



高级语言程序设计



主讲人：鲍娜娜



邮箱：nnbao@ahu.edu.cn



2025 C语言课程_周...

群号: 651161880



扫一扫二维码，加入群聊



星期三

高级语言程序设计

202520261-ZJ39020.003

(7~18周) (1-2节) 龙河校区 互联
大楼905 鲍娜娜

25级网络工程1班;25级网络工程2班
人数:80/80

高级语言程序设计

202520261-ZJ39020.003

(1~4,6~18周) (3-5节) 龙河校区 互
联大楼506 鲍娜娜

25级网络工程1班;25级网络工程2班
人数:80/80



程序设计语言

- 你有学过任何一种语言吗？
- 程序设计语言可以用来做什么？
 - ❧ 语言可以用来“沟通”
 - ❧ 与“计算机”“沟通”
 - ❧ 让计算机做你想要他做的事情

课程对象

- 没有程序设计经验/程序设计初学者
- 让初学者建立C语言的基础



课程简介及教学安排



学习要求：预习 独立编程 上机

1. 课前预习，认真听课，适当做笔记，课后消化、归纳。
2. 学会独立分析、完成所有作业，所有作业必须上机调试通过。
3. 上机实验注意**积累**编程经验、调试技巧。



教材及参考资料：

《C语言程序设计（第四版）》 李新华 梁栋等 中国电力出版社

《C语言程序设计习题解答与上机指导（第四版）》 李新华 梁栋 中国电力出版社

The C Programming Language (SE), Brian W. Kernighan, Dennis M. Ritchie, Prentice-Hall



一些建议：

1. 对待未知的、看似学会的知识、bugs等要有耐心；
2. 切忌眼高手低，渐进地成长，学会、熟悉以及融会贯通…



第1章 程序设计概述



2025-2026-1学期C语言教学进度表

| 周次 | 教学主要内容 | 学时 | 教学方式 |
|----|--------------------------------------|----|-------|
| 5 | 第1章 C程序设计概述【1.1~1.5】 | 3 | 授课、指导 |
| 6 | 第2章 数据及运算【2.1~2.7】 | 3 | 授课、指导 |
| 7 | 第2章 数据及运算【2.8~2.11, 2.13】 第2章 习题课 | 3 | 授课、指导 |
| 8 | 第3章 流程控制【3.1~3.2】 | 3 | 授课、指导 |
| 9 | 第3章 流程控制【3.3】 第3章 习题课 | 3 | 授课、指导 |
| 10 | 第4章 数组【4.1~4.2】 | 3 | 授课、指导 |
| 11 | 第4章 数组【4.3】 | 3 | 授课、指导 |
| 12 | 第4章 数组【4.4】 第4章 习题课 | 3 | 授课、指导 |
| 13 | 第5章 指针【5.1~5.3】 | 3 | 授课、指导 |
| 14 | 第5章 指针【5.4】 第5章 习题课 | 3 | 授课、指导 |
| 15 | 第6章 函数【6.1~6.3】 | 3 | 授课、指导 |
| 16 | 第6章 函数【6.5, 6.7~6.9】 第6章 习题课 | 3 | 授课、指导 |
| 17 | 第7章 结构体【7.1~7.5】 第7章 习题课 | 3 | 授课、指导 |
| 18 | 第8章 文件【8.1~8.4】 期末复习、答疑 | 3 | 授课、指导 |



高级语言程序设计学什么？

What?

计算思维



需求分析 数据组织 算法设计 编写代码 上机调试



程序设计就是写代码
实验就是上机写程序



第1章 程序设计概述



1. 了解程序和程序设计
2. 了解算法及N—S流程图表示法
3. 掌握C源程序的构成和书写规则
4. 掌握C语言标识符的概念
5. 掌握C语言基本数据输入输出的方法
6. 掌握VC++ 开发环境下C程序的开发

重点



1. 模仿例题，学会正确书写C源程序
2. 掌握VC开发环境下C程序的调试

难点



1.1 程序与程序设计

■ 1.1.1. 程序：

❖ 计算机的构成：

- 硬件—物质基础
- 软件—灵魂

❖ 程序：一组计算机指令的有序集合



■ 1.1.2. 程序设计

■ 1. 机器语言

- 第一代计算机语言，运算效率在所有语言中**最高**。
- 计算机的大脑——CPU所使用的指令是由“0”和“1”组成的二进制序列。用机器语言编写程序，就是直接书写CPU指令。虽然计算机可以理解、执行，但这类程序实在太难写、难记、难理解了，而且，由于各种计算机的指令系统往往各不相同，所以，在一台计算机上执行的这类程序很难在另一台计算机上顺利运行，必须重新编写程序。



■ 2. 汇编语言

- ❖ 为了减轻使用机器语言编程的痛苦，人们开始尝试用一些简洁的“助记符号”来替代由0、1组成的CPU指令，比如，用“ADD A, B”代表 $A+B\rightarrow A$ ，用“SUB A, B”代表 $A-B\rightarrow A$ 等等，这样一来，不仅编写程序变得方便了，而且别人也能很容易读懂程序在干什么。
- ❖ 这就是第二代计算机语言——**汇编语言**。用汇编语言编写的程序被称之为源程序，是不能直接在计算机上运行，需要一个专门的程序专门负责将这些“助记符号”翻译成二进制的机器目标代码才能在计算机上运行。
- ❖ 虽然汇编语言仍然面向计算机硬件，移植性不好，但用汇编语言编写的程序精炼，执行效率非常高，能准确发挥计算机硬件的性能和优势，所以汇编语言至今仍是一种常用而强有力的软件开发工具。



■ 3. 高级语言

- ❖ 随着人类生产、生活的各个领域对计算机程序的需求不断增多，许多接
近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程
序能在所有机器上通用的第三代计算机语言——高级语言不断涌现，影
响较大、使用较普遍的有FORTRAN、BASIC、Pascal、C、C++、
Visual C++、VB、Delphi、JAVA等等。
- ❖ 刚开始开发软件时，大家都是各自为战，缺乏科学规范的系统规划与测
试、评估标准，这种个体手工作坊式的闭门造车导致了大批耗巨资开
发好的软件由于含有太多的错误而无法使用，甚至带来了巨大损失，这就
是在计算机发展史上最早出现的“软件危机”。人们认识到：程序的设
计应易于保证正确性，也便于验证正确性，高级语言从早期语言发展到
了面向过程的结构化程序设计语言。
- ❖ 用高级语言编写的源程序需要将其翻译成目标代码才能在计算机上运行。
完成编译任务的程序又分为两类：一类是编译一句，执行一句，被称之为“解释程序”；一类是先将源程序全部翻译成目标代码以后，直接执
行在磁盘上产生的目标程序。而解释程序不会在磁盘上产生目标程序。



■ 1.1.3 C语言的发展

- ❖ 1972年 Dennis Ritchie&Brian Kernighan开始开发C语言；
- ❖ 1978年出版了名著《C程序设计语言(The C Programming Language)》；
- ❖ 1989年美国国家标准局制定了ANSI C标准，俗称C89；
- ❖ 1990年国际标准组织制定了ISO C标准，俗称C90；
- ❖ 1999年国际标准组织制定了新的ISO C标准，俗称C99；
- ❖ 2011年国际标准组织制定了新的ISO C标准，俗称C11；



■ C语言的主要特点：

C语言既可直接操作计算机硬件（汇编语言特性），可读性和移植性也很好（高级语言特性）。

语言简洁、紧凑，使用方便、灵活。

- (1) 软硬兼顾。（代码质量高，程序效率高）
- (2) 结构化程序设计。（代码和数据分离）
- (3) 数据类型丰富。（指针统一！）
- (4) 运算符极其丰富。（34种运算、15种优先级）
- (5) 移植性好。（较之汇编语言）



1.2 算法及其描述

■ 1.2.1 算法：程序设计的方法和步骤

程序=算法+数据结构

程序设计先要分析待加工的数据对象，通过在程序中指定数据的类型和数据的组织形式来描述数据对象。而算法则是程序设计中为解决实际问题而采取的方法和步骤。

程序设计的主要任务就是分析问题、提出解决问题的算法。学习高级语言，一方面要熟练掌握语言的语法规则，这是实现算法的基础，另一方面必须训练自己的思维，锻炼自己分析问题、分解问题，归纳整理出算法，最终写出高质量的程序的能力。

【例1-1】 输入三个整数，然后输出其中最大的数。



【例1-1】输入三个整数，然后输出其中最大的数。

第一步：输入三个整数存放到变量**first**、**second**、**third**中。

第二步：将**first**与**second**进行比较，若**first**大，则将**first**的值放入**max**中，若**second**大，则将**second**的值放入**max**中。

