

高级语言程序设计

主讲：王老师



尚德机构

学习是一种信仰

第六章 函数



第一節 函数的概念和模块化程序设计

C语言函数的特点和分类：

特点：

- 1、一个C程序由一个或多个函数组成，其中必须有且只能有一个main函数（称为主函数）。
- 2、C程序的执行从主函数开始，如果在主函数中调用其他函数，调用后返回到主函数，并在主函数中结束整个程序的运行。
- 3、主函数可以调用其他函数，但其他函数不能调用主函数。主函数由操作系统调用，其他函数之间可以互相调用。
- 4、函数之间，没有从属关系，互相独立，不能嵌套定义。



▶▶▶ C语言函数的特点和分类

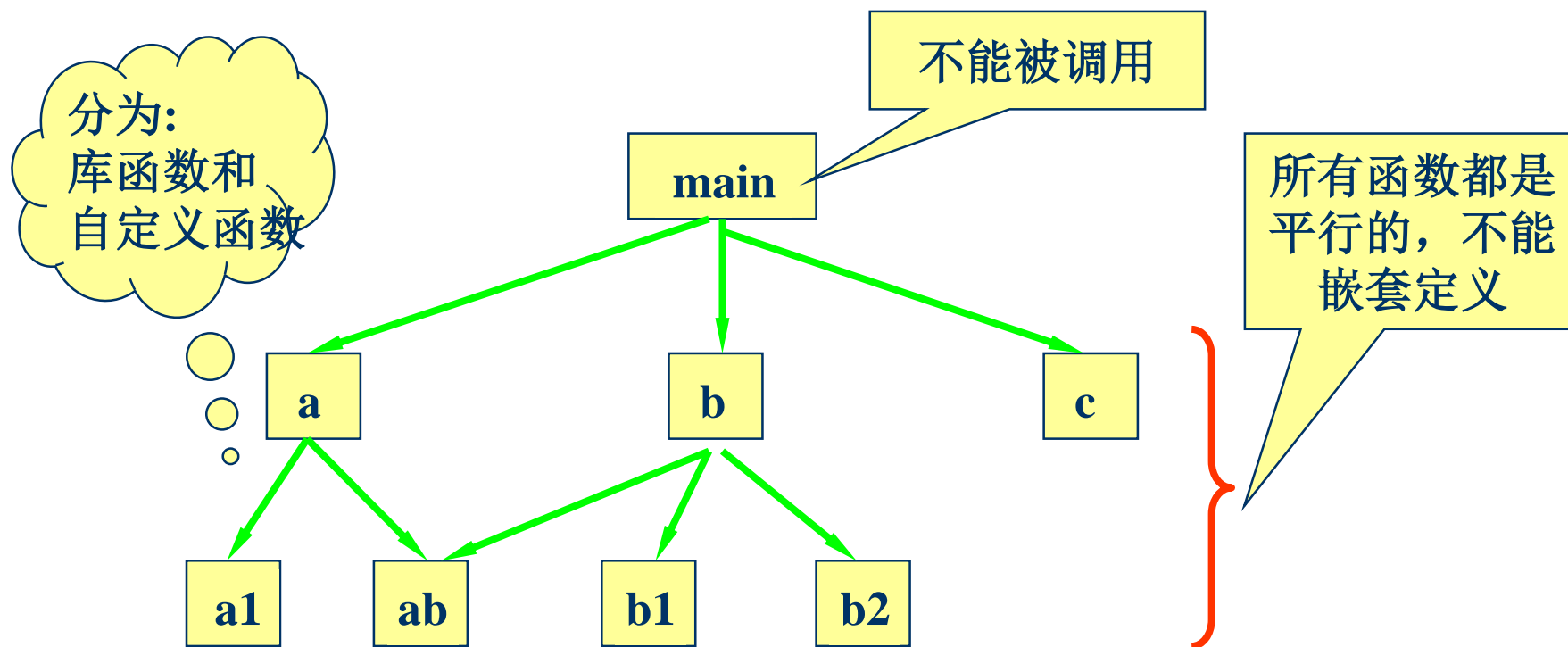
分类：

1. 从函数定义的角度，分为库函数和用户函数。
2. 从函数有无参数的角度，函数分为有参函数和无参函数。
3. 从函数有无返回值的角度，函数分为有返回值函数和无返回值函数。
4. C语言允许将一个源程序分放在不同的程序文件中，采用分块编译、链接生成一个目标程序，其中每个程序文件称为一个“编译单元”。每个编译单元可以包含若干个函数。从函数作用范围的角度，函数分为外部函数和内部函数。外部函数是可以被任何编译单元调用的；内部函数只能在本编译单元被调用。



★函数的概念

一个大的程序一般应分为若干个程序模块，每个模块实现一个特定的功能，这些模块称为子程序，在C语言中子程序用函数实现。



函数的一般格式

任何函数（包括主函数）都是由函数首部和函数体两部分组成。

函数定义的一般形式：

数据类型符 函数名(形式参数表)

{

 数据定义语句部分;

 执行语句部分;

}





函数的一般格式

1、函数首部

“数据类型符”规定本函数返回值的数据类型，可以是各种基本数据类型，也可以是指针类型（只要在函数名前面加一个*）。需要注意的是，如果函数无返回值，数据类型符使用void。

需要注意的是，函数首部末尾不能加分号。

“函数名”是一个标识符，在同一个编译单元中的函数不能重名。

“形式参数表”是使用逗号分隔的若干个形式参数及其数据类型的说明，具体格式：

数据类型符 形式参数1[,数据类型符 形式参数2,...]



函数的一般格式

2、函数体

“数据定义语句部分”由若干个数据定义语句组成，用来定义本函数中使用的变量、数组和指针变量等。

