



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

C++程序设计

Programming in C++



1011018

主讲：魏英，计算机学院

指针与函数

- ◆ 1、指针作为函数的参数.....
- ◆ 2、数组作为函数的参数.....

- ▶ 指针最重要的应用是作为函数参数，它使得被调函数除了返回值之外，能够将更多的运算结果返回到主调函数中。
- ▶ 因此：
- ▶ 指针是函数参数传递的重要工具。

- ▶ 函数形参可以是指针类型，一般形式为：

```
返回类型 函数名(指向类型 *指针变量名, ...)  
{  
    函数体  
}
```

19.1 指针作为函数的参数

【例19.1】设计一个函数swap，用于交换两个变量的值。

```
1  #include <iostream>
2  using namespace std;
3  void swap(int *p1, int *p2)
4  {
5      int t;
6      t=*p1 , *p1=*p2, *p2=t; //交换*p1和*p2
7  }
8  int main()
9  {
10     int a, b;
11     cin>>a>>b; //输入
12     if (a>b) swap(&a, &b);
13     cout<<"min="<<a<<" ,max="<<b; //输出
14     return 0;
15 }
```

19.1 指针作为函数的参数

【例19.1】以下方法是错误的。

```
1  #include <iostream>
2  using namespace std;
3  void swap(int p1, int p2)
4  {
5      int t;
6      t=p1 , p1=p2, p2=t;
7  }
8  int main()
9  {
10     int a, b;
11     cin>>a>>b;
12     swap(a, b);
13     cout<<"min="<<a<<" ,max="<<b;
14     return 0;
15 }
```

- ▶ 通过将指针作为函数参数的方法，既可以返回多个运算结果，又避免了使用全局变量。

19.1 指针作为函数的参数

【例19.2】计算a和b的平方和、自然对数和、几何平均数、和的平方根。

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  double fun(double a, double b, double *sqab, double *lnab, double *avg)
5  {
6      *sqab=a*a+b*b; /*sqab返回平方和
7      *lnab=log(a)+log(b); /*lnab返回自然对数和
8      *avg=(a+b)/2; /*avg返回几何平均数
9      return (sqrt(a+b)); //函数返回和的平方根
10 }
11 int main()
12 {
13     double x=10,y=12,fsq,fln,favg,fsqr;
14     fsqr=fun(x, y, &fsq, &fln, &favg);
15     cout<<x<<" "<<y<<" "<<fsq<<" "<<fln<<" "<<favg<<" "<<fsqr<<endl;
16     return 0;
17 }
```


- 1. （一维或多维）数组作为函数的形参，例如：

```
double average(double *a, int n)
{
    ... //函数体
}
```

- 函数调用形式如下：

```
double X[100], f;
f = average(X, 100);
```

19.2 数组作为函数的参数

【例19.3】编写函数average，返回数组n个元素的平均值。

```
1  #include <iostream>
2  using namespace std;
3  double average(double *a, int n)
4  { //等价于average(double a[], int n)
5      double avg=0.0, *p=a;
6      int i;
7      for (i=1;i<=n;i++,p++) avg=avg+*p ; //等价于avg=avg+p[i]
8      return n<=0 ? 0 : avg/n ;
9  }
10 int main()
11 {
12     double x[10]={66,76.5,89,100,71.5,86,92,90.5,78,88};
13     cout<<"average="<<average(x,10)<<endl;
14     return 0;
15 }
```

19.2 数组作为函数的参数

- ▶ 要想在子函数中改变主调函数中的数组元素，实参与形参的对应关系有如下4种，这4种情况作用相同。
- ▶ (1) 形参和实参都用数组名，例如：

```
void fun(int x[100], int n); //函数原型  
int a[100];  
fun(a, 100); //函数调用
```

- ▶ (2) 形参用指针变量，实参用数组名，例如：

```
void fun(int *x, int n); //函数原型  
int a[100];  
fun(a, 100); //函数调用
```

- ▶ (3) 形参与实参都用指针变量，例如：

```
void fun(int *x, int n); //函数原型  
int a[100], p=a;  
fun(p, 100); //函数调用
```

- ▶ (4) 形参用数组，实参用指针变量，例如：

```
void fun(int x[100], int n); //函数原型  
int a[100], p=a;  
fun(p, 100); //函数调用
```

- ▶ 2. 字符指针变量作为函数形参
- ▶ 将一个字符串传递到函数中，传递的是地址，则函数形参既可以用字符数组，又可以用指针变量，两种形式完全等价。在子函数中可以修改字符串的内容，主调函数得到的是变化后的字符串。

19.2 数组作为函数的参数

【例19.4】自定义函数实现strcpy函数的字符串复制功能。

```
1  #include <iostream>
2  using namespace std;
3  char *stringcpy(char *strDest, const char *strSrc)
4  {
5      char *p1=strDest;
6      const char *p2=strSrc;
7      while (*p2!='\0') *p1=*p2, p1++ , p2++;
8      *p1='\0';
9      return strDest; //返回实参指针
10 }
```

19.2 数组作为函数的参数

```
11 int main()  
12 {  
13     char s1[80], s2[80], s3[80]="string=";  
14     cin>>s1; //输入字符串  
15     strcpy(s2, s1); //复制s1到s2  
16     cout<<"s2:"<<s2<<endl;  
17     strcpy(&s3[7], s1); //复制s1到s3的后面  
18     cout<<"s3:"<<s3<<endl;  
19     return 0;  
20 }
```

CP 程序设计