



# C程序设计

---

段江涛

`jtduan@mail.xidian.edu.cn`

# 第 1 章

## 程序设计和C语言

# 计算机程序



## 指令

可以被计算机理解并执行的基本操作命令。



## 程序

一组计算机能识别和执行的指令。  
一个特定的指令序列用来完成一定的功能。



## 软件

与计算机系统操作有关的计算机程序、规程、规则，以及可能有的文件、文档及数据。

# 计算机语言

## 机器语言

计算机能直接识别和接受的二进制代码称为**机器指令**。机器指令的集合就是该计算机的**机器语言**。

特点：难学，难记，难检查，难修改，难以推广使用。依赖具体机器难以移植。

```
B8 7F 01
BB 21 02
03 D8
B8 1F 04
2B C3
```

## 汇编语言

机器语言的符号化。用英文字母和数字表示指令的**符号语言**。

特点：相比机器语言简单好记，但仍然难以普及。汇编指令需通过**汇编程序**转换为机器指令才能被计算机执行。依赖具体机器难以移植。

```
MOV AX 383
MOV BX 545
ADD BX AX
MOV AX 1055
SUB AX BX
```

## 高级语言

高级语言更接近于人们习惯使用的自然语言和数学语言。

特点：功能强大，不依赖于具体机器。用高级语言编写的**源程序**需要通过**编译程序**转换为机器指令的**目标程序**。

```
int x=1055, y = 383, z = 545
int S;
S = x-(y+z);
S=1055-(383+545)
```

# 高级语言的发展

非结构化的语言

01

面向对象的语言

03

02

结构化语言

规定：

程序必须由具有良好特性的基本结构(顺序结构、选择结构、循环结构)构成，程序中的流程不允许随意跳转，程序总是由上而下顺序执行各个基本结构。

特点：

程序结构清晰，易于编写、阅读和维护。

# C语言的发展



D. M. Ritchie

- 1972—1973年间，美国贝尔实验室的D.M.Ritchie 在B语言的基础上设计出了C语言。
- 最初的C语言只是为描述和实现UNIX操作系统提供一种工作语言而设计的。
- 随着UNIX的日益广泛使用，C语言也迅速得到推广。1978年以后，C语言先后移植到大、中、小和微型计算机上。C语言便很快风靡全世界，成为世界上应用最广泛的程序设计高级语言。
- 以UNIX第7版中的C语言编译程序为基础，1978年，Brian W.Kernighan和Dennis M.Ritchie 合著了影响深远的名著The C Programming Language，这本书中介绍的C语言成为后来广泛使用的C语言版本的基础，它是实际上第一个C语言标准。
- 1983年，美国国家标准协会(ANSI)，根据C语言问世以来各种版本对C语言的发展和扩充，制定了第一个C语言标准草案('83 ANSI C)。
- 1989年，ANSI公布了一个完整的C语言标准——ANSI X3.159—1989(常称为ANSI C或C 89)。
- 1990年，国际标准化组织ISO(International Standard Organization)接受C 89作为国际标准ISO/IEC 9899: 1990，它和ANSI的C 89基本上是相同的。
- 1999年，ISO又对C语言标准进行了修订，在基本保留原来的C语言特征的基础上，针对应用的需要，增加了一些功能，尤其是C++中的一些功能，并在2001年和2004年先后进行了两次技术修正，它被称为C 99，C 99是C 89的扩充。

目前由不同软件公司所提供的一些C语言编译系统并未完全实现C99建议的功能，它们多以C89为基础开发。

# C语言的特点

语言简洁、紧凑，使用方便、灵活

01

运算符丰富

02

数据类型丰富

03

具有结构化的控制语句  
用函数作为程序的模块单位，便于实现程序的模块化  
C语言是完全模块化和结构化的语言

04

语法限制不太严格，程序设计自由度大

05

06

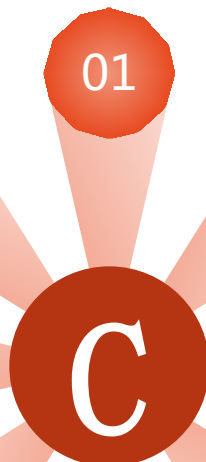
允许直接访问物理地址  
能进行位(bit)操作  
能实现汇编语言的大部分功能  
可以直接对硬件进行操作

