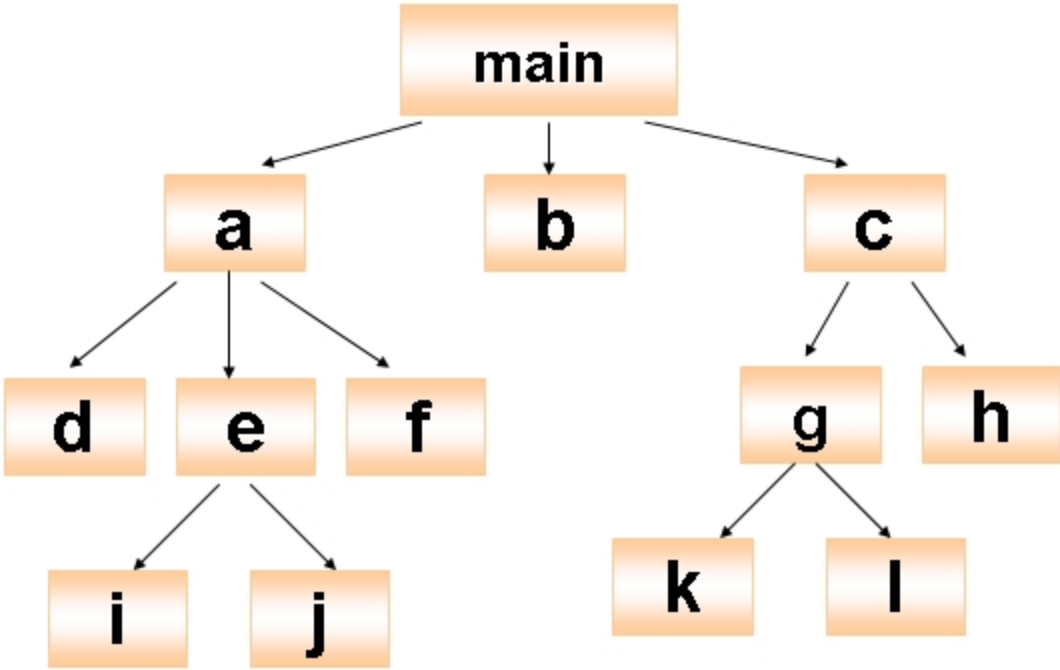
千锋嵌入式学院C语言培训

函数专题

本章主要内容

- C语言程序的结构
- 函数的定义
- 函数的调用
- 函数的嵌套调用
- 递归函数
- 数组与函数
- 全局变量与

- C程序的结构由若干个C函数组成。
- C函数是C程序的组成部分，是由完成一特定任务的说明语句和执行语句组成的基本功能单元。
- C函数的功能相当于其它程序设计语言中的一个程序模块，或一个子程序。


main
a
b
c
d
e
f
g
h
i
j
k
l

函数调用程序实例

```
int main( )
 {
    printstar();
     print_message( );
    printstar( );
    return 0;
 }
 void printstar( )
{ printf("***************\n");}
void print_message( )
{ printf("how do you do!\n");}
```

调用函数

用户自定义函数

函数的分类(1)

- 从使用角度分:

  标准库函数:以程序库的形式直接提供给用户使用。

  stdio.h.....gets()、printf()

  math.h .....sin()、sqrt()

  用户自定义函数: 由用户自己建立定义。

函数的分类(2)

- 从函数形式分:
  - 无参函数: 调用函数时主函数不将数据传送给被调函数。用于完成特定功能的操作。

  如: c=getchar( )……
  - 有参函数: 调用函数时在主调函数和被调函数之间有参数传递。

  如: putchar(c)、puts(str)、……

函数说明

- 一个C程序是由一个或多个源文件组成,每个源文件由若干函数组成。
- C程序的执行从main函数开始，也从main函数终止。
- 源程序中所有的函数在结构上都是平行的，互相独立的。
- main函数可以调用其他函数，其他函数之间可以互相调用。