“执行语句部分”由本函数中完成函数功能的程序段组成。如果是有返回值的函数，则其中至少有一条返回语句“return(表达式);”，表达式的值就是本函数的返回值。返回语句中的圆括号可以省略。函数的返回值是通过函数中的返回语句获得的。返回语句的一般格式是return(返回值表达式);。如果是无返回值的函数，则返回语句应为“return;”，也可以省略返回语句。返回语句在函数体的执行语句部分的位置是任意的。返回语句的功能是结束本函数的运行，返回到主调函数的函数调用语句继续执行。



函数的一般格式

【说明】

C语言中，所有函数（包括主函数）都是平行的。一个函数的定义，可以放在程序中的任意位置，主函数之前或之后。但是在一个函数的函数体内不能再定义另一个函数，即不允许函数的嵌套定义。



➡➡➡ 各类函数定义的格式

1、无参函数的定义

如果一个函数在被调用时不需要主调函数向其传递数据，则该函数可以定义为无参函数，其定义的一般格式：

数据类型符或void 函数名(void)

{

 数据定义语句部分;

 执行语句部分;

}

void表示函数没有参数，void也可以省略，而只在函数名后面，跟一对空括号，但这对空括号不能省略。



★无参函数的定义形式

❖类型标识符:

- 用于指定函数带回的值的类型，不写时为int型。
- 不带返回值时可以不写。

例 无参函数

```
printstar( )  
{ printf("*****\n"); }  
或  
printstar(void )  
{ printf("*****\n"); }
```

合法标识符

类型标识符 函数名 ()
{ 说明部分
 语句
}

函数体

有参函数的定义

2、有参函数的定义

如果一个函数在被调用时需要主调函数向其传递数据，则该函数可以定义为有参函数，其定义的一般格式：

数据类型符或void 函数名(形式参数表)

```
{  
    数据定义语句部分;  
    执行语句部分;  
}
```



★有参函数定义的一般形式

函数返回值类型
隐含为int型

现代风格:

函数体

类型标识符 函数名 (形式参数表列)
{
说明部分
语句
}

例 有参函数 (现代风格)

```
int max(int x,int y)
{
    int z;
    z=x>y?x:y;
    return(z);
}
```

例 有参函数 (现代风格)

```
int max(int x, y)
{
    int z;
    z=x>y?x:y;
    return(z);
}
```