07

程序可移植性好

08

生成目标代码质量高，程序执行效率高



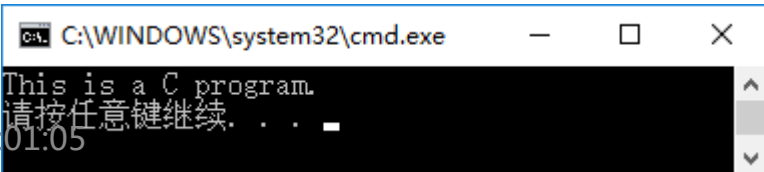
因此C语言既具有高级语言的功能，又具有低级语言的许多功能，可用来编写系统软件

# 最简单的C语言程序

## 【例1.1】要求在屏幕上输出：This is a C program.

解题思路：在主函数中用printf函数原样输出以上文字。

```
#include <stdio.h> // 编译预处理指令，包含标准库输入/输出函数说明
// standard input/output function
#include <stdlib.h> // standard library function, eg. for system() 函数
int main() // 定义主函数
{ // 函数开始的标志
    printf( "This is a C program.\n" ); // 输出所指定的一行信息
    system( "pause" ); // 窗口暂停，与编译器相关, [可选]
    return 0; // 函数执行完毕时返回函数值0
} // 函数结束的标志
```

A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe". The window contains the text "This is a C program." followed by a prompt "请按任意键继续. . .". The time "10:01:05" is visible in the bottom left corner of the window.

```
C:\WINDOWS\system32\cmd.exe
This is a C program.
请按任意键继续. . .
10:01:05
```



# 最简单的C语言程序

```
#include <stdio.h>           //编译预处理指令，包含标准库输入/输出函数说明
int main()                   //定义主函数
{                             //函数开始的标志
    printf("This is a C program.\n"); //输出所指定的一行信息
    return 0;                //函数执行完毕时返回函数值0
}                             //函数结束的标志
```



## 程序分析

- **main**是函数的名字，表示“主函数”；每一个C语言程序都必须有一个 `main` 函数。
- `main`前面的**int**表示此函数的类型是int类型(整型)，即在执行主函数后会得到一个值(即函数值)，其值为整型。
- **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。
- 函数体由花括号 **{ }** 括起来。

# 最简单的C语言程序

```
#include <stdio.h>           //编译预处理指令，包含标准库输入/输出函数说明
int main()                   //定义主函数
{                             //函数开始的标志
    printf("This is a C program.\n"); //输出所指定的一行信息
    return 0;                //函数执行完毕时返回函数值0
}                             //函数结束的标志
```



## 程序分析

- **printf**是C编译系统提供的函数库中的输出函数(详见第4章)。printf函数中**双引号**内的字符串" This is a C program." 按原样输出。**\n**是换行符，即在输出" This is a C program." 后，显示屏上的光标位置移到下一行的开头。
- 每个语句最后都有一个**分号**，表示语句结束。

# 最简单的C语言程序

```
#include <stdio.h>           //编译预处理指令，包含标准库输入/输出函数说明
int main()                   //定义主函数
{                             //函数开始的标志
    printf("This is a C program.\n"); //输出所指定的一行信息
    return 0;                //函数执行完毕时返回函数值0
}                             //函数结束的标志
```



## 程序分析

- 在使用函数库中的输入输出函数时，编译系统要求程序提供有关此函数的信息，程序第1行“**#include <stdio.h>**”的作用就是用来提供这些信息的。**stdio.h**是系统提供的一个文件名，**stdio**是standard input & output的缩写，文件后缀**.h**的意思是头文件(header file)，因为这些文件都是放在程序各文件模块的开头的。输入输出函数的相关信息已事先放在stdio.h文件中。

# 最简单的C语言程序

```
#include <stdio.h>
int main()
{
    printf( "This is a C program.\n" );
    return 0;
}
```

```
//编译预处理指令，包含标准库输入/输出函数说明
//定义主函数
//函数开始的标志
//输出所指定的一行信息
//函数执行完毕时返回函数值0
//函数结束的标志
```



## 程序分析

- `//`表示从此处到本行结束是“注释”，用来对程序有关部分进行必要的说明。在写C程序时应当多用注释，以方便自己和别人理解程序各部分的作用。在程序进行预编译处理时将每个注释替换为一个空格，因此在编译时注释部分不产生目标代码，注释对运行不起作用。注释只是给人看的，而不是让计算机执行的。

# 注释

## 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。如果注释内容一行内写不下，可以用多个单行注释。

```
//第一行注释  
//继续注释
```

```
/*一整块都是  
注释*/
```