函数概念和定义

- 函数的作用
  - 函数使我们的程序清晰明白
  - 为开发人员提供解决问题的方法：细化
  - 一次定义，处处使用，利用以有的代码
  - 抽象出公共的部分，隔离开易变部分
- 函数用法
  - 使用之前必须先定义
  - 通过函数调用来使用，类似上下级管理形式
  - 调用时指定函数名字和所需要的信息（参数）
  - 调用完成后向老板报告工作，递交报告（返回值）

函数概念和定义

- 使用已有函数
  - 包含函数所在头文件
  - 在自己的程序里调用库函数
  - 链接上函数所在的库
  - 数学库：math.h　-lm
  - 字符和字符串：stdlib.h string.h
  - 输入输出函数库：stdio.h

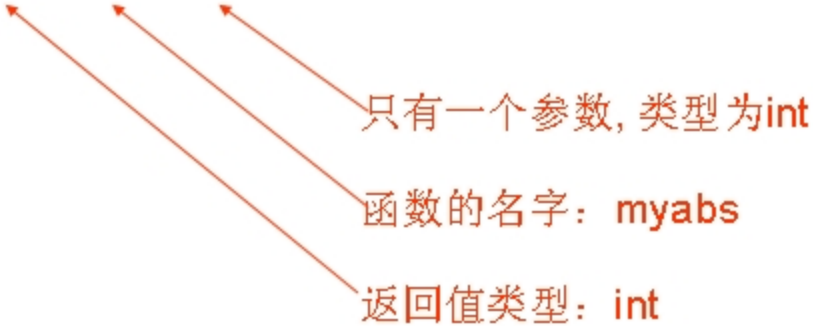
函数概念和定义

- 什么情况下自定义函数
  - 需要一个功能相对独立的子模块
  - 一段代码多次使用
- 如何自定义函数
  - 明确函数功能，参数和返回值类型
  - 声明函数原型，建议放在头文件中
  - 定义函数体内容

函数概念和定义

- 声明函数原型
  - 声明了函数的名字，参数个数和类型，返回值类型
  - 让调用者知道如何调用

```
int myabs(int n);  /*计算整数n的绝对值*/
```



函数概念和定义

- 定义函数体

```
int myabs(int n)  /*计算整数n的绝对值*/
{
      if(n<0)
            n = -n;
      return n; //执行完后把结果给调用者
} /*函数结尾*/
/* 这里是函数外部 */
```

函数概念和定义

- **函数体执行**
  - 从大括号内第一行代码开始
  - 执行到return或是函数结尾结束

```
int myabs(int n)
{ /*函数入口点，从这里开始执行
      if(n<0)
            n = -n;
      return n; /*函数在这里结束*/
    /*n ++  这里的代码不会被执行*/
} /*函数结尾*/
```

参数和返回值

- 函数调用-形参，实形和返回值
  - 调用者和被调用者之前沟通的桥梁

```
int abs(int n);  /* 声明时的n为形式参数,说明被调用者需要的信息*/
int main()
{
        int a = -3;
        int b = abs(a); /*调用时的a为实际参数,调用者提供实际信息*/
        /* b 用于保存返回值: 调用者收到工作报告 */
        /* int c = abs(9); 用实际参数9调用*/
        ……
}
```

- 传值调用：把实际参数复制一份，将复本传入子函数

```
int abs(int n);  /*函数声明时指明参数是数值类型*/
int main()
{
        int a = -3;
        int b = abs(a); /* 把a复制一份，将复本传进去进行操作
                         * a本身并没有被子函数修改
                         * a 是main中的变量，abs没有办法访问到a */
        ……
}
```

局部变量作用域

- **函数内部定义的变量，只有在该函数内部才能通过变量名找到，函数外部不可见**

```
int add(int a, int b)
{
        int sum = 0;        /* 只有在add内部才能使用sum */
        ch = 'z';           /* 错误：add内看不到main的局部变量ch
        … …
}
/* 函数外面看不见 add 内定义的变量sum */
int main
{
        char ch = 'x';      /* ch只能在main内部使用 */
        sum ++; /* 错误：在main里看不到add内定义的局部变量sum */
}
```