第三步：将**max**与**third**进行比较，若**third**大，则将**third**的值放入**max**中。

第四步：在屏幕上输出**max**的值

算法虽然正确，但写起来实在是太繁琐
进一步改进的算法如下：

S1: 输入三个整数**first**、**second**、**third**

S2: **max = first**

S3: 若**second > max**，则**max = second**

S4: 若**third > max**，则**max = third**

S5: 输出**max**。



算法优劣的衡量标准：

- 1. 可行性：** 算法必须是可行的，即算法所描述的操作都是可以通过执行有限次已知基本运算来实现。
- 2. 有穷性：** 算法必须能在有穷时间内执行有穷步后完成。
- 3. 确定性：** 算法中每一条指令都必须有确切的含义，不存在二义性。算法只有一个入口和一个出口。



■ 1.2.2 算法的描述

描述算法的方法有很多，通常可以使用自然语言、传统流程图、N-S流程图等。

1、自然语言

例1-1自然语言算法描述：

第一步：输入三个整数存放到变量**first**、**second**、**third**中。

第二步：将**first**与**second**进行比较，若**first**大，则将**first**的值放入**max**中，若**second**大，则将**second**的值放入**max**中。

第三步：将**max**与**third**进行比较，若**third**大，则将**third**的值放入**max**中。

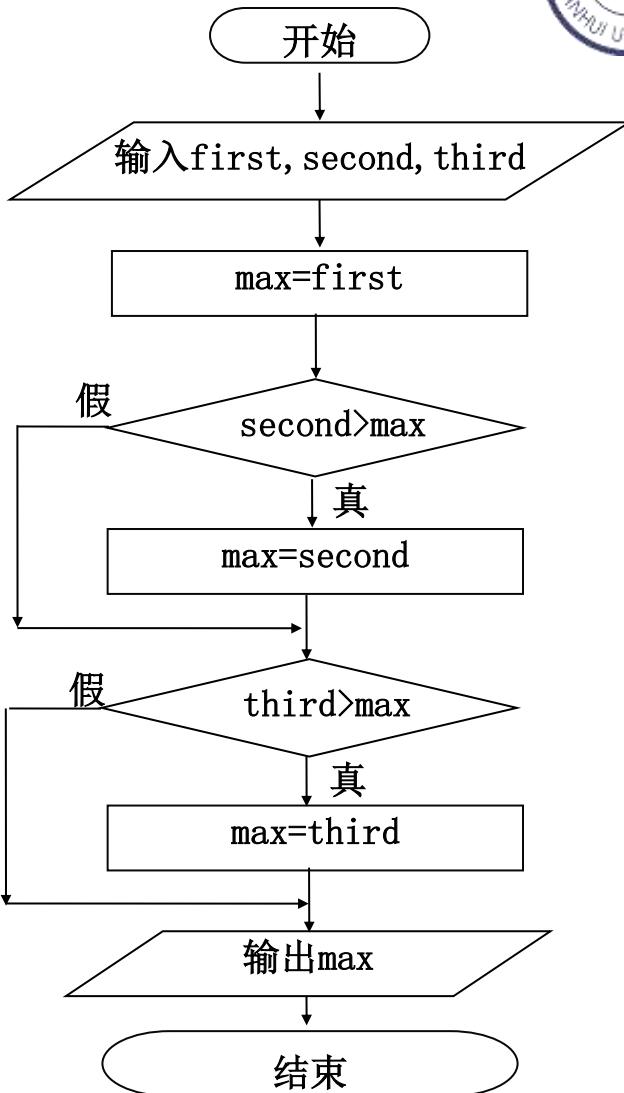
第四步：在屏幕上输出**max**的值



2、传统流程图

- 传统流程图利用不同形状的几何图形来代表各种不同性质的操作，用流程线来指示算法的执行方向，早期程序设计阶段广泛应用。

| 符号名称 | 图形 | 功能 |
|--------|-------|--------------|
| 起止框 | 椭圆 | 表示算法的开始和结束 |
| 输入/输出框 | 平行四边形 | 表示算法的输入/输出操作 |
| 处理框 | 矩形 | 表示算法中的各种处理操作 |
| 判断框 | 菱形 | 表示算法中的条件判断操作 |
| 流程线 | → | 表示算法的执行方向 |
| 连接点 | ○ | 表示流程图的延续 |



例1-1：传统流程图描述



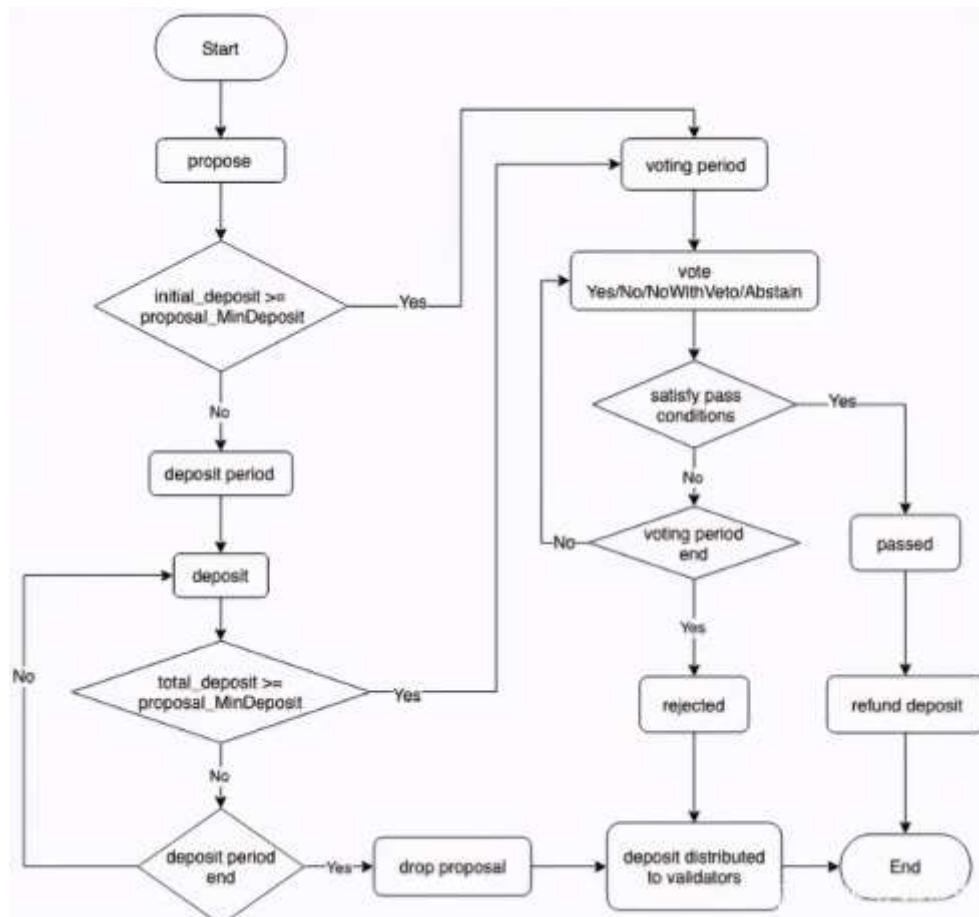
2、传统流程图

优点：

程序设计十分方便灵活，减少了人工复杂度。

缺点：

- ①用箭头随意地对控制进行转移，与结构化程序设计的精神相悖。
 - ②当目标系统较复杂时，流程图会变得复杂紊乱，难以看懂，也难以验证程序的正确性，如果有错，纠错十分困难。

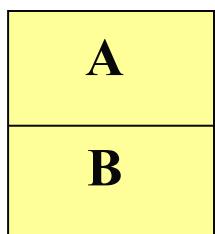


传统流程图示例

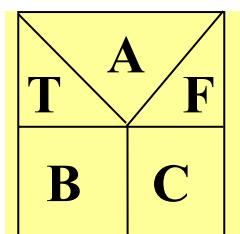


3、N-S流程图

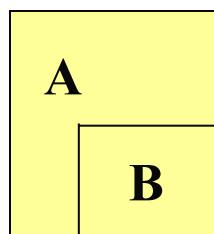
- 经过研究发现，任何复杂的算法都可以由顺序结构、选择结构和循环结构这三种基本结构组成，基本结构之间可以并列、可以相互包含，但不允许交叉，不允许从一个结构直接转到另一个结构的内部去。当整个算法都由三种基本结构组成时，算法结构清晰，易于书写，易于阅读，易于实现，易于纠错。
- 遵循这种方法的程序设计被称之为结构化程序设计。
- N-S流程图**是美国学者I.Nassi和B.Shneiderman提出的一种无流程线的结构化描述方法。**N - S**图描述的算法是一个大矩形框，大框内又包含若干基本结构的矩形框，三种基本结构的**N-S流程图**描述如下所示：