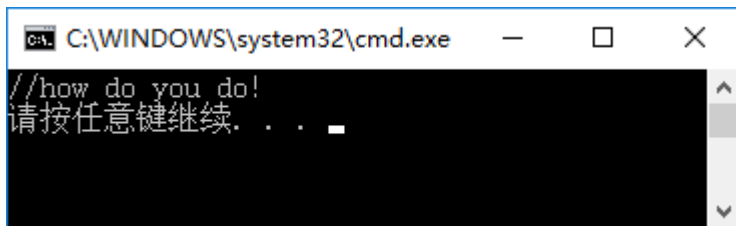
## 以/\*开始，以\*/结束的块式注释

这种注释可以包含多行内容。它可以单独占一行（在行开头以/\*开始，行末以\*/结束），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。

注意：在字符串中的//和/\*都不作为注释的开始。而是作为字符串的一部分。

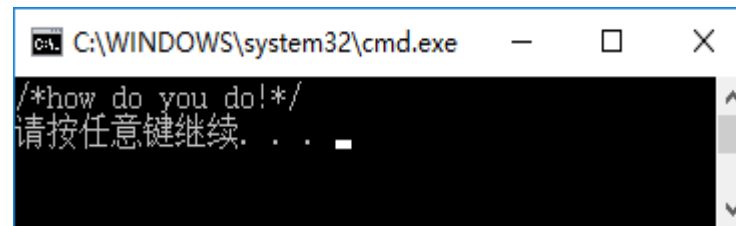
# 注释

```
#include <stdio.h>
int main()
{
    printf("//how do you do!\n");
    return 0;
}
```



A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has a black background with white text. The first line shows the output of the program: "//how do you do!". The second line shows the prompt "请按任意键继续. . . ." followed by a cursor.

```
#include <stdio.h>
int main()
{
    printf("/*how do you do!*/\n");
    return 0;
}
```

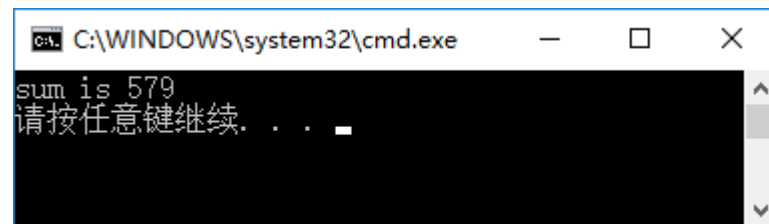


A screenshot of a Windows command prompt window titled "C:\WINDOWS\system32\cmd.exe". The window has a black background with white text. The first line shows the output of the program: "/\*how do you do!\*/". The second line shows the prompt "请按任意键继续. . . ." followed by a cursor.

# 最简单的C语言程序

## 【例1.2】求两个整数之和

**解题思路：** 设置3个变量，a和b用来存放两个整数，sum用来存放和数。用赋值运算符“=”把相加的结果传送给sum。



```
C:\WINDOWS\system32\cmd.exe
sum is 579
请按任意键继续. . .
```

```
#include <stdio.h>           //这是编译预处理指令
int main( )                 //定义主函数
{                             //函数开始
    int a,b,sum;             //本行是程序的声明部分，定义a, b, sum为整型变量
    a=123;                   //对变量a赋值
    b=456;                   //对变量b赋值
    sum=a+b;                 //进行a+b的运算，并把结果存放在变量sum中
    printf("sum is %d\n",sum); //输出结果
    return 0;                //使函数返回值为0
}                             //函数结束
```

# 最简单的C语言程序

```
#include <stdio.h>           //这是编译预处理指令
int main( )                  //定义主函数
{                             //函数开始
    int a, b, sum;           //本行是程序的声明部分，定义a, b, sum为整型变量
    a=123;                   //对变量a赋值
    b=456;                   //对变量b赋值
    sum=a+b;                 //进行a+b的运算，并把结果存放在变量sum中
    printf("sum is %d\n", sum); //输出结果
    return 0;                //使函数返回值为0
}                             //函数结束
```

输出时用sum的值取代%d

printf("sum is %d\n", sum);

A red arrow points from the text '输出时用sum的值取代%d' to the '%d' in the printf statement, and another red arrow points from the same text to the 'sum' variable in the same statement.

