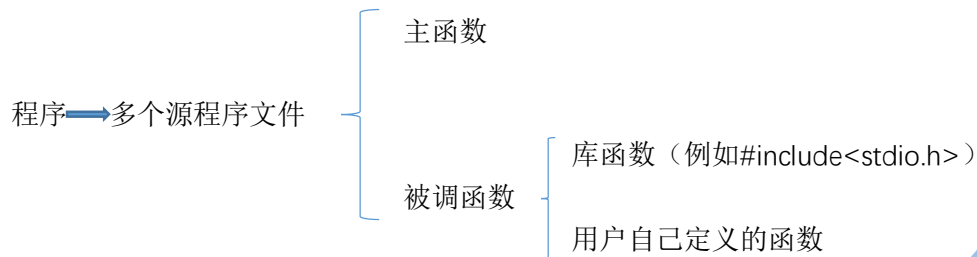


第七章、函数

一、程序的结构



函数特点:

- 1、库函数类型包括 I/O, 分类, 转换, 诊断, 目标控制, 图形, 接口, 进程控制, 处理, 数学, 文本窗口, 内存, 时间, 日期等。
- 2、C 程序从 main 函数开始, 调用其他函数后流程回到 main 函数, 在 main 函数中结束整个函数的运行。main 函数由系统定义。
- 3、所有函数都是平行的, 在定义函数时是互相独立的, 一个函数并不从属于另一函数, 函数不能嵌套定义, 但可以互相调用 (不能调用 main 函数)。

二、函数的结构

例:

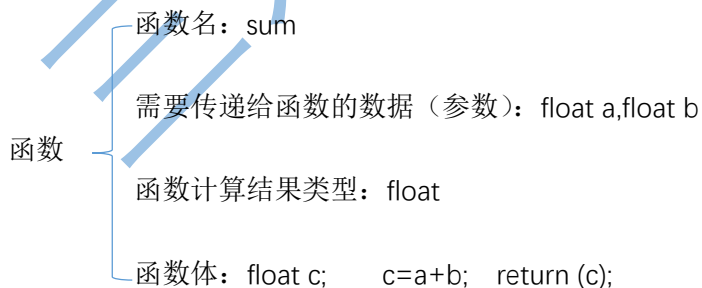
```
float sum(float a,float b)
{
    float sum;
    sum=a+b;
    return sum;
}
```

无分号

或

```
float sum(a,b)
float a,b;
{
    float sum;
    sum=a+b;
    return (sum);
}
```

无分号



说明:

- 1、调用函数时, 变量名不能与函数名一致。 例: max=max(a,b); ❌
- 2、函数名也不能是 main。
- 3、若无参数, 写 void 或者省略。
- 4、函数返回值的类型缺省类型为整型。

5、无参函数一般不需要带回函数值，因此可以不写类型标识符。

6、可以有空函数。

例：dummy(){};

三、函数的参数

例：

```
#include<stdio.h>
float sum(float a,float b)
{   float sum;
    sum=a+b;
    return (sum);    }
void main()
{   float c,x=2.,y=3.;
    c=sum(x,y);
    printf("sum=%f",c);    }
```

形式参数(形参):在定义函数时后面括弧中的变量名。

例：a b

实际参数(实参):在调用函数时函数名后面的表达式。

例：x y

说明：

- 1、形参变量在未出现函数调用时并不占内存单元，只有在发生函数调用时形参才被分配内存单元。
- 2、实参可以是常量、变量或表达式，要求有确定的值。
- 3、在被定义的函数中，必须指定形参的类型。
- 4、实参与形参的类型应该一致。
- 5、C 语言规定，实参对形参的数据传递是“值传递”（单向传递），只由实参传给形参，而不能由形参传递给实参。因此，在执行一个被调用函数时，形参的值如果发生改变，并不会改变实参的值。
- 6、在内存中，实参单元与形参单元是不同的单元。调用函数时，才给形参分配单元，并将实参的值传给形参，调用结束后，形参单元释放。
- 7、各个函数（包括主函数）函数名可以一样。

四、return 语句

1、一个函数中有多个 return 的情况：

```
例：int max(int x, int y)
{
    if (x>y) return(x);
    else return(y);
}
```

- 2、若不需要带回函数值，将函数定义为 void 型，否则需要有返回语句。
- 3、若无 return 语句，遇}时，自动返回调用函数
- 4、return 语句中表达式类型与函数返回值类型不一致时转换数据类型

五、函数的调用

调用方式	函数语句	例：printstar();
	函数表达式	例：c=2*max(a,b);
	函数参数	例：m=max(a,max(b,c));

