

# 3.函数与指针

杨振平

- 
- ▶ 一个函数在编译时被分配一个入口地址，这个入口地址就称为函数的指针。在C++中，函数名代表函数的入口地址。

(1) 指针作函数的参数-实现地址传递

(2) 返回指针的函数

(3) 利用指向函数的指针调用函数

# 指针作函数的参数-实现地址传递

---

指针作函数的参数，具有以下用途：

- 指针作函数参数，这时形参接受的是实参的地址。函数中通过对指针的间接访问实现参数的按“引用传递”功能。
- 设置多个指针参数可从函数中带回多个结果值。
- 对于传递一块连续的内存区域数据，传递首地址比传递数据值不仅开销小而且效率高。

例：编写交换两个变量值的函数（地址传递）

# 关键点描述:

```
void swap(int *xp, int *yp)
```

形参为指针变量

```
{    int t;  
    t= *xp;  
    *xp=*yp;  
    *yp=t;  
}
```

交换时通过间接访问运算符

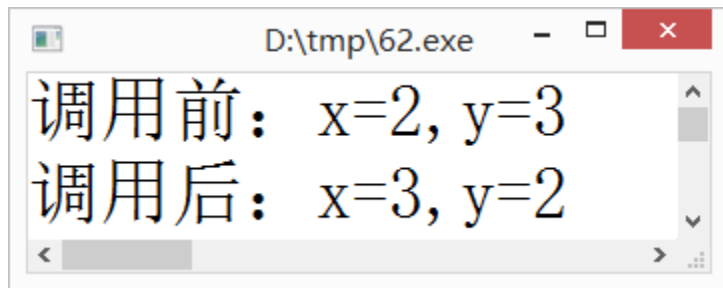
调用时:

```
swap(&x,&y);
```

实参为变量的地址

# 调用swap函数的主函数

```
int main()
{
    int x=2,y=3;
    cout<<"调用前: x="<<x<<",y="<<y<<endl;
    swap(&x,&y);
    cout<<"调用后: x="<<x<<",y="<<y<<endl;
    return 0;
}
```



# 例：利用指针参数带回函数中的多个值

计算一维数组元素的平均值，并能带回数组中的最大值与最小值。

函数原型设计如下：

```
double faver(int s[],int n,int *max,int *min);
```

其中： s—一维数组

n—数组中元素个数

max—指向最大值

min—指向最小值

将平均值作为函数的返回值

# faver函数

```
double faver(int s[],int n,int *max,int *min)
{
    double aver=s[0];
    *max=*min=s[0]; } 变量定义及初始化
    for(int i=1;i<n;i++)
    {
        aver+=s[i];
        if(s[i]>*max)*max=s[i];
        if(s[i]<*min)*min=s[i];
    }
    return aver/n;
}
```

# 主函数

```
int main()
{
    int a[5]={80,89,67,76,98},max,min;
    double aver;
    aver=faver(a,5,&max,&min); //调用
    cout<<"max="<<max<<"\n"<<"min="<<min<<endl;
    cout<<"aver="<<aver<<endl;
    return 0;
}
```