## 程序分析

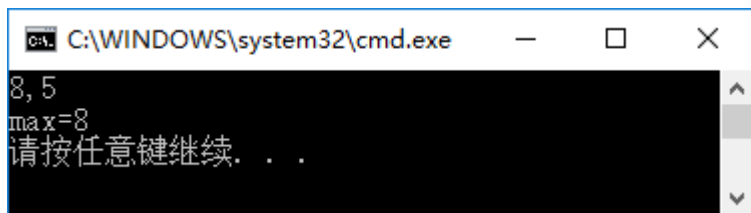
- **printf("sum is %d\n", sum);** printf函数圆括号内有两个参数。第一个参数是双引号中的内容sum is %d\n，它是输出格式字符串，作用是输出用户希望输出的字符和输出的格式。其中sum is是用户希望输出的字符，**%d**是指定的输出格式，**d**表示用“十进制整数”形式输出。圆括号内第二个参数sum表示要输出变量sum的值。在执行printf函数时，将sum变量的值(以十进制整数表示)取代双引号中的%d。



# 最简单的C语言程序

## 【例1.3】求两个整数中的较大者

**解题思路：** 用一个函数来实现求两个整数中的较大者。在主函数中调用此函数并输出结果。



```
C:\WINDOWS\system32\cmd.exe
8, 5
max=8
请按任意键继续. . .
```

```
#include <stdio.h>
//主函数
int main()
{
    int max(int x, int y); //定义主函数
    //主函数体开始
    int a, b, c; //对被调用函数max的声明
    //定义变量a, b, c
    scanf("%d, %d", &a, &b); //输入变量a和b的值
    c=max(a, b); //调用max函数，将得到的值赋给c
    printf("max=%d\n", c); //输出c的值
    return 0; //返回函数值为0
} //主函数体结束

//求两个整数中的较大者的max函数
int max(int x, int y) //定义max函数，函数值为整型，形式参数x和y为整型
{
    int z; //max函数中的声明部分，定义本函数中用到的变量z为整型
    if(x>y) z=x; //若x>y成立，将x的值赋给变量z
    else z=y; //否则(即x>y不成立)，将y的值赋给变量z
    return(z); //将z的值作为max函数值，返回到调用max函数的位置
}
```

# 最简单的C语言程序

```
#include <stdio.h>
//主函数
int main()
{
    int max(int x, int y); //定义主函数
    //主函数体开始
    int a, b, c; //对被调用函数max的声明
    //定义变量a, b, c
    scanf( "%d, %d" , &a, &b); //输入变量a和b的值，非格式符原样输入
    c=max(a, b); //调用max函数，将得到的值赋给c
    printf("max=%d\n", c); //输出c的值
    return 0; //返回函数值为0
} //主函数体结束

//求两个整数中的较大者的max函数
int max(int x, int y) //定义max函数, 函数值为整型，形式参数x和y为整型
{
    int z; //max函数中的声明部分，定义本函数中用到的变量z为整型
    if(x>y) z=x; //若x>y成立，将x的值赋给变量z
    else z=y; //否则(即x>y不成立)，将y的值赋给变量z
    return(z); //将z的值作为max函数值，返回到调用max函数的位置
}
```



## 程序分析

- 本程序包括两个函数：①主函数main；②被调用的函数max。
- max函数的作用是将x和y中较大者的值赋给变量z，最后通过return语句将z的值作为max的函数值返回给调用max函数的主函数。
- **scanf**是输入函数的名字(scanf和printf都是C的标准输入输出函数)。该scanf函数的作用是输入变量a和b的值。
- **max(a, b)**调用max函数。在调用时将a和b作为max函数的**实际参数**的值分别传送给max函数中的**形式参数**x和y。

# 最简单的C语言程序

```
#include <stdio.h>
//主函数
int main()                //定义主函数
{                          //主函数体开始
    int max(int x,int y);  //对被调用函数max的声明
    int a,b,c;            //定义变量a, b, c
    scanf("%d,%d",&a,&b);  //输入变量a和b的值
    c=max(a,b);           //调用max函数, 将得到的值赋给c
    printf("max=%d\n",c); //输出c的值
    return 0;            //返回函数值为0
} //主函数体结束

//求两个整数中的较大者的max函数
int max(int x,int y) //定义max函数, 函数值为整型, 形式参数x和y为整型
{
    int z;            //max函数中的声明部分, 定义本函数中用到的变量z为整型
    if(x>y) z=x;      //若x>y成立, 将x的值赋给变量z
    else z=y;        //否则(即x>y不成立), 将y的值赋给变量z
    return(z);       //将z的值作为max函数值, 返回到调用max函数的位置
}
```