顺序



选择



循环

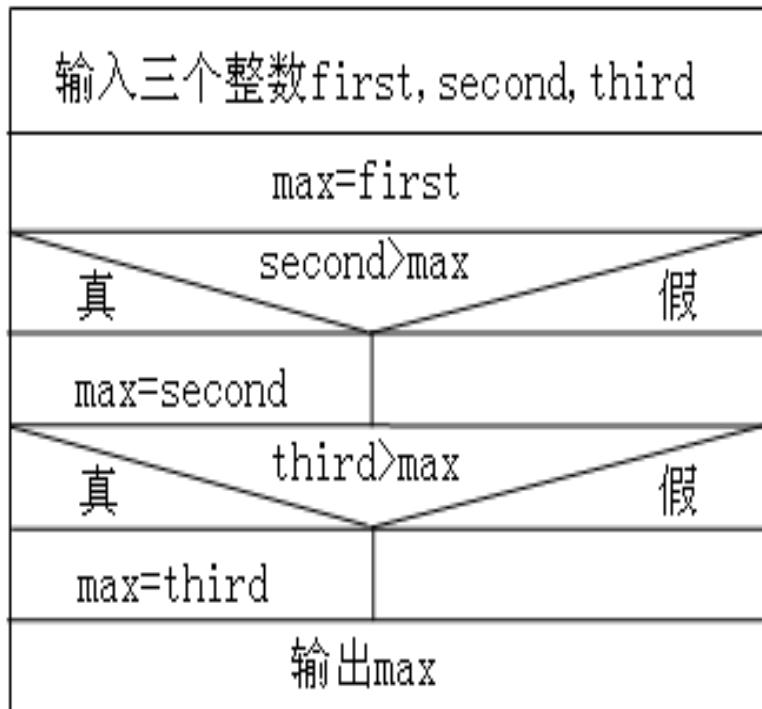


图1-5 【例1-1】的算法N-S流程图



1.3 C程序的构成和格式

学习C语言，首先要学习C语言特定的语法规则和规定的表达方法。一个严格按照C语言语法和表达方式的规定编写出来的C程序，不仅容易阅读、理解，还能保证编写的程序能在Visual C++环境中正确地执行。

一个标准的C程序结构一般包括以下内容：

```
/* 注释 */  
#include <.....>  
Global declarations;  
type functionname ( type variable, ... )  
{  
    local variable declarations;  
    statements;  
}  
void main( void )  
{  
    local variable declarations;  
    statements;  
}
```

文件包含说明
全局变量/函数说明
函数说明

函数体: 局部变量说明
函数体: 执行语句

主函数说明

函数体: 局部变量说明
函数体: 执行语句



C语言的程序长什么样？

```
#include <stdio.h>
int main() {
    printf("Hello world!\n");
    return 0;
}
```

文字编辑器
(text editor)

编辑 (edit)

编译器
(compile)

编译(compile)

可执行档
(executable file)

执行(execute)

源码
(source code)



【例1-3】简化一下【例1-1】的要求：输入两个整数，然后输出其中最大的数。

EG0103.c

```
/* Program: EG0103.c */
/* Description: Comparing first and second, see which is the bigger. */
#include <stdio.h> // 包含标准I/O库函数的说明
void main( void ) // 主函数说明
{
    int first, second, big; // 局部变量说明
    printf("请输入两个整数:"); // 显示输入提示文字
    scanf("%d,%d", &first, &second); // 从键盘输入数据存入变量
    if( first > second ) // 求大数
        big=first;
    else
        big=second;
}
```

运行结果

```
on "E:\LXH C Programming\EG01-03\Debug\EG01-03.exe"
请输入两个整数:19,68
第一个数是 19
第二个数是 68
大数是 68
Press any key to continue
```



解 释

1. 位于程序开始的“`/* */`”表示程序注释，用于说明算法、标明变量的用途，帮助其他人阅读、维护该程序。注释文字可以是英文，也可以是中文。注释不会增加可执行程序的大小。编译器在编译程序时会忽略所有注释。

我们可以根据需要在程序的任意位置添加注释，比如本例题中的程序注解和末尾的运行结果。通常在要解释的代码上方或右侧放置注释。

“`/* */`”被称为块注释或多行注释，书写时可以跨行；但“`/* */`”不能嵌套，这意味着在块注释文字中不能再次出现“`/*`”、“`*/`”。初学C语言很容易犯这种错误。

在Visual C++中有两种类型的注释，除了上述的块注释之外，还提供了以双斜线(`//`)开头的单行注释。单行注释中双斜线右侧的所有内容都是注释，它的作用范围只能到本行结束，不允许跨行。由于单行注释简洁明了，允许在注释文字中再次出现双斜线，不易出错，在现代C语言程序设计中得到了广泛应用。



解 释

2. “**#include <stdio.h>**”（或“**#include "stdio.h"**”）是预处理命令中的文件包含命令，其意义是把尖括号<>或引号""内指定的文件包含到本程序来，成为本程序的一部分。被包含的文件通常是由系统提供的，其扩展名为.h。因此也称为头文件或首部文件。

头文件名可以用双引号括起来，也可以用尖括号括起来。例如以下写法都是允许的：

#include <math.h>

#include "LXHAPI.h"

这两种形式是有区别的：使用**尖括号**表示在**VC**系统设定的头文件文件夹中去查找（一般是由**VC**在安装时设置的），而不在源文件文件夹去查找。这种格式**通常用于包含系统提供的头文件**；使用**双引号**则表示首先在源文件所在当前文件夹中查找，若未找到才到系统文件夹中去查找。**用户自定义头文件通常用这种格式**包含进来。

C语言的头文件中包括了编译系统提供的各个标准库函数的有关说明（函数原型）。如果在程序中调用了一个库函数时，就必须包含该函数原型所在的头文件。在本例中，使用了两个库函数：标准输入函数**scanf**，标准输出函数**printf**，其头文件为**stdio.h**，所以需要用**include**命令包含了**stdio.h**文件。



3. “**void main(void){.....}**”是C语言程序定义的主函数，**main**是主函数名，**main**后的括号()是函数的标志，括号()中的**void**表示没有入口参数，即调用**main**函数时不需要什么已知条件，**main**前的**void**表示调用完**main**函数不返回任何数据。**void main(void)**是主函数说明。“{.....}”中包含的是函数的函数体。
4. “**int first, second, big;**”定义了三个仅在**main**函数内部使用的能够存放整数的变量（**int**:整型变量），前两个用于存放输入的两个整数，**big**用于存放即将找到的大数。
5. “**printf("请输入两个整数:");**”调用了标准输出库函数**printf**，在屏幕上显示输入提示文字。



6. “`scanf("%d,%d", &first,&second);`”调用了标准输入库函数`scanf`从键盘输入两个整数，存入变量`first`、`second`对应的内存单元。输入格式字符串“`%d,%d`”表示要输入两个十进制整数，彼此之间用“,”分隔。取地址运算符“`&`”可以获得其后变量的内存单元地址。
7. “`if.....else.....`”是选择结构，将`first`与`second`进行比较，若`first`大，则将`first`的值放入`big`中，若`second`大，则将`second`的值放入`big`中。
8. 程序最后的三个`printf`调用负责在屏幕上显示程序运行结果。输出格式字符串中的“`%d`”表示在这个位置上输出紧跟输出格式字符串后面的变量的十进制整数值。“`\n`”是换行符，将本行以后要显示的文字换到下一行显示（回车换行）。



【例1-4】用C语言编程实现【例1-2】：输入四个浮点数，然后输出其中最小的数。

EG0104.c

```
/* Program: EG0104.c */  
/* Description: 比较输入的四个浮点数, 打印最小数 */  
#include <stdio.h> // 包含标准I/O库函数的说明  
void main( void ) // 主函数说明  
{  
    float first, second, third, fourth, min; // 局部变量说明  
    printf("请输入四个浮点数:"); // 显示输入提示文字  
    scanf("%f, %f, %f,%f", &first, &second, &third, &fourth); // 输入数据  
    min = first;  
    if( second < min ) //求first、second中的小数  
        min = second;  
    if( third < min )  
        min = third;  
    if( fourth < min )  
        min = fourth;  
    printf("%f、%f、%f、%f",  
          first, second, third, fourth); // 输出结果  
}
```

运行结果

```
C:\>E:\LXH C Programming\EG01-04\Debug\EG01-04.exe  
请输入四个浮点数:22.6,33.5,12.7,56.8  
22.600000、33.500000、12.700000、56.799999中的最小数是12.700000  
Press any key to continue
```



