调用	104	7.2.3 指针变量的引用	152
5.5 变量的作用域及存储类别	107	7.3 空指针与 void 指针	155
5.5.1 变量的作用域	107	7.3.1 空指针	155
5.5.2 变量的存储类别	111	7.3.2 void 指针	156
5.6 内部函数及外部函数	115	7.4 指针与数组	156
5.6.1 内部函数	116	7.4.1 指针与一维数组	156
5.6.2 外部函数	116	7.4.2 指针与二维数组	161
本章小结	117	7.4.3 指针与字符串	167
习 题	117	7.4.4 指针数组	168
<b>第 6 章 数 组</b>	<b>120</b>	7.4.5 指向指针的指针	170
6.1 一维数组	120	7.5 指针与函数	172
6.1.1 一维数组的定义	120	7.5.1 指针变量作为函数参数	172
6.1.2 一维数组的引用	121	7.5.2 指针作为函数的返回值	179
6.1.3 一维数组的初始化	123	7.5.3 指向函数的指针	180
6.1.4 一维数组应用举例	124	本章小结	182
6.2 二维数组	127	习 题	182
6.2.1 二维数组及多维数组的定义	127	<b>第 8 章 结构体及其他构造数据类型</b>	<b>186</b>
6.2.2 二维数组的引用	128	8.1 结构体	186
6.2.3 二维数组的初始化	128	8.1.1 结构体类型声明	187
6.2.4 二维数组应用举例	129	8.1.2 结构体变量的定义	188
6.3 字符数组和字符串	131	8.1.3 结构体变量的引用	191
6.3.1 字符数组的定义和引用	131	8.1.4 结构体变量的初始化	192
6.3.2 字符数组的初始化	132	8.2 结构体数组	193
6.3.3 字符数组的输入输出	133	8.2.1 结构体数组的定义及初始化	193



8.2.2 结构体数组的引用	195	第 10 章 预处理	239
8.3 结构体指针	197	10.1 概述	239
8.3.1 指向结构体变量的指针	197	10.2 宏定义	239
8.3.2 指向结构体数组的指针	199	10.2.1 无参数宏定义	239
8.3.3 结构体作为函数参数	200	10.2.2 有参数宏定义	242
8.4 链表	202	10.3 文件包含	246
8.4.1 链表概述	202	10.4 条件编译	247
8.4.2 处理链表所需的函数	203	10.4.1 条件编译指令	248
8.4.3 建立动态链表	204	10.4.2 条件编译形式	248
8.5 共用体	207	本章小结	250
8.5.1 共用体类型声明	207	习 题	251
8.5.2 共用体变量的定义	208	第 11 章 程序设计	253
8.5.3 共用体变量的赋值和使用	209	11.1 程序设计方法	253
8.6 枚举类型	211	11.1.1 程序设计语言与 程序设计方法	253
8.7 位域结构	213	11.1.2 程序设计方法的演变	254
8.8 用户自定义类型	214	11.1.3 程序设计的过程	255
本章小结	216	11.2 综合实例——学生信息 管理系统	256
习 题	216	11.2.1 总体设计	256
第 9 章 文 件	220	11.2.2 详细设计	257
9.1 文件概述	220	本章小结	273
9.2 文件指针	221	习 题	273
9.3 文件打开与关闭	222	第 12 章 C++简介	275
9.3.1 文件打开——fopen()函数	222	12.1 面向对象方法	275
9.3.2 文件关闭——fclose()函数	224	12.1.1 面向对象的概念	275
9.4 文件读写	224	12.1.2 面向对象程序设计	277
9.4.1 字符读写函数 fgetc()和 fputc()	224	12.2 C++概述	278
9.4.2 字符串读写函数 fgets()和 fputs()	227	12.2.1 C++程序的开发过程	278
9.4.3 数据块读写函数 fread()和 fwrite()	229	12.2.2 C++程序的组成	279
9.4.4 格式化读写函数 fscanf()和 fprintf()	231	12.2.3 C++程序的结构	280
9.5 文件定位	232	12.3 C++对 C 的扩充	282
9.6 文件检测	234	12.4 C++的面向对象特征	289
本章小结	235	12.4.1 类和对象	289
习 题	236	12.4.2 构造函数和析构函数	291
		12.4.3 友元	293
		12.4.4 静态成员	294