说明：

- 1、调用的函数必须存在。
- 2、调用库函数时前面加上#include。例如：#include<stdio.h>
- 3、如果用自己定义的函数，且该函数与调用它的函数（主调函数）在同一个文件中，一般应该在主调函数中对被调用函数的返回值的类型作说明。
可以不作说明的几种情况：
 - ①如果函数的返回值是整型或字符型，可以不必进行说明，系统对它们自动按整型说明。
 - ②如果被调用函数的定义出现在主调函数前，可以不必加以说明。
 - ③如果已在所有函数定义之前，在文件的开头，在函数的外部已说明了函数类型，则在各个主调函数中不必对所调用的函数再作类型说明。

六、数组作为函数参数

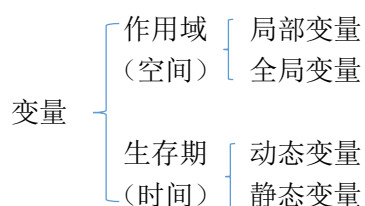
1、参数传递

- ①数组元素作实参，与单个变量一样，这是因为数组元素可以作为表达式的组成部分；
- ②数组名作函数参数，形、实参数都应是数组名，类型要一样，传送的是数组首地址
- ③实参数组与形参数组大小可以不一样，但实参>形参；

2、数组为函数参数

- ①数组元素做函数实参时，单向传递。
例：y=large(a[0],b[0]);
- ②可以用数组名作函数参数，此时实参与形参都应用数组名。
例：y=large(a[10]);
 - i.数组名作函数参数，应该在主调函数和被调用函数分别定义数组（形参数组、实参数组）。
 - ii.实参数组和形参数组类型应该一致。
 - iii.实参数组和形参数组大小可以不一致，如果要求形参数组得到实参数组全部的元素值，则应当指定形参数组和实参数组大小一致。
 - iv.数组名作函数参数时，不是单纯“值传递”，而是把实参数组的起始地址传递给形参数组，这样两个数组就共同占一段内存单元。
 - v. 形参数组也可以不指定大小。

七、变量



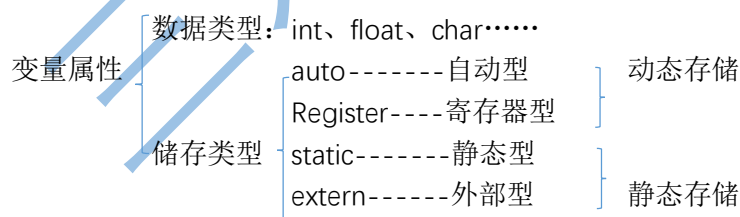
局部变量说明:

- 1、主函数 `main()` 中定义的变量也是局部变量, 它只能在主函数中使用, 其它函数不能使用。同时, 主函数中也不能使用其它函数中定义的局部变量。
- 2、形参变量属于被调用函数的局部变量; 实参变量则属于全局变量或调用函数的局部变量。
- 3、**允许在不同的函数中使用相同的变量名**, 它们代表不同的对象, 分配不同的单元, 互不干扰, 也不会发生混淆。
- 4、在复合语句中定义的变量也是局部变量, 其作用域只在复合语句范围内。其生存期是从复合语句被执行的时刻到复合语句执行完毕的时刻。

全局变量说明:

- 1、**应尽量少使用全局变量。**
 - ①全局变量在程序全部执行过程中始终占用存储单元;
 - ②降低了函数的独立性、通用性、可靠性及可移植性
 - ③降低降低程序清晰性, 容易出错
- 2、**如果在同一个源文件中, 外部变量与局部变量同名, 在局部变量的作用范围内, 外部变量不起作用。**
- 3、全局变量定义必须在所有的函数之外, 且只能定义一次, 并可赋初始值。全局变量定义的形式为:

```
extern float a=2;          (extern 可省略)
```



定义变量完整形式:

存储类型说明符 数据类型说明符 变量名 1, 变量名 2, ..., 变量名 n;

例: `auto int a,b,c,d,e;`

说明:

- 1、`auto` 和 `register` 变量若被初始化, 则每当进入该程序块后, 都执行该变量的初始化赋值功能。

-
- 2、没有进行初始化的 auto 和 register 变量，其初值不定，不能直接在程序中使用。
 - 3、static 和全局变量若不初始化，自动赋 0。
 - 4、static 和全局变量初始化时必须用常量为其赋初值。
 - 5、auto 和 register 变量初始化时可以用常量或变量赋初值。
 - 6、static 型局部变量的初始化仅执行一次。
 - 7、extern 型的变量不进行初始化。
 - 8、局部变量默认为 auto 型。
 - 9、register 型变量个数受限,且不能为 long, double, float 型。
 - 10、局部 static 变量具有全局寿命和局部可见性。
 - 11、局部 static 变量具有可继承性。
 - 12、extern 不是变量定义,可扩展外部变量作用域。

三、学习生活部