通过上述几个例题，总结 C 语言程序的以下几个特点：

- 1. 函数是 C 语言程序的基本单位。一个 C 程序至少包含一个**main()**函数或一个**main()**函数和若干个其它函数。
- 2. C 程序总是从**main()**函数开始执行。当主函数执行完毕时，亦即程序执行完毕。其它函数通过**main()**函数的调用得以执行。**main()**函数在程序中的位置任意。习惯上将**main()**放在最后面。
- 3. 除了预处理语句以外，所有 C 语句都必须以分号；结束。预处理语句通常放在 C 源程序的开始。
- 4. C 程序的书写格式自由。一行可以写几个语句，也可以将一个语句写好几行。但为了让程序看起来更加清晰，更容易阅读，建议一行写一条语句，不同结构层次的语句从不同的起始位置开始书写。低一层次的语句可比高一层次的语句缩进若干格后书写。按照锯齿状缩进书写程序是一个优秀程序员必备的素质。



注意：函数体内用{}括起来的部分，通常表示了程序的某一层次结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

```
/* 对a,b按从小到大排序 */  
if( a > b ) /* 若a > b, 交换 a, b的值 */  
{  
    int c=a;  
    a=b;  
    b=c;  
}
```

5. C程序的标识符有特殊的规定，详见“第一章第四节1.4中C语言标识符”。C语言习惯用小写字母，且大小写敏感。

6. 块注释/* */中 “/*”和 “*/” 必须成对使用，且 “/”和 “*” 、以及 “*” 和 “/” 之间不能有空格。行注释//.....中 “//”和 “/”之间不能有空格。

7. 程序的基本输入、输出可调用系统函数scanf()和printf()来完成。更进一步说明详见“第二章第九节 数据的输入输出”相关内容。



1.4 C语言标识符

1.4.1 C语言的字符集

字符是组成语言的最基本的元素。C语言字符集由字母，数字，空格，标点和特殊字符组成。

- ❖ 1. 字母。小写字母a~z共26个，大写字母A~Z共26个。
- ❖ 2. 数字。0~9共10个。
- ❖ 3. 分隔符：空白符和“，”。空格符、制表符、换行符等统称为空白符。空白符只在字符常量和字符串常量中起空白字符作用。在其它地方出现时，只起间隔作用，编译程序对它们忽略不计。分隔符“，”主要用于分隔同类项。
- ❖ 4. 运算符和特殊字符。C语言提供了极其丰富的运算符（详见附录B）。特殊字符是一些有着特定含义的符号：“{}”、“{}”、“/*”、“*/”、“//”等。

注意：

- ❖ 1. 在书写程序注释“/* */”、“//.....”和输入/输出格式字符串“.....”（字符串常量）中还可以使用汉字或其它特殊符号。
- ❖ 2. 书写程序必须严格使用C语言的字符集。比如书写程序时，不得使用中文标点（这点尤其要注意！！）



1.4 C语言标识符

1.4.2 标识符

定义：标识符是用来标识变量、常量、函数等程序操作对象的有效字符序列。

C语言对标识符有以下规定：

- 1. 标识符只能由英文字母、数字、下划线组成，且第一个字母必须是字母或下划线。

如first, Number1, num1, Computer_Book等标识符是合法的，而以下标识符是非法的：

2num 以数字开头，可改为num2

Computer Book 出现非法字符空格，可改为Computer_Book

Class-One 出现非法字符减号，可改为Class_One

- 2. ANSI C规定标识符最多可以使用31个字符作为有效长度。这意味着如果两个标识符前31个字符相同，即使后继字符不同，如

HeFeiNoFiveMiddleSchoolStudentName和

HeFeiNoFiveMiddleSchoolStudentNumber, C语言会把它们看成同一个标识符。

- 3. C语言大小写敏感。这意味着SUM和sum、Name和name表示四个独立的标识符。



4. C语言规定了下列标识符具有特殊含义（被称之为“关键字”）：

| | | | | |
|----------|---------|--------|----------|--------|
| 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 | | | |

如：if在C语言中表示选择结构，我们就不能用if来命名其他变量了。

5. 建议不要使用预定义标识符。C语言通过预处理定义了一些函数名和宏名：main、printf、define、RED等等。虽然可以强制赋予预定义标识符以新的含义，但这样容易引起误解。

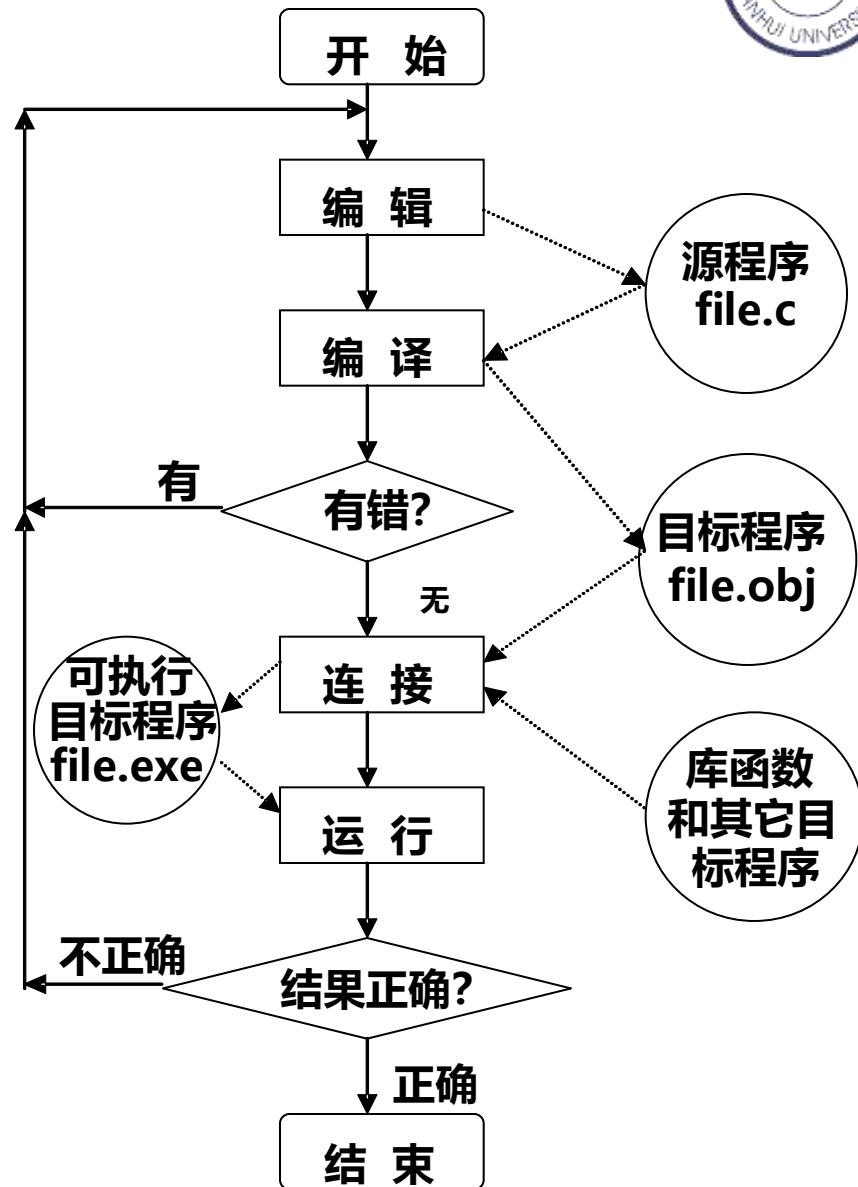
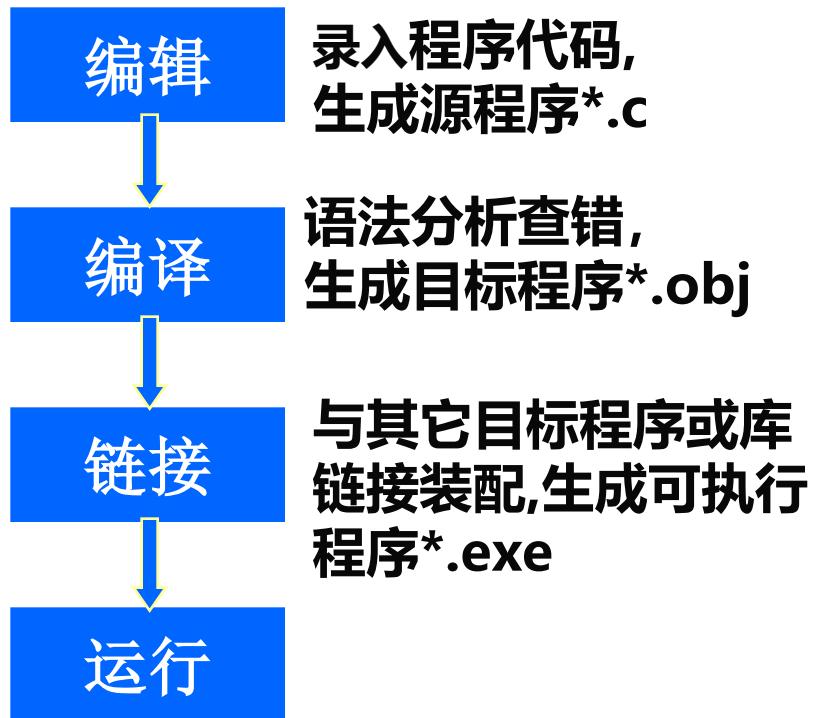


