



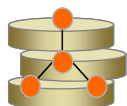
第七单元（1）：指针变量的定义和引用

教学内容	指针变量的定义和引用
教学目标	
应知	指针的意义及使用
应会	单个变量的指针应用
难点	通过指针引用变量的意义

教学方法

直观教学法，将指针的内容在程序中直接打印输出进行观察，去除指针的神秘性

教学过程



1. 定义两个整型变量 a、b,从键盘输入，并且输出 a、b 的值及 a、b 的地址
（澄清变量的地址和变量的值的概念）

2. 定义两个整型指针变量 p1、p2，分别指向变量 a,b,从键盘输入数据并输出 p1、p2 和 *p1、*p2（观察结果，并与第一步的结果对比，引入指针的概念，定义、赋值、引用方式）



（学生易错：定义指针后不赋值就引用）

3. 在程序中输出 &*p1, &*p2, *&a, *&b 比较输出的结果，熟悉运算符 & 和 *

4. 学生操作练习：分别定义浮点数、字符型变量及其指针，并采用指针进行输入输出。（进一步熟悉指针的定义、赋值、引用）

5. 将变量 a、b 的值按从大到小顺序输出，分别以以下三种方式进行，是否都能实现？比较有什么不同（学习指针的应用，进一步理解指针）

1. 以普通变量进行交换
{ t=a; a=b; b=t; }

2. 以普通变量作中间变量，使用指针所值变量的值进行交换
(t=*p1; *p1 = *p2; *p2 = t;)

3. 以指针变量作中间变量，使用指针进行交换
{ p = p1; p1 = p2; p2 = p; }



此法应定义 `int *p=&t;`

6. 将以上各步的整型变量改为字符型,再进行各步骤的操作
7. 将浮点型变量 a、b 的值按从大到小顺序输出,分别以以下三种方式进行,比较有什么不同
 1. 以普通变量进行交换
`{ t=a; a=b; b=t; }`
 2. 以普通变量作中间变量,使用指针所值变量的值进行交换
`(t=*p1; *p1 = *p2; *p2 = t;)`
 3. 以指针变量作中间变量,使用指针进行交换
`{ p = p1; p1 = p2; p2 = p; }`
此法应定义 `float *p=&t;`



学生容易出错的地方

- ❑ 定义指针后不经过赋值直接使用
- ❑ 指针与指向的数据类型不同
- ❑ 错用 & 和 *

问题与讨论

- ❑ 指针定义后为什么一定要复制才能使用?
- ❑ 如果不赋值直接使用可能会发生什么后果?
- ❑ 为什么指向某个变量的指针一定要同变量的类型一致?

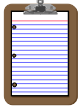




小结（可由问题与讨论方式给出）

- ❑ 变量的存取方式有直接访问（通过变量的名称访问变量）和间接访问（通过变量的指针访问变量）两种
 1. 变量的指针是变量的首地址
 2. 指针变量用来存放变量的地址
- ❑ 指针变量的定义也必须要指定类型,表示指针所要指向的变量数据类型
- ❑ 在使用指针变量之前,必须为其赋值,即使指针变量指向某一个确定的值
- ❑ & 为取地址运算符, * 为指针运算符（也叫取值运算符）



课后任务



-  输入三个整数，按由小到大的顺序输出，要求用指针实现
-  整理笔记