✗

★空函数

❖为扩充功能预留，在主调函数中先占一个位置。

类型标识符 函数名 ()
{ }

例 空函数
dummy()
{ }

函数体为空

★对形参的声明的传统方式

❖即把对形参的声明放在函数定义的下一行

类型标识符 函数名 (形参表)
形参类型说明
{
说明部分
语句
}

例 有参函数 (传统风格)
int max(x,y)
int x,y;
{
int z;
z=x>y?x:y;
return(z);
}

无返回值函数的定义

3、无返回值函数的定义

void 函数名(形式参数表或void)

{

 数据定义语句部分;

 执行语句部分;

}



有返回值函数的定义

4、有返回值函数的定义

数据类型符 函数名(形式参数表或void)

```
{  
    数据定义语句部分;  
    执行语句部分;  
}
```

有返回值函数和无返回值函数的主要区别体现在两个方面。

一是数据类型符，有返回值函数的数据类型符可以选取任意的数据类型符，而无返回值函数的数据类型符只能是void；二是有返回值函数的函数体中必须有语句return(表达式)；而无返回值函数的函数体中的语句return；可以有也可以省略。



返回值类型

函数返回值的类型即函数类型，是指返回给主调函数的结果的类型，应根据具体函数的功能确定。如果函数不返回任何值，则函数返回值类型定义为 `void`，称为“无类型”。



函数调用的一般格式

函数调用的一般形式：函数名([实际参数表]);

- 1、实参的个数、类型和顺序，应该与被调函数所要求的形参个数、类型和顺序一致，才能正确地进行函数之间的数据传递。
- 2、当有多个实参时，实参之间用逗号分隔。
- 3、无参函数没有参数，则调用时函数名后面跟一对空的圆括号，但这对圆括号必须保留。





函数的形式参数和实际参数

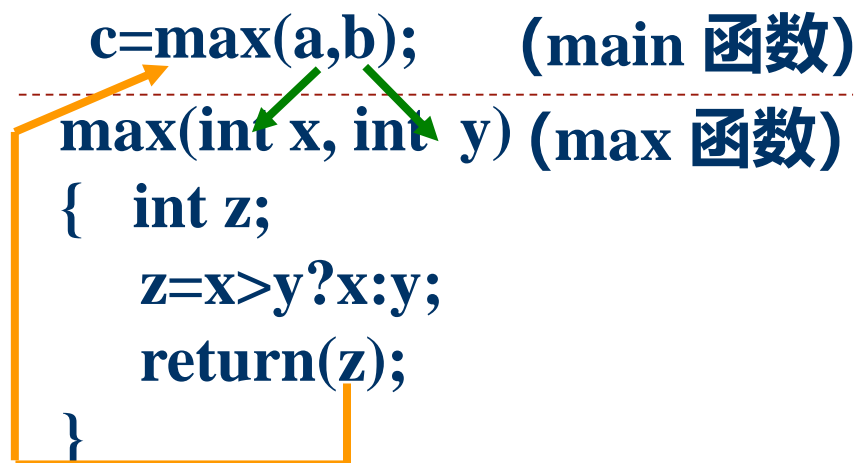
形参是在函数定义时设置的，用来接收从主调函数传来的对应的实参数据。实参是调用函数时的实际参数，实参可以是常量、变量或表达式，也可以是函数的返回值，无论哪种形式必须有确定的值。**实参在数量、类型和顺序上与形参必须一一对应和匹配。**