12.4.5 运算符重载 ..... 296

12.4.6 继承和派生 ..... 299

12.4.7 多态性和虚函数 ..... 300

本章小结 ..... 303

习 题 ..... 303

附 录 ..... 305

附录 1 常用字符与 ASCII 代码  
对照表 ..... 305

附录 2 运算符的优先级和  
结合性表 ..... 306

参考文献 ..... 307

# 第 1 章 绪 论

## 1.1 C 语言的发展历史

20 世纪 60 年代,随着计算机科学技术的发展,高级程序设计语言得到了广泛的应用。但是当时的系统软件只能用汇编语言来编写,由于汇编语言依赖于计算机硬件,存在着不可移植、可读性差、描述问题效率低等缺点,给编程带来了很大不便。因此,一个很自然的想法就是能否找到一种语言工具,既具有汇编语言的某些功能,又具有高级语言的功能? C 语言的出现很好地解决了上述问题。

C 语言是在 B 语言(BCPL, 全称 Basic Combined Programming Language)的基础上发展起来的。而 B 语言是 1970 年由美国贝尔实验室设计的,并用于编写了第一个 UNIX 操作系统。其优点是精练,接近硬件;缺点是过于简单,数据无类型,描述能力有限。

1973 年,贝尔实验室的 D.M.Ritchie 在 B 语言的基础上设计出了 C 语言,并与 K.Thompson 用 C 语言改写了原来用汇编语言编写的 UNIX(即 UNIX 第 5 版)操作系统,但仅在贝尔实验室使用。

1975 年,UNIX 第 6 版发布,C 语言的突出优点引起了人们的关注。

1977 年,出现了不依赖于具体机器的 C 语言编译文本——可移植 C 语言编译程序,推动了 UNIX 在各种机器上的实现,C 语言也得到了推广。

1978 年,Brian W.Kernighan 和 Dennis M.Ritchie 合著了影响深远的名著《The C Programming Language》,该书介绍的 C 语言成为了后来广泛使用的 C 语言版本的基础,被称为标准 C。

1983 年,美国国家标准化协会(ANSI)根据 C 语言各种版本对 C 进行了扩充,制定了新的标准 ANSI C。

1989 年,ANSI 发布了第一个完整的 C 语言标准“ANSI X3.159-1989”,简称“C89”。

C89 在 1990 年被国际标准化组织(ISO, International Organization for Standardization)一字不改地采纳,并将其命名为“ISO/IEC 9899:1990”,简称“C90”。

1999 年,在做了一些必要的修正和完善后,ISO 发布了新的 C 语言标准,命名为“ISO/IEC 9899:1999”,简称“C99”。

2011 年 12 月,ISO 又正式发布了新的标准,称为“ISO/IEC9899:2011”,简称“C11”。



## 1.2 C 语言的特点

一种语言之所以能存在和发展,并具有生命力,总是有些不同于(或优于)其他语言的特点。C 语言之所以能成为目前用途最广泛的语言之一,在于其具有以下几个基本特点:

(1) C 语言简洁、紧凑,使用方便,语法规定少,数据类型丰富。C 语言一共只有 37 个关键字、45 个标准运算符,以及 9 种控制语句,程序书写形式自由,主要用小写字母表示,压缩了一切不必要的成分,相对其他语言源程序短,因此输入程序时工作量少。

(2) C 语言是处于汇编语言和高级语言之间的中级语言。它既具有高级语言的通用性和易写、易读的特点,又具有汇编语言的位处理、地址操作等能力,能实现汇编语言大部分功能,可以直接对硬件进行操作。

(3) C 语言是一种结构化程序设计语言,即程序的逻辑结构可以用顺序、分支和循环 3 种基本结构组成。C 语言具有结构化控制语句(如 if-else、switch、while、do-while、for 等语句),十分便于采用由顶向下、逐步求精的结构化程序设计方法。C 语言程序的函数结构,十分有利于把整体程序分割成若干相对独立的功能模块,并且为程序模块间的相互调用以及数据传递提供便利。因此,用 C 语言编写的程序,具有容易理解、便于维护的优点。

