

# C语言基本知识

陈 斌

## 目录

- 第一节 概述
- 第二节 基本知识
- 第三节 流程控制
- 第四节 指针与数组
- 第五节 函数

## 函数

### ❖ 有返回值函数

类型 函数名 (类型 虚参1, 类型 虚参2, .....)

```
{  
    return 表达式;  
}
```

### ❖ 无返回值函数

void 函数名 (类型 虚参1, 类型 虚参2, .....)

```
{  
    return;  
}
```

## 函数

```
#include<stdio.h>  
  
float average(float, float);  
  
int main()  
{  
    float a, b;  
    scanf("%f %f", &a, &b);  
    printf("the average of %5.1f and %5.1f is %5.1f\n",  
        a, b, average(a, b));  
  
    return 0;  
}  
  
float average(float x, float y)  
{  
    return (x + y) / 2;  
}
```

## 函数

❖ 发生函数调用时，主调函数把**实参的值**传递给被调函数的形参，从而实现主调函数向被调函数的数据传送

- 实参可以是常量、变量、表达式。
- 实参和形参必须在类型、数量、顺序上严格一致。
- 当传递实际参数本身而不是其地址时，函数调用中发生的数据传送是单向的。即只能把实参的值传递给形参，而不能把形参的值反向地传回给实参。

❖ 数组同样可以作为函数的参数从主调函数向被调函数传递。此时，数组变量的类型在主调函数和被调函数中必须相同。

❖ 当数组作为函数的参数时，只传递数组的首地址，而不是将整个数组元素都复制到函数中去，即用数组名作为实参，调用函数时该数组第一个元素的地址就被传递给被调函数。 例10-24 数组作为函数的参数

## 函数

例10-25 传递变量地址

```
#include <stdio.h>

int main()
{
    int i = 1, j = 2;
    swap(&i, &j);
    printf("%d %d\n", i, j);
    return 0;
}

void swap(int *a, int *b)
{
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
    return;
}
```

## 外部函数

```
//main.c
#include<stdio.h>

extern float average(float, float);

int main()
{
    float a, b;
    scanf("%f %f", &a, &b);
    printf("the average of %5.1f and %5.1f is %5.1f\n", a, b,
        average(a, b));
    return 0;
}

//fun.c
float average(float x, float y)
{
    return (x + y) / 2;
}
```

## 命令行参数

❖ 在命令行方式下运行该程序的时候，可以通过在文件名后面加入命令行参数来直接为主函数提供参数

**main(int argc, char\* argv[])**

- argc: 存放实际输入的参数个数（包括可执行文件名）
- argv: 每个字符指针都指向一个实际输入的字符串（用空格分隔）

**dir -ah**

argc = ?  
argv[0] = ?  
argv[1] = ?

## 指向函数的指针

❖ 指向函数的指针可简称为函数指针，它存放同类型函数的入口地址（即函数名），可以通过这个地址以间接访问的形式来调用所指向的函数

函数类型（\*指针变量名）（形式参数列表）；

❖ 函数指针只能指向具有相同返回值类型且具有相同形式参数（数量和对应的数据类型均相同）的函数

```
float (*fp) (float a, float b);  
float sum (float x, float y);  
fp = sum;  
(*fp) (1.0, 2.0); 或 fp (1.0, 2.0);
```

❖ 例10-26 命令行参数和指向函数的指针

能源与动力工程学院

# Thank You !