★形式参数和实际参数

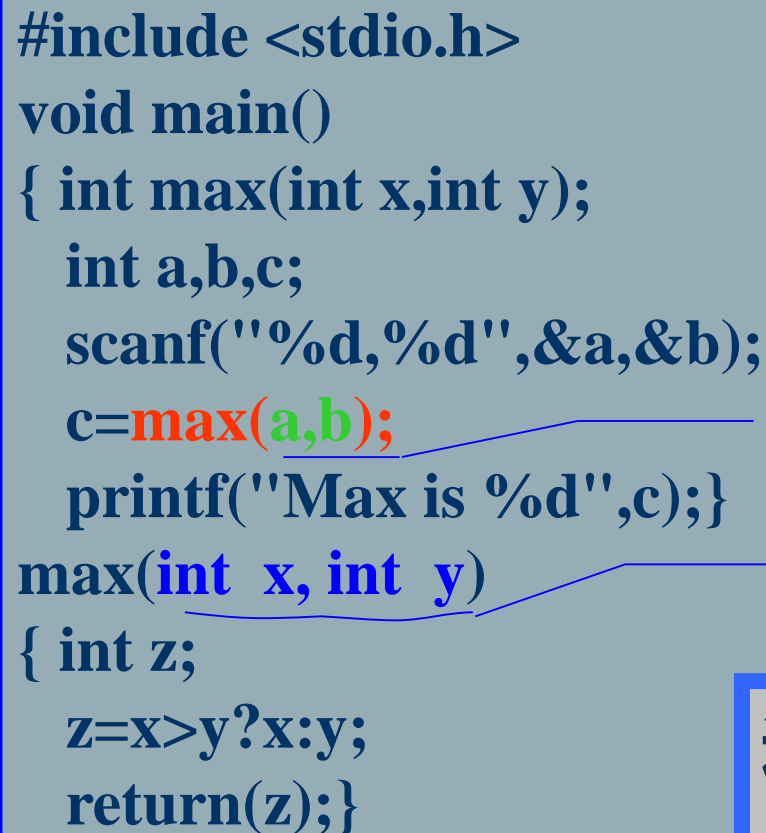
- ❖形式参数：定义函数时函数名后面括号中的变量名
- ❖实际参数：调用函数时函数名后面括号中的表达式

```
c=max(a,b);    (main 函数)
max(int x, int y) (max 函数)
{ int z;
  z=x>y?x:y;
  return(z);
}
```



例 比较两个数并输出大者

```
#include <stdio.h>
void main()
{ int max(int x,int y);
  int a,b,c;
  scanf("%d,%d",&a,&b);
  c=max(a,b);
  printf("Max is %d",c);}
max(int x, int y)
{ int z;
  z=x>y?x:y;
  return(z);}
```



实参

形参

运行：7 8
Max is 8

❖ 几点说明:

- 实参可以是常量、变量或表达式。必须有确定的值。当函数调用时，将实参的值传递给形参，若是数组名，则传送的是数组首地址。
- 形参必须指定类型，只能是简单变量或数组，不能是常量或表达式
- 形参与实参类型一致，个数相同顺序相同。
- 若形参与实参类型不一致，自动按形参类型转换——函数调用转换
- 形参在函数被调用前不占内存;函数调用时为形参分配内存；调用结束，内存释放
- 实参对形参的数据传送是值传送，也是单向传送，当被调函数的形参发生变化时，并不改变主调函数实参的值。形、实参占据的是不同的存储单元

例：形、实参占据的是不同的存储单元

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

```
void main( )
```

```
{
```

```
    int a=2,b=3;
```

```
    printf ("a=%d, b=%d\ n",a, b);
```

```
    printf("&a=%0x,&b=%0x\n" ,&a,&b);
```

```
    add(a,b);
```

```
    printf("a=%d,b=%d\n", a,b);
```

```
    printf("&a=%0x,&b=%0x\n", &a,&b);
```

```
}
```

```
add(int x,int y)
```

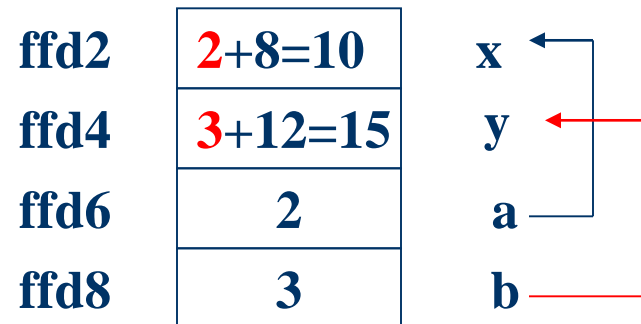
```
{
```

```
    x=x+8; y=y+12;
```

```
    printf("x=%d,y=%d\ n",x,y);
```

```
    printf("&x=%0x,&y=%0x\n",&x,&y);
```

```
}
```



