本堂课的内容：

1. 结构体
2. 结构体与指针
3. Typedef 与结构体
4. 公用体 与 枚举

# 结构体

## 定义：

学生为例来说明学生这个结构如何表示。

学生具有信息有： 姓名，年龄，性别，班级。

C 语言定义的基本数据类型是无法来表示这些数据的，数组也不行，数组是表示一组相同数据类型的集合。如何表示这组不同数据类型集合呢？ 结构体出现了。

Struct.

定义方式：

Struct student ｛

Char name[20];

Int age;

Char gender[7];

Int class;

｝

此时的编译器并没有给这个结构体分配内存空间

结构体变量声明：

三种变量定义方式：

一：

Int a；

Struct student Lilei；

二：

全局变量定义方式

Struct student ｛

Char name[20];

Int age;

Char gender[7];

Int class;

｝Lily;

这个时候内存就会分配一片内存给 全局变量

三：

直接定义结构体类型变量

Struct ｛

Char name[20];

Int age;

Char gender[7];

Int class;

｝student1， student2；

这个就是匿名结构体，它的作用是：锁定结构体的变量的数量

注意事项：

1. 类型与变量是不同的概率，不能混淆。结构体名代表的是一个结构体类型， 它不会被分配内存空间。所以，它就不能进行赋值，存取。
2. 对结构体中的成员可以单独的使用。

## 结构体变量的初始化和引用

初始化

和其他类型的变量一样，可以在定义时初始化结构体。 举例说明 尤其是字符串数组的赋值

可以在定义后，使用前初始化。举例说明。

结构体变量成员引用

举例说明：

## 结构体数组

和其他类型的数组一样，可以在定义结构体的时候同时申请数组变量

举例说明。

初始化的方式：

Student stu[3] = {……}; 这种方式的初始化是可行的，系统会自己去将这些数据按照位置匹配

但是，这些数据并不是安全，如果漏了一个，或者位置错了，系统不会察觉，容易产生运行时的错误。

Student stu[3] = ｛｛｝，｛｝，｛｝｝；

## 结构体指针

一个结构体变量的指针代表的意义：这个变量所占据的内存段的起始地址。

举例说明如何使用。

## 结构体中添加函数指针成员变量

结构体非常像java 的类，类有成员变量和成员函数，而结构体也可以拥有这些，只不过我们使用最多的还是结构体的成员变量。

## 结构体中添加结构体指针

这个是最常见的一种情景了，在C语音中链表，堆栈，队列，树，图，都是使用了这样的结构。

就以单链表来举例说明看看这个数据结构的使用方式。

举例说明。

# Typedef 使用

Typedef 是一个功能，它用于建立新的数据类型名。

例如：

一：

Typedef int length；

Length len， maxlen； 等价于 int len，maxLen；

二：

Typedef char \* String；

这个时候我们就创建了一个String

String p;

String str;

p = (String) malloc(100);

strcpy(p, str);

三：

Typedef int（\*PFI）(char \*, char \*）

这样就定义了一个类型PFI，“一个指向函数的指针，该函数具有两个char \* 类型的参数，返回值类型是int ”。

定义的好处，我们可以用字符串操作符的使用举例说明

PFI strcmp，strcpy；

四：定义一个二叉树的数据结构

Typedef struct tnode \* Treeptr;

Typedef struct tnode{

Char \*word;

Int count;

Treeptr left;

Treeptr right;

} TreeNode;

分解上面的例子。

强调的是：typedef 并没有创建新的类型，只是给一个已经存在的类型创建了一个新的名字。可以举例jni中的使用进行 说明

综上所述：

1. Typedef 可以使程序参数化，以提高程序的可移植性。如果typedef 所声明的数据类型和机器相关，那么在有了这个统一的别名后，使用这个别名的数据都不需要修改了，而只需要修改typedef 的类型定义就好了，这个思想是不是有点像我们的面向对象啊。
2. 为程序提供了更好的说明性，且代码书写更简便。在例子TreeNode 来进行说明，

# 公用体

如何用一个变量保存不同类型和长度的对象的变量。

公用体又叫联合体，由关键字union进行定义。

作用是让不同类型的对象公用一个内存单元，。

例如：

Union data ｛

Int a；

Char b；

Int c；

｝；

目的是为了节省内存，但是在同一时刻我们只能使用一个成员。

data存储空间的大小就是共用体中，内存最大的变量所占用的字节数，所以，这个data的大小是4字节。

&Data.a

&Data.b

&Data.c

他们的值是一样的