注意：本例程序中两个函数都有return语句，请注意它们的异同。

两个函数都定义为整型，都有函数值，都需要用return语句为函数指定返回值。

main函数中的return语句指定的返回值一般为0。

max函数的返回值是max函数中求出的两数中的最大值z，只有通过return语句才能把求出的z值作为函数的值并返回调用它的main函数中。

# C语言程序的结构

- 一个程序由一个或多个源程序文件组成
  - 源程序文件包括：预处理指令、全局声明、函数定义
- 函数是C程序的主要组成部分
  - 一个C语言程序是由一个或多个函数组成的，其中必须包含唯一——一个main函数
  - 程序中被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己编制设计的函数
- 一个函数包括两个部分：**函数头部和函数体**，函数体一般包括声明部分和执行部分
- 程序总是从main函数开始执行
- 程序中的操作是由函数中的C语句完成的
- 在每个数据声明和语句的最后必须有一个分号
- C语言本身不提供输入输出语句，输入输出操作由函数完成
- 程序应当包含注释

```
#include <stdio.h>

int main()
{
    int max(int x, int y);
    int a, b, c;
    scanf("%d, %d", &a, &b);
    c = max(a, b);
    printf("max=%d\n", c);
    return 0;
}

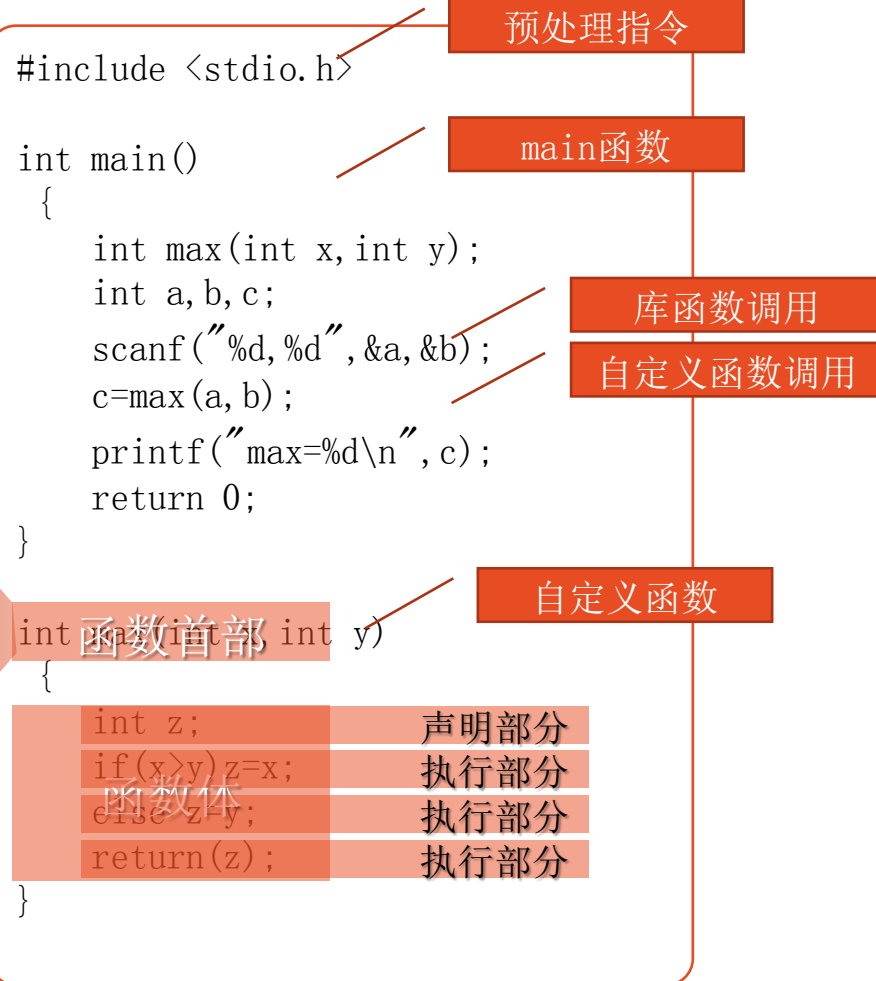
int max(int x, int y)
{
    int z;           声明部分
    if(x > y) z = x;  执行部分
    else z = y;      执行部分
    return(z);       执行部分
}
```