- ☞ 6. 用户自定义标识符是用户根据需要自行定义的标识符，通常用作变量名、函数名等，习惯上小写或单词的首字母大写，如`isleapyear`、`IsLeapYear`等等。它不仅不能和C语言的关键字、预定义标识符相同，也不能和用户已编制的函数同名。
- ☞ 7. 用户自定义标识符应尽可能做到“见名知意”，即选用英文单词、拼音或其缩写做标识符，如`name/xm`、`age/nl`、`salary/gz`等，尽可能避免使用容易引起混淆的单字符变量名：`a`、`b`、`c`、`x`、`y`、`i`、`j`等等。
- ☞ 8. 标识符应尽可能避免使用一些容易引起混淆的字符：数字1与字母I，数字0与Oo, 数字2与Zz等。



1.5 C程序的开发

一. 开发过程：





■ 二. VC上机步骤：

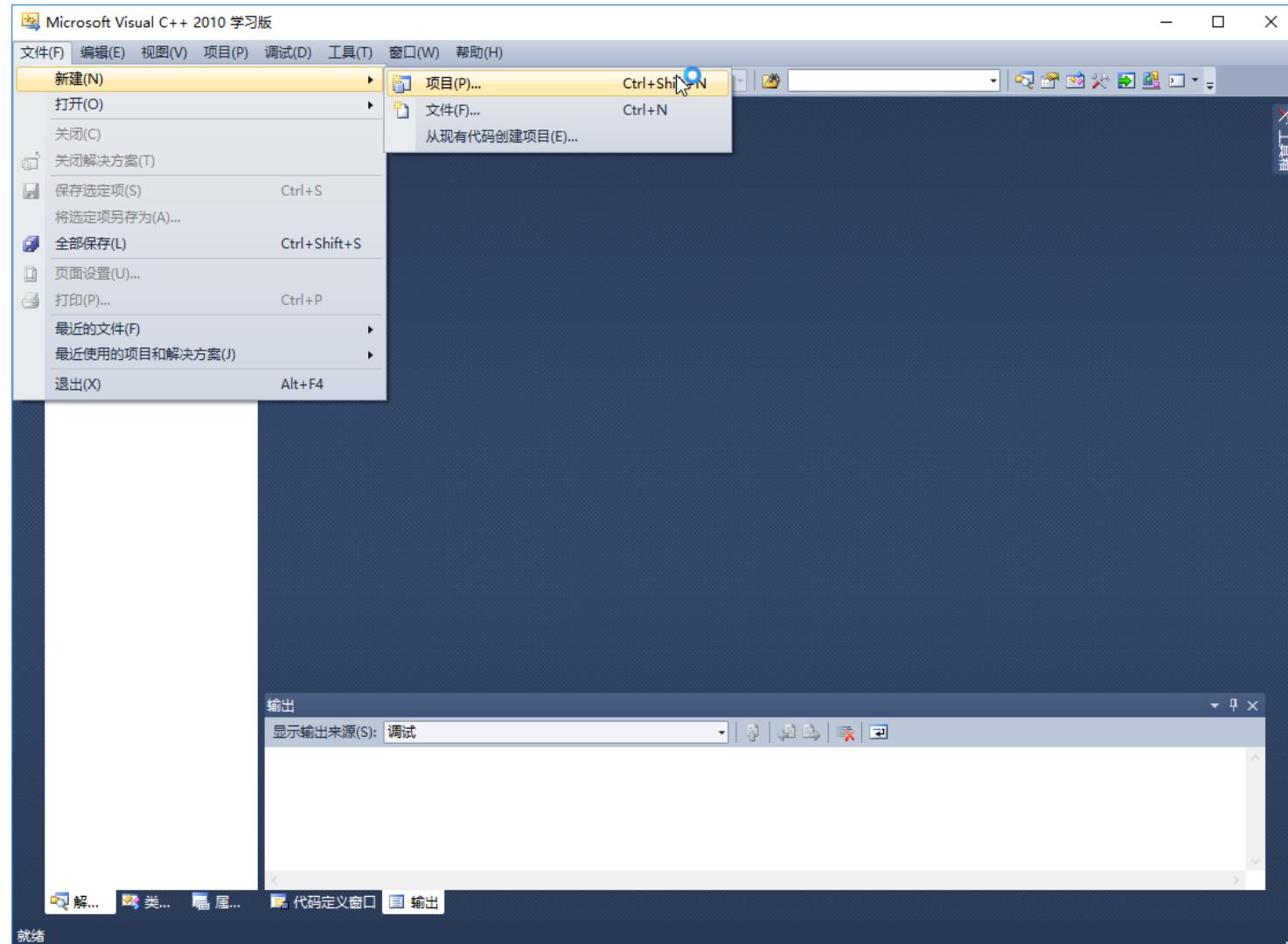
1 创建工作文件夹，如E:\LXH

