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

指针的定义与使用

1、指针的概念

2、指针的定义

- 在程序设计过程中，无论是存入数据还是取出数据都需要与内存单元打交道，计算机通过地址编码来表示内存单元。指针类型就是为了处理计算机的地址数据的。

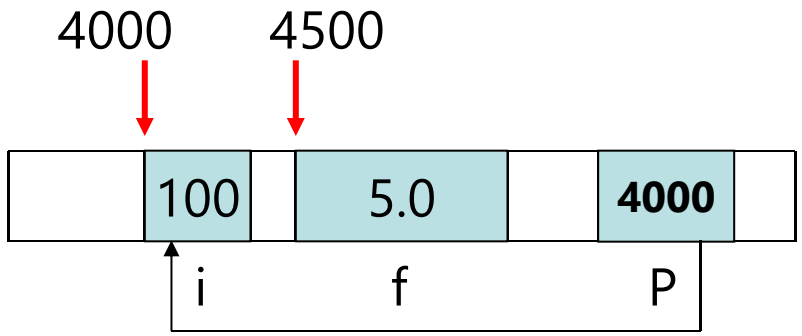
- 指针除了能够提高程序的效率，更重要的作用是能使一个函数访问另一个函数的局部变量，因此指针是两个函数进行数据交换必不可少的工具。

- 1. 地址和指针的概念

- 程序中的数据对象总是存放在内存中，在生命期内这些对象占据一定的存储空间，有确定的存储位置。

- 实际上，每个内存单元都有一个地址，即以字节为单位连续编码。编译器将程序中的对象名转换成机器指令能识别的地址，通过地址来存取对象值。

```
int i; //定义整型变量
double f; //定义双精度浮点型变量
```



- 按对象名称存取对象的方式称为对象直接访问。
- 如：i=100;　f=3.14
- 通过对象地址存取对象的方式称为指针间接访问。

- C++将专门用来存放对象地址的变量称为指针变量，定义形式如下：

```
指针类型 *指针变量名，.....
```

- 例如：

```
int *p, i; //定义p为指针变量，i为整型变量
p = &i ; //指针变量p指向i
```

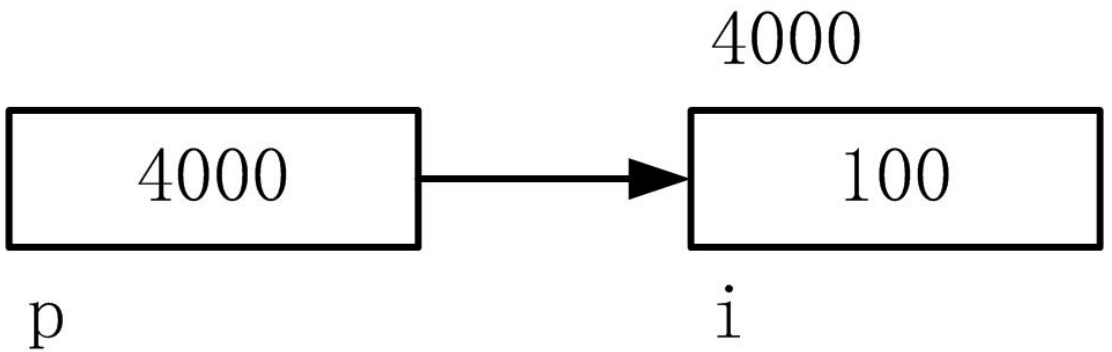
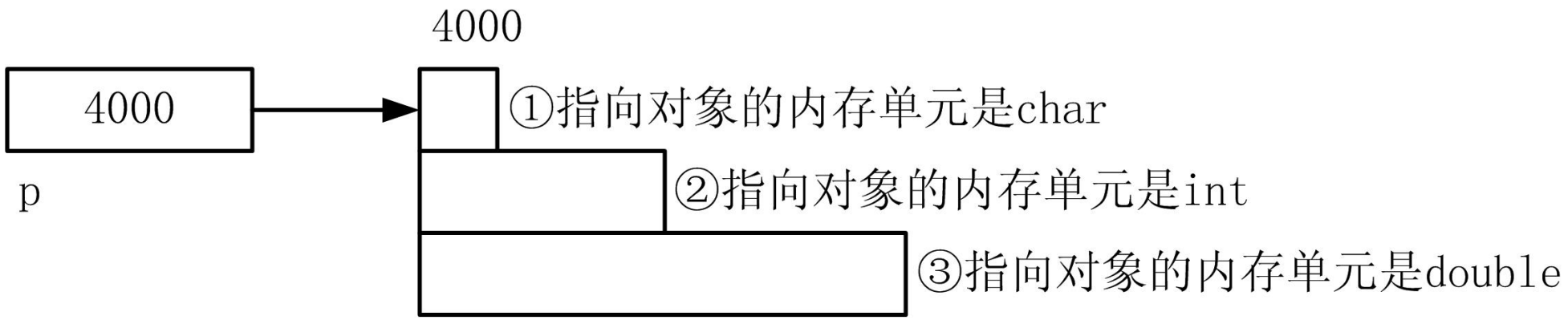