运行结果：

a=2,b=3

&a=ffd6,&b=ffd8

x=10,y=15

&x=ffd2,&y=ffd4

a=2, b=3

&a=ffd6,&b=ffd8



函数的返回

返回语句有以下两种使用方式：

- 1、无返回值的函数 `return;`
- 2、有返回值的函数 `return(返回值表达式);`或`return 返回值表达式;`

函数的返回值是通过被调函数中的返回语句获得的。若函数没有返回值，则应将函数返回值的类型定义为`void`，在函数体中可以没有返回语句，也可以使用无返回值的`return`语句，但是不能使用有返回值的`return`语句；若函数有返回值，则应将函数定义为非`void`类型，在函数体中只能使用有返回值的`return`语句，而且必须有该语句。函数的返回值只能有一个，但可以有多条`return`语句，一旦执行到其中一个`return`语句，则立即返回主调函数，被调函数中的其他语句不再执行。

`return`语句中表达式的数据类型应与函数首部定义的函数返回值的类型一致。



★函数的返回值

❖返回语句形式:

`return(表达式);` 或 `return 表达式;`

❖功能: 使程序控制从被调用函数返回到调用函数中, 同时把返回值带给调用函数

❖说明:

- 函数的返回值，必须用 `return` 语句带回。
- `return` 语句只能把一个返回值传递给调用函数。
- 函数中可有多条`return`语句，执行哪一个由程序执行情况来定。

`if(a>b) return(a);`

`else return(b);`

- `return` 后的值可以是一个表达式，如: `return(x > y ? x : y);`
- 返回值的类型为定义的函数类型，不指定的按整型处理。

如: `int max(int x, int y)`

`float min(float a, float b)`

`double abc(float d1, float d2)`

例：无返回值函数

```
void swap(int x,int y )
{
    int temp;
    temp=x;
    x=y;
    y=temp;
}
```

例：函数返回值类型转换

```
输入： 1.5, 2.5
输出： Max is 2
```

```
#include <stdio.h>
void main()
{
    int max(float x,float y);
    float a,b;
    int c;
    scanf("%f,%f",&a,&b);
    c=max(a,b);
    printf("Max is %d\n",c);
}

max(float x, float y)
{
    float z;
    z=x>y?x:y;
    return(z);
}
```

课堂练习

1、在C语言中，当定义一个函数的类型为void时，说明执行该函数后_____。





课堂练习

1、在C语言中，当定义一个函数的类型为void时，说明执行该函数后_____。

答案：无返回值

解析：如果函数不返回任何值，则函数返回值类型定义为void，称为“无类型”。



课堂练习

2、在一个文件的不同函数中不能重名的是（ ）

- A. 函数名
- B. 形式参数
- C. auto型内部变量
- D. register型内部变量





课堂练习

2、在一个文件的不同函数中不能重名的是（ ）

- A. 函数名
- B. 形式参数
- C. auto型内部变量
- D. register型内部变量

答案：A

解析：“函数名”是一个标识符，在同一个编译单元中的函数不能重名。



第二節 函数声明

函数声明又称为函数原型，其一般格式：

[存储类型][数据类型符] 函数名(形参类型[形参名1][,形参类型[形参名2],...]);

若被调函数为：

- 1、库函数，除了少数库函数（如scanf()、printf()）外，都要求在本文件开头用文件包含命令#include<头文件名.h>或#include “头文件名.h” 包含被调函数的信息。
- 2、用户函数，若主调函数与被调函数不在同一个编译单元，则在定义函数的编译单元中必须将该函数定义为外部函数，同时主调函数的函数体或主调函数所在编译单元的开头将被调函数按照如下格式进行声明：



函数声明

`extern` 数据类型符 函数名(形参表);