2 启动VC++2010express

3 文件→新建→项目→Win32控制台应用程序 → eg0103

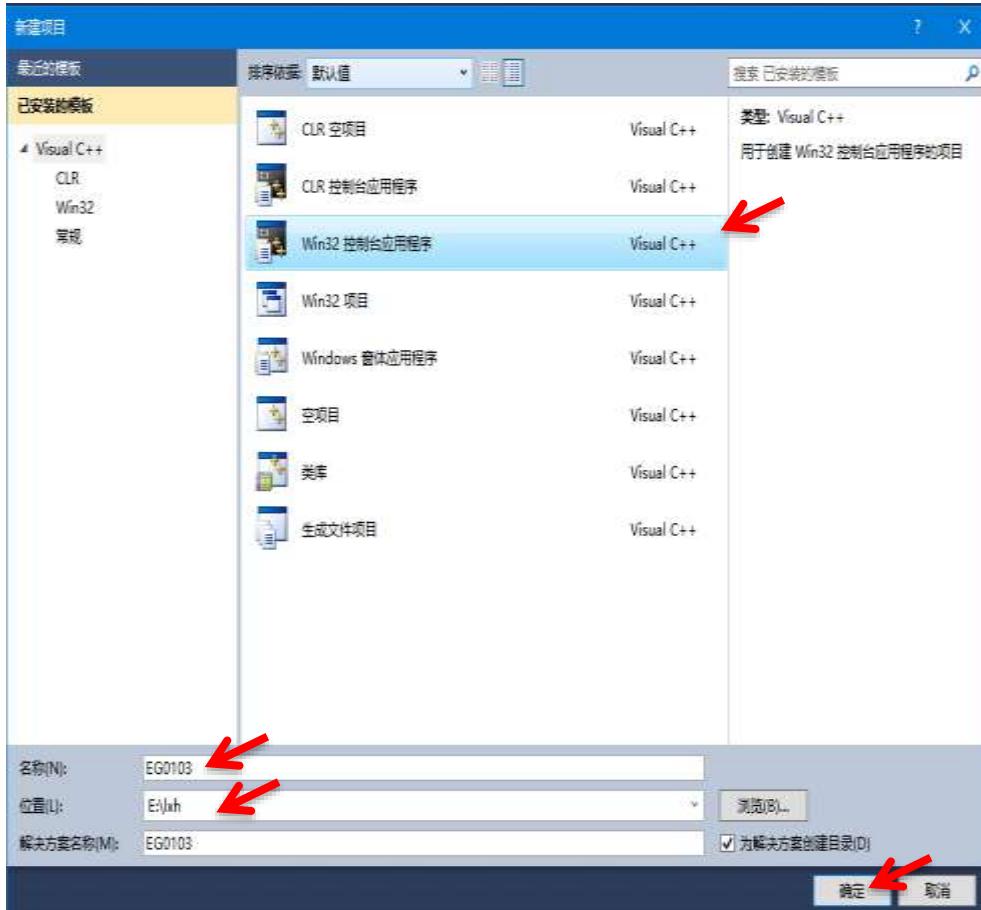


菜单栏-文件-新建-项目





新建项目，如E:\LXH\eg0103





Win32 应用程序向导 - EG0103

欢迎使用 Win32 应用程序向导



概述

应用程序设置

这些是当前项目设置：

- 控制台应用程序

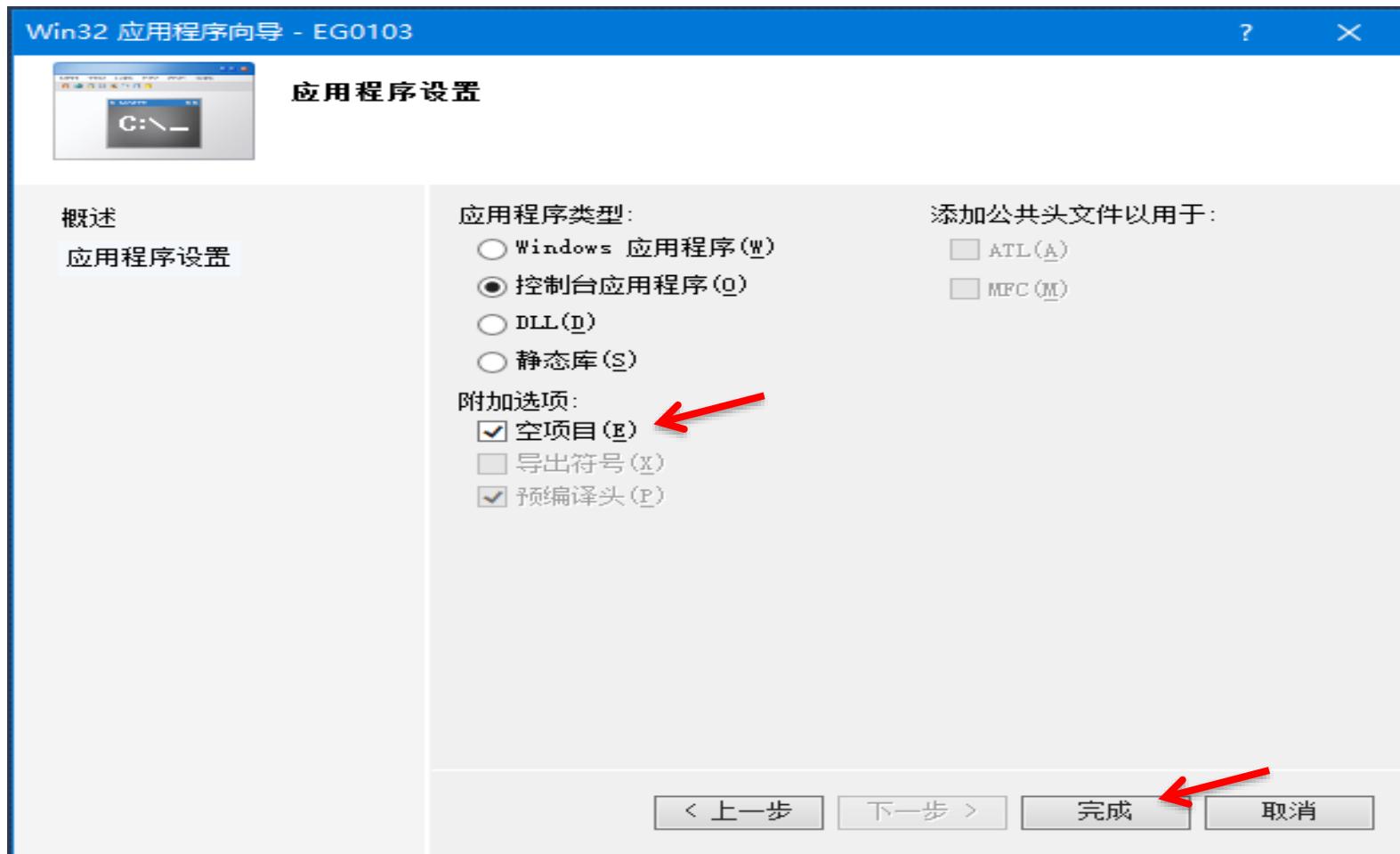
在任一窗口中单击“完成”，接受当前设置。

创建项目后，请参阅该项目的 `readme.txt` 文件，了解有关项目功能和所生成的文件的信息。

< 上一步 **下一步 >** 完成 取消

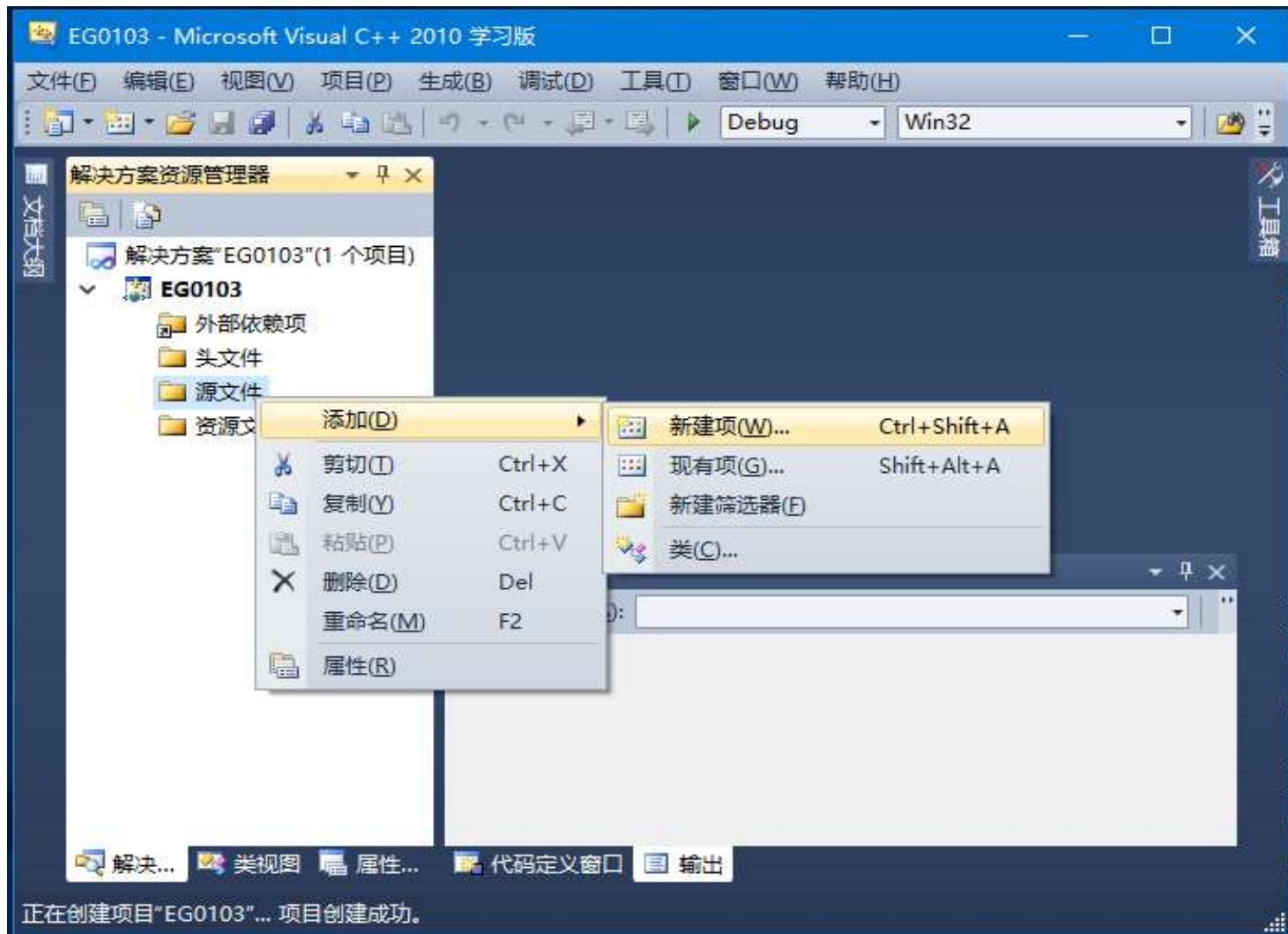


选择“空项目”