(4) C 语言程序可移植性好。C 语言程序本身并不依存于计算机硬件系统,从而便于在硬件结构不同、操作系统不同的计算机之间实现程序的移植。

(5) 目标程序质量高。C 语言的大多数运算符与一般机器指令一致,可直接翻译成机器指令。因此,用它编写的程序代码质量高。实践表明,C 语言的代码效率只比汇编语言低 10%~20%,而且 C 语言在描述问题时编程迅速、可读性好、表达能力强等优点是汇编语言无法比拟的。

由于 C 语言具有上述众多特点,因此得到了应用和普及。很多大型系统软件都是用 C 语言编写的,如 Unix 操作系统、早期的 Oracle 系统等。另外,许多以前只能用汇编语言处理的问题现在也可以用 C 语言来处理了。C 语言也因此被称为“高级汇编语言”。

当然,C 语言本身也有缺点。例如,运算符的优先级较多,不容易记忆,有些特殊的运算符容易出错;C 语言的语法限制不严格,在增强了程序设计灵活性的同时,也降低了安全性。所以用户要认真掌握、正确使用 C 语言,才能充分发挥其优点,编写出高效率的程序。

## 1.3 C 语言的结构

下面通过几个例子说明 C 语言程序的结构、特点及其设计风格。

**【例 1.1】** 输出显示“This is my first program.”。

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

```
/*编译预处理指令*/
```



```
int main()                /*主函数的函数头*/
{
    printf("This is my first program. \n"); /*输出要显示的字符串*/
    return 0;              /*程序返回值 0*/
}
```

运行这个程序时，在屏幕上会显示以下一行英文：

This is my first program.

上述程序是一个由头文件和主函数组成的简单 C 语言程序，下面分别解释一下各行代码的意义。

第 1 行：

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

这是一个预处理操作。“include”称为文件包含命令，后面尖括号中的内容称为头文件。“stdio.h”是 C 语言的系统文件，“stdio”是“standard input & output（标准输入输出）”的缩写，“.h”是文件的扩展名。由于程序的第 4 行使用了库函数 printf，编译系统要求程序提供有关此函数的信息（如对这些输入输出函数的声明、宏的定义、全局变量的定义等），所以此处需要这条命令。

第 2 行：

```
int main()
```

这一行代码是函数头，其中 main 是函数的名字，表示“主函数”，main 前面的 int 表示函数的返回值是 int 类型（整型）。每个 C 语言程序都必须有一个 main 函数。

第 3 行到第 6 行：

```
{
    printf("This is my first program. \n");
    return 0;
}
```

由花括号 {} 括起来的部分是函数体，该程序主函数的函数体由两条语句构成，每条语句后都要加分号，表示语句结束。其中 printf 是 C 编译系统提供的函数库中的输出函数，用来在屏幕输出内容，输出语句中双引号中间可以是字母、符号及中文字符等；“return 0;”的作用是当 main 函数执行结束前将整数 0 作为函数值，返回到调用函数处。

在程序各行的右侧都可以看到一段关于这行代码的文字描述（用/\*和\*/括起来），称为代码注释。其作用是对代码进行解释说明，便于日后自己阅读或者他人阅读源程序时理解程序代码含义和程序设计思路。



### 提示

字符串末尾的“\n”是 C 语言中规定的一个特殊符号，作为控制代码，其作用是回车换行。本程序中“\n”使得输出“This is my first program.”后执行一个回车换行操作，如果之后还有输出，将从下一行的左端开始输出。



【例 1.2】 计算给定长、宽后的矩形面积。

```
#include <stdio.h> /*编译预处理指令*/
int main( ) /*主函数的函数头*/
{
    int length, width, area; /*定义变量*/
    length=25; /*给 length 赋值为 25*/
    width=10; /*给 width 赋值为 10*/
    area=length*width; /*计算面积*/
    printf("The area equals %d",area); /*输出面积的值*/
    return 0; /*程序返回值 0*/
}
```