# C语言程序的结构

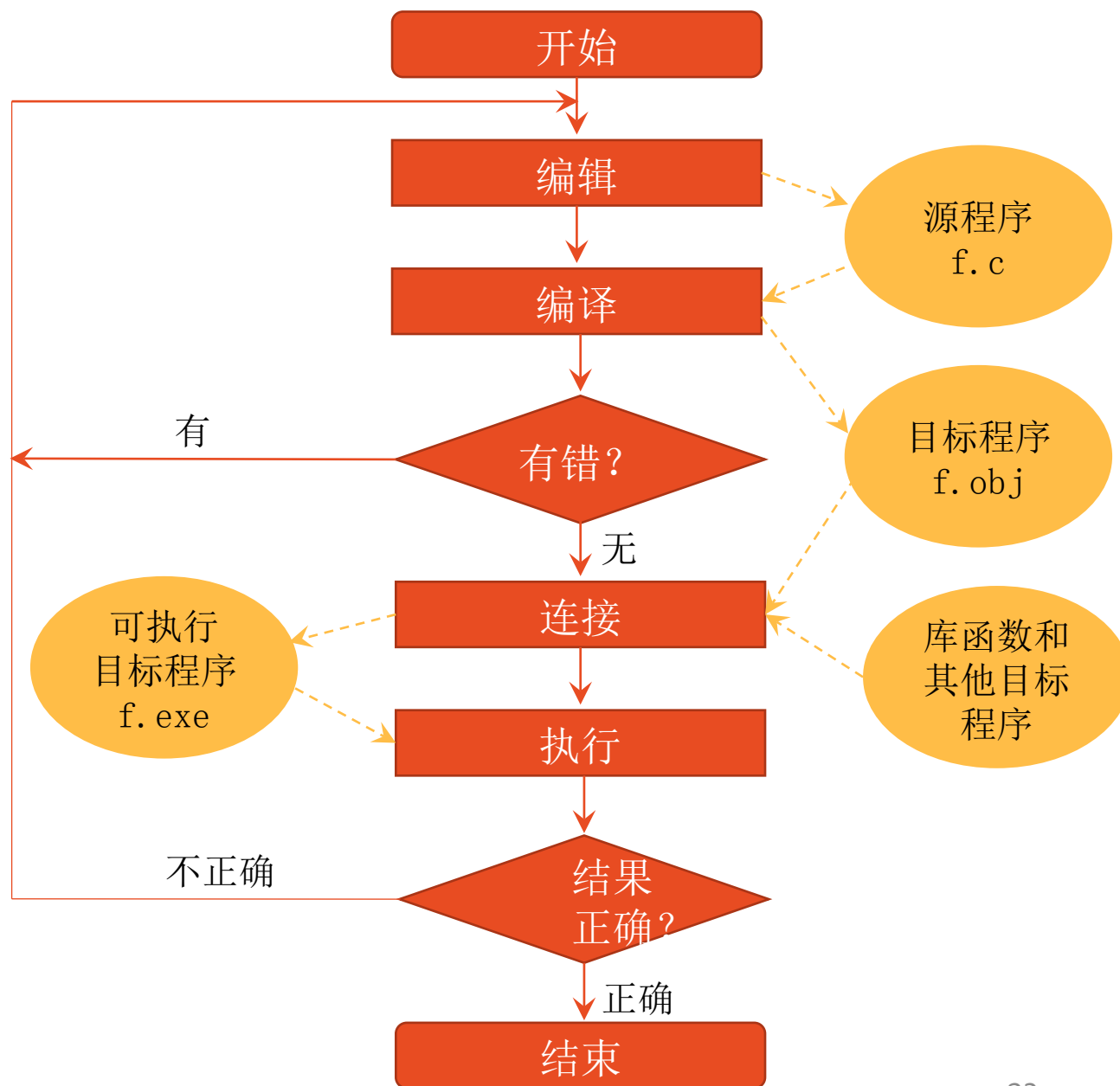
- 一个程序由一个或多个源程序文件组成
  - 源程序文件包括：预处理指令、全局声明、函数定义
- 函数是C程序的主要组成部分
  - 一个C语言程序是由一个或多个函数组成的，其中必须包含唯一一个main函数
  - 程序中被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己编制设计的函数
- 一个C程序由函数首部、函数体和函数调用组成
- 程序首部由预处理指令、全局声明、函数定义组成
- 程序首部由函数首部、函数体和函数调用组成
- 在每个数据声明和语句的最后必须有一个分号
- C语言本身不提供输入输出语句，输入输出操作由函数完成
- 程序应当包含注释

函数首部

int	max	(int	x,	int	y)
↓	↓	↓	↓	↓	↓
函数类型	函数名	参数类型	参数名	参数类型	参数名



# 运行C程序的步骤与方法



# 集成开发环境/编译系统

- **Bloodshed Dev-C++**
- **Turbo C**
- **Visual C++6.0**
- **Visual Studio (VS2015, VS Community 2017 等)**