需要注意的是，函数声明是一条语句，末尾加分号；而函数首部不是语句，末尾不加分号。





函数声明

说明：

- 1、函数声明告知编译程序函数返回的数据类型、函数所要接收的参数个数、类型和顺序，编译程序利用函数声明校验函数调用是否正确。
- 2、函数声明中可以只说明形参类型和形参个数，而无须说明形参名。
- 3、函数声明可以在主调函数中，也可以在所有函数的外部。
- 4、函数声明、函数定义、函数调用要保持一致。
- 5、如果程序中没有包含函数声明，则编译程序会用第一次出现时的该函数来构造函数声明。



例 对被调用的函数作声明

```
#include <stdio.h>
void main()
{
    float add(float x,float y ); /*对被调用函数的声明*/
    float a,b,c;
    scanf("%f,%f",&a,&b);
    c=add(a,b);
    printf("sum is %f",c);
}
float add(float x, float y) /*函数首部*/
{
    float z;                /*函数体
    z=x+y;
    return(z);
}
```

float add(float,float);

输入： 3.6 , 6.5

输出： sum is 10.100000

第三節 函数的参数和数据传递方式

C语言规定在函数之间传递数据包括**值传递、地址传递、返回值和全局变量传递**四种方式。

值传递方式：

值传递方式是在形参和实参之间传递数据的一种方式，值传递方式传递的是参数值。判断是否是值传递方式的唯一方法是看函数定义时的形参是不是变量形式。如果形参是变量，则是值传递方式。

值传递方式能够确保不管在被调函数中怎样操作或改变形参的内容，但主调函数中的实参并不会发生变化。实参对形参的数据传递是单向的，即只能从主调函数将实参值传递给被调函数的形参，而不能把被调函数的形参值反向传递给主调函数的实参。



数组元素作为函数实参

数组元素只能用做函数实参，和变量作为函数实参一样。在调用函数时，把数组元素的值传递给形参，实现单向的值传递方式。

需要注意的是：

- 1、数组元素作为实参时，只要数组的类型和函数形参的类型一致即可，并不要求函数的形参也是下标变量。
- 2、普通变量或下标变量作为函数参数时，形参变量和实参变量由编译程序分配不同的内存单元。





数组名作为函数参数

数组名作为函数参数时，既可以作为形参，也可以作为实参。数组名作为函数参数时，要求形参和相应的实参必须是同类型的数组，都必须有明确的数组定义。

如果形参是数组名，则传递方式称为“地址传递方式”。

数组名作为形参时，实参向形参传递的是实参数组的首地址，实参数组和形参数组的各元素按照存储结构一一对应共享存储空间。



利用返回值的数据传递方式

利用返回值的数据传递方式并不是在形参和实参之间进行数据传递，而是通过函数调用直接返回一个值到主调函数。

利用返回值传递数据，在定义函数时需要注意以下两点：

- 1、函数首部中需要有“数据类型符”，说明该该函数返回值的数据类型。
- 2、函数体中需要有语句“return(表达式);”，其中的表达式即是函数的返回值。



课堂练习

3、在函数间传递数据时，函数的形参和调用时的实参都是数组，传递方式为_____。





课堂练习

3、在函数间传递数据时，函数的形参和调用时的实参都是数组，传递方式为_____。

答案：地址传递

解析：数组名作为函数参数时，既可以作为形参，也可以作为实参。如果形参是数组名，则传递方式称为“地址传递方式”。



课堂练习

4、函数调用时，若形参是一个变量，而对应的实参是一个有确定值的表达式，则函数参数的传递方式是_____传递。





课堂练习

4、函数调用时，若形参是一个变量，而对应的实参是一个有确定值的表达式，则函数参数的传递方式是_____传递。

答案：值

解析：值传递方式是形参和实参之间传递数据的一种方式，值传递方式传递的是参数值。判断是否是值传递方式的唯一方法是看函数定义时的形参是不是变量形式，如果形参是变量，则是值传递方式。





祝大家顺利通过考试!