程序中使用的变量是 length, width, area。在 C 语言里, 每个变量的数据类型必须在函数中事先定义。

第 4 行:

```
int length, width, area;
```

定义了 3 个变量均为整数。int 是关键字, 用以说明整型数据。在函数体的左大括号之后, 必须马上给出变量定义, 变量在定义之前不能使用。



### 提示

在计算机语言中, 关键字是为表达某些概念或操作而特设的保留字, 它们不能用作变量名。

第 5 行到第 7 行:

```
length=25;
width=10;
area=length*width;
```

给变量 length 和 width 赋值, 再通过表达式 “length\*width” 计算面积 (星号代表乘法运算), 并将其值赋给变量 area, 因此, 此例的输出面积等于 250。

第 8 行:

```
printf("The area equals %d",area);
```

这里 printf() 函数既输出数据, 又输出文本, 而例 1.1 中的 printf() 函数只输出文本。此处的输出不仅包含了信息, 还包含有变量 area 的数值。字符序列 “%d” 是格式控制符, 它不仅说明要打印的数据是一个十进制整数, 还明确指出了插入数据的位置 (此例是在 equals 之后)。变量 area 出现在 printf() 中信息部分的后面, 说明在 “%d” 位置处应填入的变量值。

该程序运行结果为:

```
The area equals 250
```

【例 1.3】 通过 while 语句打印 1~5 的平方数。



```

#include<stdio.h>          /*编译预处理指令*/
int main()                /*主函数的函数头*/
{
    int i;                /*定义变量*/
    i=1;                  /*变量赋值*/
    while(i<=5)          /*当 i 小于等于 5 进行循环*/
    {
        printf("%d\t%d\n",i, i*i); /*输出 i 和 i 的平方的值*/
        i=i+1;           /*i 自加 1*/
    }
    return 0;            /*程序返回值 0*/
}

```

`i` 表示需要计算平方数的变量，此处 `i` 以初值 1 进入平方计算，如果希望平方数开始于其他数值，`i` 需做相应调整。

只要 `i` 满足小于等于（条件小于等于在 C 语言中记作“`<=`”）5 的条件，`while` 就重复执行其后大括号内的所有语句。每执行一次 `while` 后的语句序列，就打印一次 `i` 和 `i` 的平方，同时 `i` 的值通过下列语句增 1：

```
i=i+1;
```

该语句可读作“设置 `i` 为其当前值加 1”。最后，当 `i` 增加到 6 时，测试条件（`i<=5`）不再满足，`while` 循环结束。

该程序运行结果为：

```

1  1
2  4
3  9
4  16
5  25

```

从上述 3 个示例中可以看出 C 语言程序的基本结构：

(1) C 语言程序由函数组成，而且必须有且仅有一个以 `main` 命名的主函数，其他被调用的函数可以是系统提供的库函数（如 `printf()` 函数），也可以是用户自定义函数。

(2) C 语言程序总是从 `main()` 函数开始执行，在执行过程中完成对其他函数的调用。

(3) 一个函数由两部分组成：① 函数头，包括函数名、函数类型、函数属性、函数形参名、形参类型。一个函数名后面必须跟一对圆括号，括号内可以有参数也可以没有参数，如 `main()`；② 函数体，即函数头下由花括号 `{}` 括起来的部分。如果一个函数内有多对大括号，则最外层的一对 `{}` 为函数体的范围。

(4) 任何变量必须先定义类型，后使用。

(5) C 程序书写格式自由，一行内可以连续写几个语句，一个语句也可以分写在多行中。但任何一条 C 语句必须以分号“`;`”结束。

(6) 程序中可以使用编译预处理命令。



(7) 程序中可以有注释(以“/\*”和“\*/”界定),用于解释程序的功能,增强可读性,但注释对程序的编译和运行不起作用。



### 提示

C 语言中允许用两种注释方式:

① 以//开始的单行注释。这种注释可以单独占一行,也可以出现在一行中其他内容的右侧。此种注释的范围从//开始,以换行符结束,即这种注释不能跨行。若注释内容一行内写不下,可以用多个单行注释。如:

```
printf("This is my first program. \n");    //输出要  
//显示的字符串
```