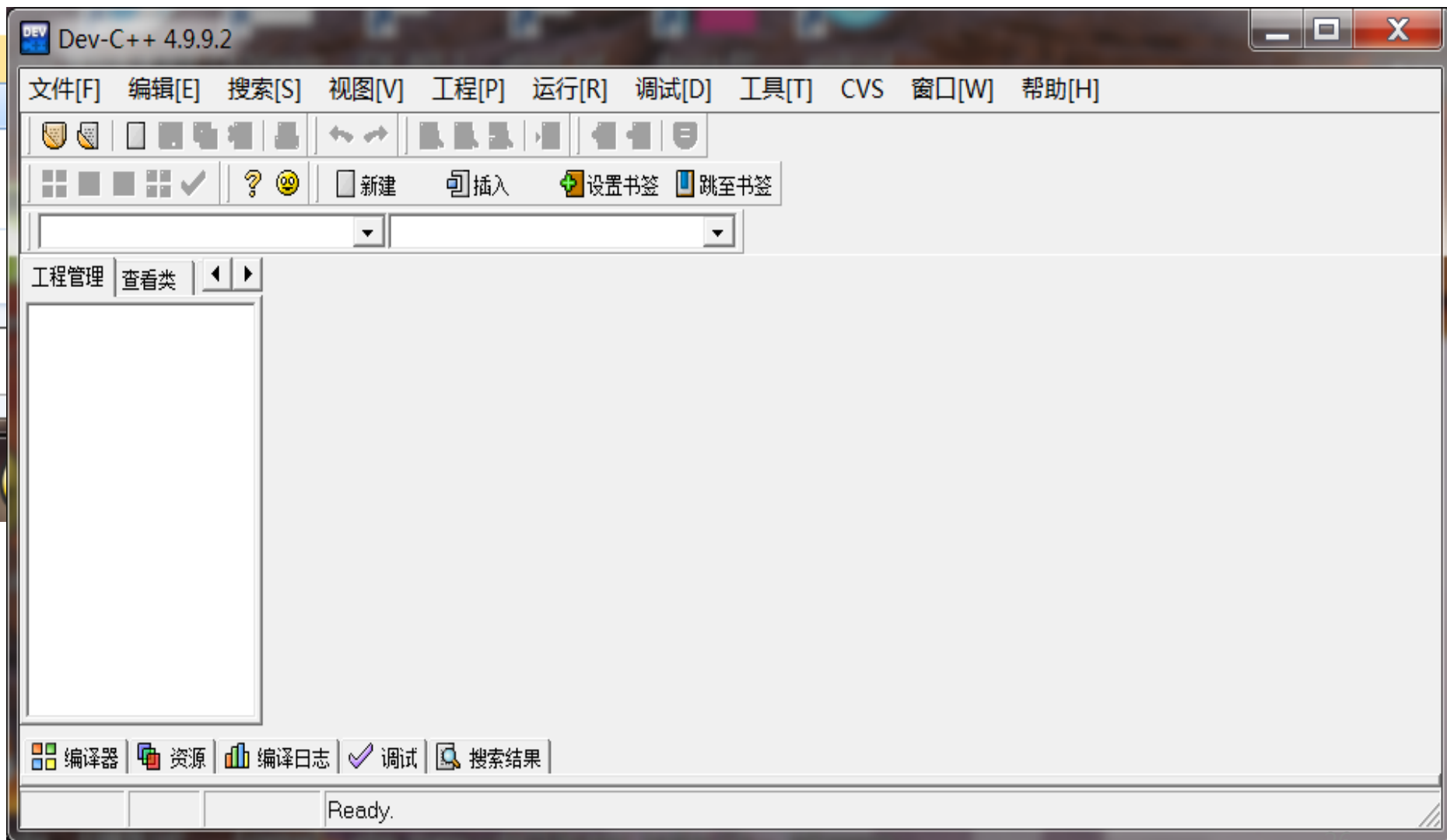
# Bloodshed Dev-C++集成开发环境

- **Bloodshed** 是一个著名的免费软件组织
- **Dev C++**是一个Windows环境下C/C++的集成开发环境(IDE)
- **GCC**编译器在Windows中的变种**MingW**或**Cygwin**
- 默认使用**MingW/TDM-GCC**编译器
- 遵循**C++ 11**标准，同时兼容**C++98**标准。
- 在工程编辑器中集成了编辑器、编译器、连接程序和执行程序。
- <http://www.bloodshed.net/index.html>