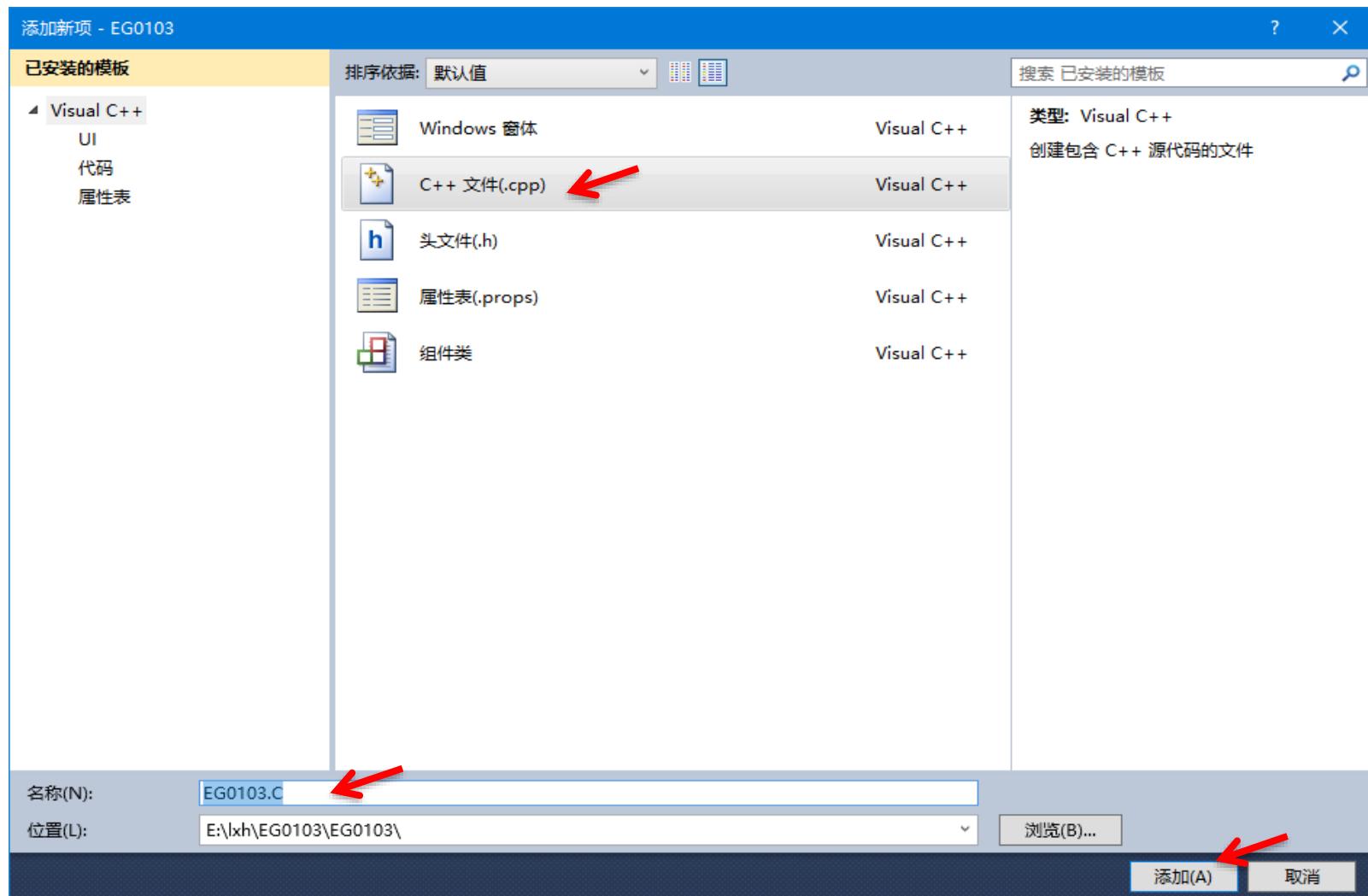


4 新建C源程序：源文件→添加





新建eg0103.c源程序文件





- 5 输入、编辑源程序：Ctrl+S 保存
- 6 编译、链接：调试（生成）菜单-生成解决方案（F7）
- 7 运行：调试菜单-启动调试，或者直接按Ctrl + F5
- 8 出现语法或逻辑错误后，返回5修改源程序。
- 9 设置断点后可以单步执行程序，同时观察各变量的值如何变化。
- 10 如果不退出VC接着开发下一个C 程序，请先关闭当前项目（解决方案）。



基本设置

EG0103 - Microsoft Visual C++ 2010 学习版

文件(F) 编辑(E) 视图(V) 项目(P) 调试(D) 工具(T) 窗口(W) 帮助(H)

启动调试(S) F5 n32 进制

生成解决方案(B) F7

逐语句(I) F11

通过(O) F10

切换断点(G) F9

窗口(W)

清除所有数据提示(A)

导出数据提示(X) ...

导入数据提示(P) ...

选项和设置(G) ...

main(void)

```
/* first and second, see which is the bigger. */
// 包含标准I/O库函数的说明
// 主函数说明

int main()
{
    int first, second;
    // 将first作为初值置为max
    // 比较first和second
    // 将较大的存入first
    if( first < second )
        max=second;

    printf("第一个数是 %d\n", first );
    printf("第二个数是 %d\n", second );
    printf("大数是 %d\n", max);

    return 0;
}
```

100 %

断点

新建... X 新点 条件 即时窗口

名称 条件 条件命中次数

EG0103.C 行 11 (无条件) 总是中断

输出

EG0103 - Microsoft Visual C++ 2010 学习版

文件(F) 编辑(E) 视图(V) 项目(P) 调试(D) 工具(T) 窗口(W) 帮助(H)

启动调试(S) F5

开始执行不调试(H) Ctrl+F5

附加到进程(E) Ctrl+Alt+E

逐语句(I) F11

通过(O) F10

切换断点(G) F9

清除所有断点(D) Ctrl+Shift+F9

启用所有断点(N)

逐行数据提示(A)

导出数据提示(X) ...

导入数据提示(P) ...

选项和设置(G) ...

main(void)

```
/* see which is the bigger. */
// 包含标准I/O库函数的说明
// 主函数说明

int main()
{
    // 提示输入第一个数
    // 从键盘输入数据存入变量
    // 将first作为初值置为max
    // 比较max和second
    // 将较大的second值赋给max
    printf("第一个数是 %d\n", max );
}

100 %

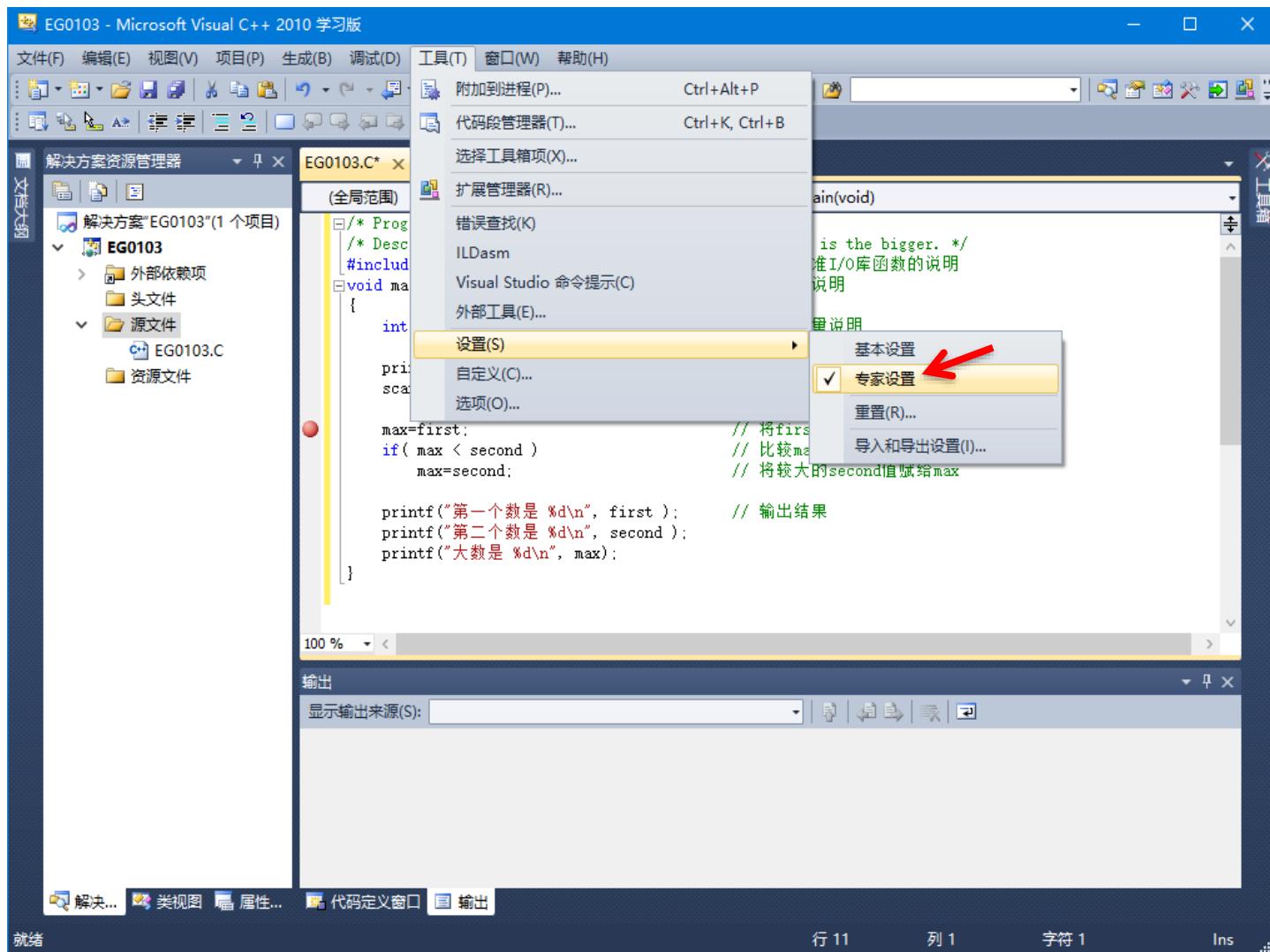
输出
```