② 以/\*开始,以\*/结束的块式注释。这种注释可以单独占一行,也可以包含多行。编译系统在发现一个/\*后,会开始找注释结束符\*/,把二者间的内容作为注释。

(8) C 语言本身没有输入输出语句,输入输出操作是通过调用系统提供的标准库函数完成的,其目的是将与硬件设备直接有关的输入输出等操作交由库函数实现,使程序不直接对硬件设备进行管理,增加程序的可移植性。

(9) C 语言要求严格区分字母大小写,同一字母的大小写被视为两个不同的字符。C 程序中的语句、类型名、定义符、函数名、变量名等一般均用小写字母表示。

这里只对 C 语言程序的结构做了简单的介绍,读者只有学习更多的 C 语言知识,才能更全面地了解 C 语言程序的结构特点。

## 1.4 Visual Studio 2010 简介

Visual C++ 6.0 被称为史上最经典的 VC,现在还有很多企业使用它,大量的教材也都是基于这个版本的 VC 来阐述的。然而,实际上 Visual C++ 6.0 并没有想象中的那么美好,孱弱的 IDE、不完善的 STL 等使它越来越不适应时代的发展,同时由于后来的版本和它的差别越来越大,让很多习惯于 Visual C++ 6.0 的人在新版本上无所适从。2018 年 3 月的全国计算机等级考试中,C 语言考试环境也由 Visual C++ 6.0 更新为 Visual Studio 2010,因此,本教材也选择 Visual Studio 2010 作为书中程序的编译环境。

Visual Studio 2010(简称为 VS 2010)是微软公司于 2010 年推出的集成开发环境,相比 Visual C++ 6.0 而言,其界面被重新设计和组织,项目管理更加简单明了。Visual Studio 2010 也是一个多语言集成开发环境,支持 Visual C++、Visual Basic、C#和 ASP 等应用系统开发。对于学习 C 及 C++程序设计的人来说,Visual Studio 2010 完全是一个足够好的编程环境。

Visual Studio 2010 有多个不同版本,分别是:

(1) 专业版:面向个人开发人员,提供集成开发环境、开发平台支持、测试工具等,属于商业版本。



(2) 高级版：创建可扩展、高质量程序的完整工具包，相比专业版增加了数据库开发、Team Foundation Server (TFS)、调试与诊断、MSDN 订阅、程序生命周期管理 (ALM)，属于商业版本。

(3) 旗舰版：面向开发团队的综合性 ALM 工具，相比高级版增加了架构与建模、实验室管理等，属于商业版本。

(4) 学习版：一个免费的集成开发环境。从本质上讲，Visual Studio 2010 Express 是轻量级版本的集成开发环境，可在 Microsoft DreamSpark 上获得（必须有微软的 dreamspark），也可以通过邮箱等方式验证你的学生身份来获取该版本。

下面以 Visual Studio 2010 学习版为例进行介绍。

### 1.4.1 Visual Studio 2010 集成开发环境

Microsoft Visual Studio 2010 是基于 Microsoft 的应用程序集成开发环境，能够帮助开发人员快速地创建高质量的应用程序。

选择“开始”→“所有程序”→“Microsoft Visual Studio 2010 Express”→“Microsoft Visual C++ 2010 Express”菜单，打开 Visual C++ 2010，其集成开发环境包括了菜单栏、工具栏、工具箱、属性窗口、解决方案资源管理器等，如图 1-1 所示。