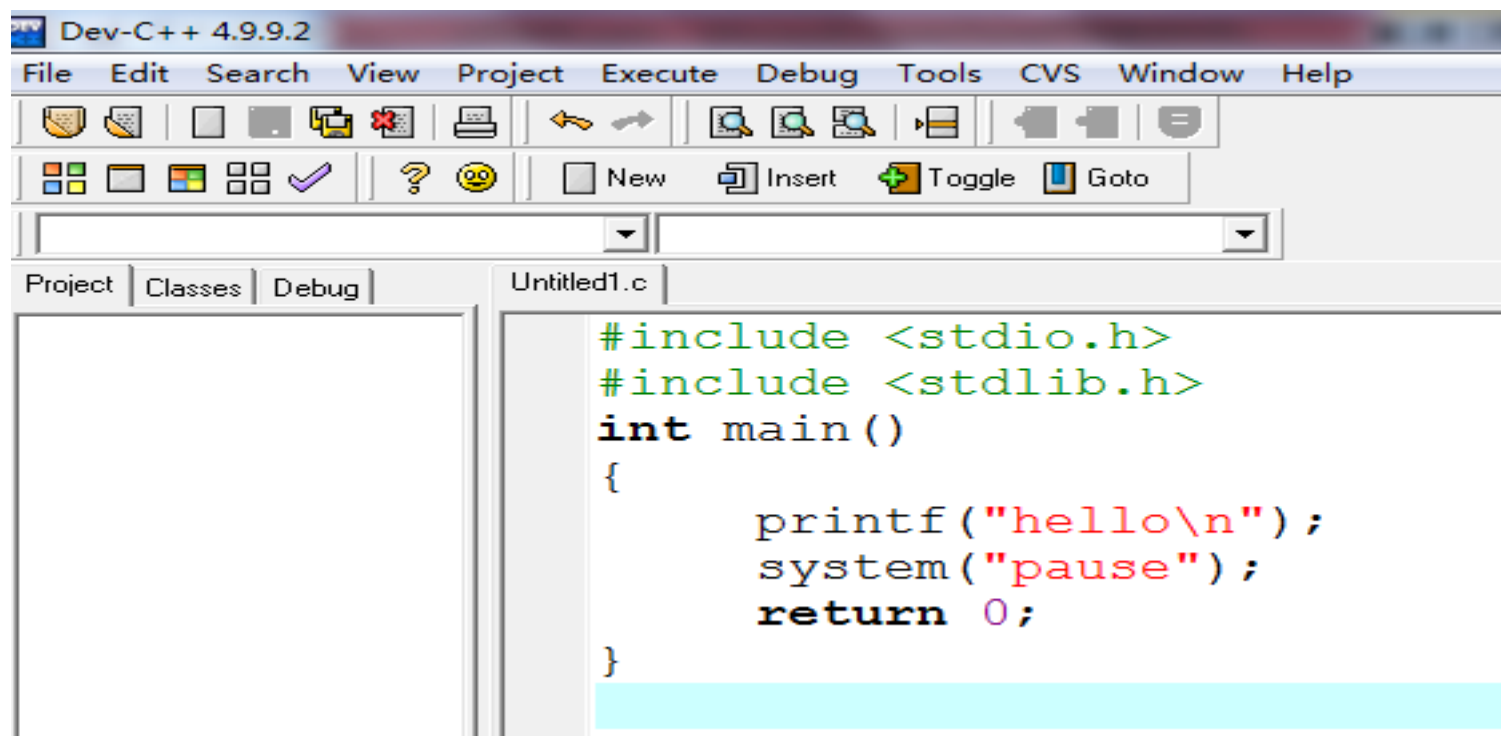
# Bloodshed Dev-C++集成开发环境

## ➤ 启动



# Bloodshed Dev-C++集成开发环境

- 选择“文件”菜单,选择“源文件”,编辑程序。
- 保存时,保存为 .cpp或 .c文件。如果选择 .cpp文件,main函数必须是int类型。
- 选择“编译和运行”菜单,生成.exe文件,运行程序。



# Bloodshed Dev-C++集成开发环境

```
#include <stdio.h> // for printf(),scanf()
#include <stdlib.h> // for system("pause");
int main(int argc, char* argv[])
{
    int a,b;
    scanf("%d%d",&a,&b); // 输入两个数值之间以空格分割
    printf("a+b=%d!\n",a+b); // “ \n” 回车(到第一个字符位置), 换行
    system("pause");        /* 保持显示程序运行结果的DOS窗口 */
    return 0;
}
```

- 调用输入输出函数, 需要头文件#include <stdio.h>
- 调用system函数, 需要头文件#include <stdlib.h>
- 程序中只能有一个主函数, 注意函数的返回类型与return保持一致

# 程序设计的任务