显示输出源代码

专家设置



开启专家设置





设置断点调试

The screenshot shows the Microsoft Visual Studio 2010 interface with the following details:

- Title Bar:** EG0103 - Microsoft Visual C++ 2010 学习版(管理员)
- Toolbar:** Includes standard icons for file operations, toolbars, and windows.
- MenuBar:** File(E) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 工具(I) 窗口(W) 帮助(H)
- Toolbox:** Standard development tools.
- Solution Explorer:** Shows the solution "EG0103" with one project "EG0103" containing files like "EG0103.C".
- Code Editor:** Displays the "EG0103.C" file with the following code:

```
/* Program: EG0103.c */
/* Description: Comparing first and second, see which is the bigger. */
#include <stdio.h>
void main( void )
{
    int first, second, max;
    printf("请输入两个整数:");
    scanf("%d,%d", &first, &second);
    max=first;
    if( max < second )
        max=second;

    printf("第一个数是 %d\n", first );
    printf("第二个数是 %d\n", second );
    printf("大数是 %d\n", max);
}
```

A red circle labeled ① points to the line `max=first;`. A red arrow labeled ② points to the "Debug" button in the toolbar.
- Output Window:** Shows build logs:

```
1> EG0103.vcxproj -> E:\lxh\EG0103\Debug\EG0103.exe
===== 全部重新生成: 成功 1 个, 失败 0 个, 跳过 0 个 =====
```
- Status Bar:** 行 7 列 1 字符 1 Ins



开启自动、内存窗口观察变量变化

The screenshot shows the Microsoft Visual Studio 2010 interface during debugging. The 'Break' menu is open, and the 'Memory' option is highlighted with a red arrow. A tooltip below the menu shows the text: '较大的second值赋给max' (Assign the larger second value to max). To the right of the menu, the 'Memory' window is visible, displaying memory dump information for variable &first at address 0x006FFC74.

EG0103 (正在调试) - Microsoft Visual C++ 2010 学习版

文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 工具(T) 窗口(W) 帮助(H)

窗口(W)

- 断点(B) Alt+F9
- 输出(O)
- 监视(W)
- 自动窗口(A) Ctrl+Alt+V, A
- 局部变量(L) Alt+4
- 即时(I) Ctrl+Alt+I
- 调用堆栈(C) Alt+7
- 线程(H) Ctrl+Alt+H
- 模块(O) Ctrl+Alt+U
- 进程(P) Ctrl+Shift+Alt+P
- 内存(M)
- 反汇编(D) Alt+8
- 寄存器(G) Alt+5

类视图 EG0103.C X

(全局范围)

```
/* Program Description */#include<stdio.h>void main(){    int first, second;    printf("请输入第一个数:");    scanf("%d", &first);    printf("请输入第二个数:");    scanf("%d", &second);    if (first > second)        max = first;    else        max = second;    printf("较大的数是: %d", max);}
```

自动窗口

| 名称 | 类型 |
|---------|-------|
| &first | int * |
| &second | int * |
| first | int |
| max | int |

内存 1

| 地址 | 数据 | 类型 |
|------------|-------------|-----|
| 0x006FFC74 | 14 00 00 00 | ... |

解决方案资源管理器 窗口



■ 思考：

- 程序运行时出现闪退怎么办？
- 如何观察程序运行时变量的变化情况？
- 如果编译时提示**scanf.....**文件损坏怎么办？



文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 工具(T) 窗口(W) 帮助(H)

Debug Win32

解决方案资源管理器 EG0103.C.x

(全局范围) main(void)

```
/* Program: EG0103.c */
/* Description: Comparing first and second, see which is the bigger. */

#include <stdio.h>
int main(void)
{
    int first, second, max;
    printf("请输入两个整数: ");
    scanf("%d", &first, &second);
    if (first > second)
        max = first;
    else
        max = second;
    printf("第一个数是 %d\n", first);
    printf("第二个数是 %d\n", second);
    printf("最大数是 %d\n", max);
}
```

文档大纲

解决方案“EG0103”(1个项目)

EG0103

- 生成(U)
- 重新生成(E)
- 清理(N)
- 仅用于项目(J)
- 生成自定义(B)...
- 添加(D)
- 引用(F)...
- 设为启动项目(J)
- 调试(G)
- 剪切(T) Ctrl+X
- 粘贴(P) Ctrl+V
- 移除(V) Del
- 重命名(M) F2
- 卸载项目(L)
- 重新扫描解决方案(S)
- 属性(R)

请右击当前解决方案，选择“属性”



配置(C): 活动(Debug)

平台(P): 活动(Win32)

配置管理器(O)...

| |
|-------------------------------------------|
| > 通用属性 |
| 配置属性 |
| 常规 |
| 调试 |
| VC++ 目录 |
| > C/C++ |
| > 链接器 |
| ▼ 清单工具 |
| 常规 |
| <input checked="" type="checkbox"/> 输入和输出 |
| 独立 COM |
| 高级 |
| 命令行 |
| > XML 文档生成器 |
| > 浏览信息 |
| > 生成事件 |
| > 自定义生成步骤 |

| | |
|------------------------|----------------------------------------------------------|
| 附加清单文件 | |
| 输入资源清单 | |
| 嵌入清单 | 否 |
| 输出清单文件 | \$(IntDir)\$(TargetName)\$(TargetExt).embed.manifest |
| 清单资源文件 | \$(IntDir)\$(TargetName)\$(TargetExt).embed.manifest.res |
| 生成目录文件 | 否 |
| 从 ManagedAssembly 生成清单 | |
| 取消依赖元素 | |
| 生成类别标记 | |
| 启用 DPI 识别功能 | 否 |

➤选择“属性”中“输入输出”项的“嵌入清单”为“否”。

嵌入清单

指定是将清单文件嵌入程序集中还是将其创建为独立的文件。

确定

取消

应用(A)



小结

本章首先简要介绍了什么是计算机程序、如何设计程序以及C语言的发展史及其特点，随后叙述了什么是算法及如何利用自然语言和N-S流程图描述算法。学习本章，应重点了解C程序开发的基本知识，其中包括C程序的标准结构，C语言的字符集和标识符，如何在Visual C++平台上编辑、编译、链接、运行和调试C源程序等等。

通过本章的学习，对如何利用C语言设计程序解决实际问题建立初步的认识。



习题练习

【例 1-1】面向过程的程序设计语言是（ ）。

- A) 机器语言 B) 汇编语言
- C) 高级语言 D) 第四代语言

答案：C。

【例 1-2】以下不是 C语言的特点的是（ ）。

- A) C 语言简洁、紧凑
- B) 能够编制出功能复杂的程序
- C) C 语言可以直接对硬件进行操作
- D) C 语言移植性好

答案：C。



1. 面向过程的程序设计语言是（ ）。

- A 机器语言
- B 汇编语言
- C 高级语言
- D 第四代语言

提交

第1章 程序设计概述





小测试

以下正确的 C 语言标识符是（ ）。

A

% x

B

a+b

C

a123

D

test!

提交



【例 1-3】 算法的特点是（ ）。

- A) 有穷性、确定性、稳定性
- B) 可执行性、移植性、扩充性
- C) 有零个或多个输入、有输出、可操作性
- D) 易读性、稳定性、安全性

答案：A。

【例 1-4】 C 语言源程序的基本单位是（ ）。

- A) 过程 B) 函数
- C) 子程序 D) 标识符

答案：B。

【例 1-5】 一个 C 程序的执行是从（ ）。

- A) `main()` 函数开始，直到 `main()` 函数结束
- B) 第一个函数开始，直到最后一个函数结束
- C) 第一个语句开始，直到最后一个语句结束
- D) `main()` 函数开始，直到最后一个函数结束

答案：A。



【例 1-6】 下面叙述中不正确的是（ ）。

- A) 注释说明只能位于一条语句的后面
- B) 注释说明被计算机编译系统忽略
- C) 注释说明必须括在 “/*.....*/”之间或 “//”的后面
- D) 注释符 “/”和 “*” 之间不能有空格

答案：A。

【例 1-8】 以下正确的 C 语言标识符是（ ）。

- A) %x B) a+b C) a123 D) test!

答案：C。



课后学习视频



慕课视频片段

视频名称：例1.1案例实验与分析



温馨提示：此视频框在点击“上传手机课件”时会进行转换，用手机进行观看时则会变为可点击的视频。此视频框可被拖动移位和修改大小。