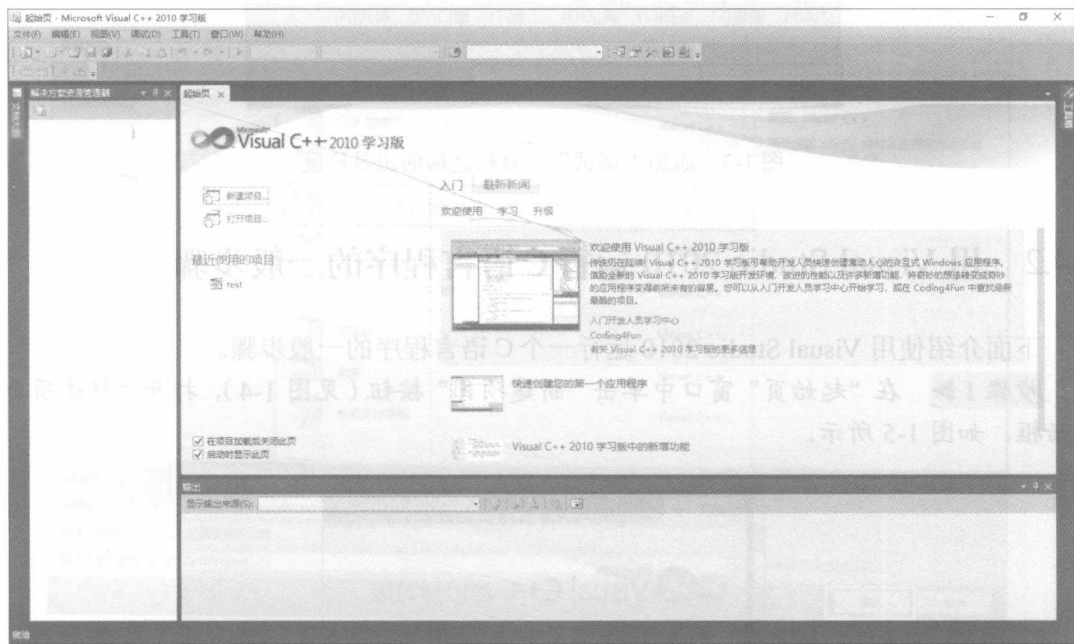


图 1-1 Visual Studio 2010 的主窗口

Visual C++ 2010 集成开发环境有一个默认起始页，使用起始页可以访问或创建项目，了解未来的产品版本和会议，或者阅读最新的开发文章。若要访问起始页，在“视图”菜单上单击“起始页”项即可。



选择“视图”→“工具栏”→“调试”菜单，可将“调试”工具栏添加到工具栏区中，如图 1-2 所示，添加之后的工具栏区如图 1-3 所示。

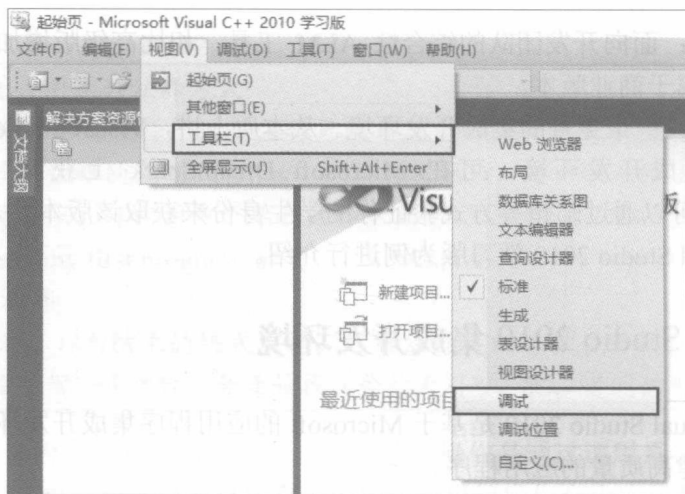


图 1-2 添加“调试”按钮

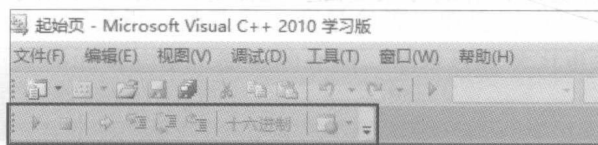


图 1-3 添加“调试”工具栏之后的工具栏区

### 1.4.2 用 Visual Studio 2010 运行 C 语言程序的一般步骤

下面介绍使用 Visual Studio 2010 运行一个 C 语言程序的一般步骤。

**步骤 1** 在“起始页”窗口中单击“新建项目”按钮（见图 1-4），打开“新建项目”对话框，如图 1-5 所示。

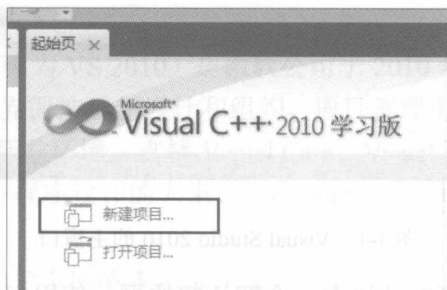


图 1-4 “起始页”窗口中“新建项目”按钮