表17-1 取地址运算符

运算符	功能	目	结合性	用法
&	取地址	单目	自右向左	&expr

- 假定指针变量p的值是4000，下面三种写法：

```
①char *p;
②int *p;
③double *p;
```

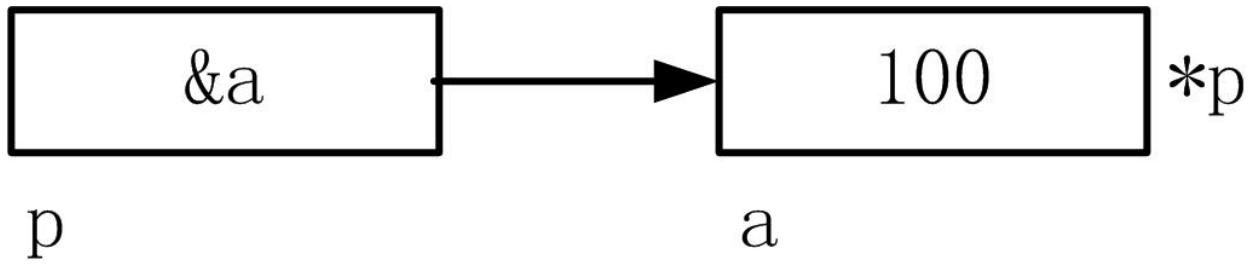


- 2. 指针的间接访问
- 通过间接引用运算（*）可以访问指针所指向的对象或内存单元

表17-2 间接引用运算符

运算符	功能	目	结合性	用法
*	间接引用	单目	自右向左	*expr

```
int a, *p=&a;
a=100; //直接访问a（对象直接访问）
*p=100; //*p就是a，间接访问a（指针间接访问）
*p=*p+1; //等价于a=a+1
```



【例17.1】已知：

```
int a, b, *p1=&a, *p2;
```

则①&*p1的含义是什么？②*&a的含义是什么？

&*p1 ⟷ p1 ⟷ &a

*&a ⟷ a ⟷ *p

【例17.2】指针举例。

```
1   int main()
2   {
3       int i=100, j=200;
4       int *p1, *p2;
5       p1=&i, p2=&j; //p1指向i, p2指向j
6       *p1 = *p1 + 1; //等价于i=i+1
7       p1=p2; //将p2的值赋值给p1, 则p1指向j
8       *p1 = *p1 + 1; //等价于j=j+1
9       return 0;
10  }
```

- 3. 指针的初始化
- 可以在定义指针变量时对其初始化，一般形式为：

```
指针类型 *指针变量名=地址初值,.....
```

```
int a;
int *p=&a; //p的初值为变量a的地址
int b, *p1=&b; //正确，p1初始化时变量b已有地址值
```

- 由于指针数据的特殊性，其初始化和赋值运算是有约束条件的，只能使用以下四种值：
- （1）0值常量表达式，例如：

```
int a, z=0;
int *p1=a; //错误，地址初值不能是变量
p1=z;//错误，整型变量不能作为指针，即使值为0
p1=4000; //错误，整型数据不能作为指针
p1=null; //正确，指针允许0值常量表达式
p1=0; //正确，指针允许0值常量表达式
```

- （2）相同指向类型的对象的地址。例如：

```
int a, *p1;
double f, *p3;
p1=&a; //正确
p3=&f; //正确
p1=&f; //错误，p1和&f指向类型不相同
```

- （3）相同指向类型的另一个有效指针。例如：

```
int x, *px=&x; //正确
int *py=px; //正确，相同指向类型的另一个指针
```

- （4）对象存储空间后面下一个有效地址，如数组下一个元素的地址。

```
int a[10], *px=&a[2]; //正确
int *py=&a[++i]; //正确，相同指向类型的另一个指针
```

C程序设计