局部变量作用域

- **在函数外部看不到局部变量，只能通过地址间接访问**

```
int main()
{
        int a = -3;
        int b = abs(&a);        /* 传址调用，让子函数可以通过地址找到
        ……                       * 父函数中的局部变量，并进行操作*/
}
```

- **返回值：把局部变量复制后得到复本，将复本传回去**

```
int add(int a, int b)
{
        int sum = a + b;
        return sum;             /* 复制sum, 把复本传给调用者
}                               /* 调用者得不到sum本身  */
```

递归函数

- 递归调用：

  在调用一个函数的过程中又出现直接或间接的调用该函数本身，称为函数的递归调用。

注意：

递归结构构成了另外一种形如循环的结构。

递归函数设计举例：求n!

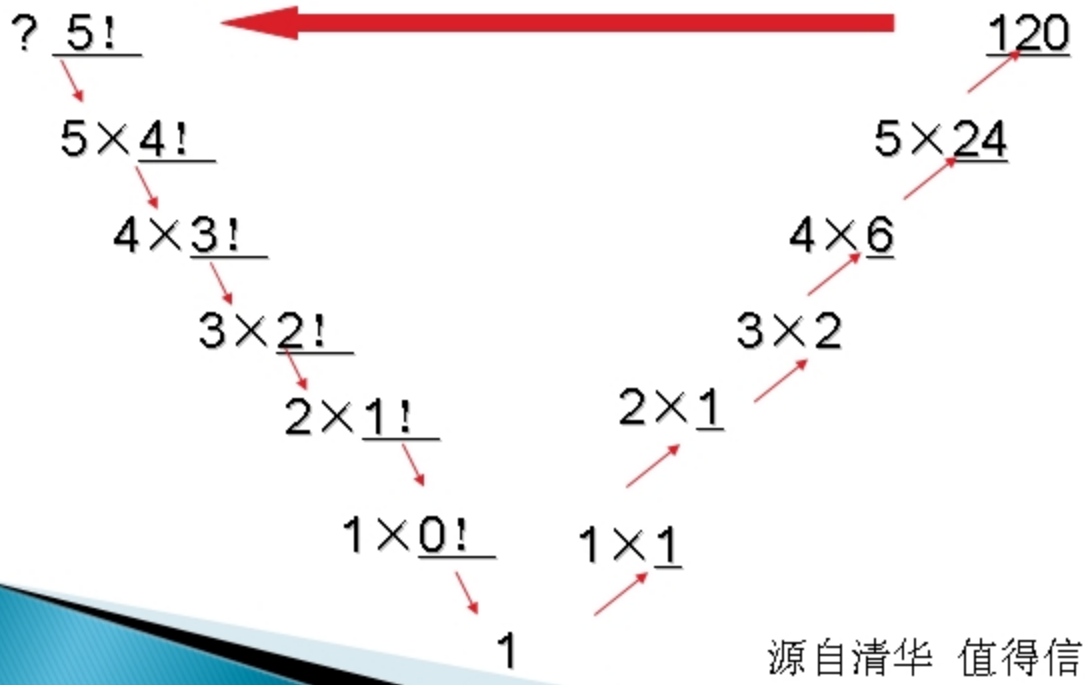
- **算法描述：**

n!=1 × 2 × 3 × ..... × (n−1)×n

- **求积公式：**

$$f(n) = \begin{cases} 1 & n=0 \\ n \times (n-1)! & n>0 \end{cases}$$

5!的运算过程：



源自清华 值得信赖

课堂练习：
有两个数，分别采用非递归和递归的方式来求的最大公约数
设我们有两个数m、n
1、用m除以n，得余数r
2、使m=n ， n=r
3、若r=0，则m就是最大公约数；若r不等于0，返回第1步
我们可以看到，如果m>n，那么没说的；如果m<n，在第一次m除以n后，余数恰好为m，这样一交换，最后还会变成m>n的情况。

分别用非递归和递归的方式打印倒三角形，输入一个底边长n
n = 9
```
*******
 *****
  ***
   *
```

Thank you