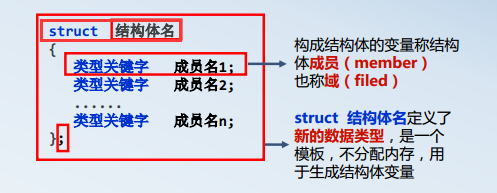
**11.1 结构体**

1. 结构体是一种自定义数据类型，结构体变量代表的结构体类型的数据对象作为一个整体存储在内存中。

创建结构体对象：

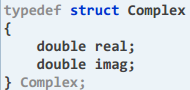
（1）结构体类型 结构体变量

（2）及认购提类型\*结构体变量=（结构体类型）malloc(sizeof(结构体类型))  
2. ❖结构体类型定义的基本语法形式为：



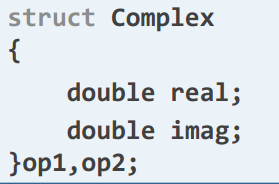
3. 结构体定义可以嵌套

4. 成员变量的类型**不能和本结构体类型相同，即不能递归定义。**结构体的成员变量可以是**本结构体类型的指针变量**。  
5. 可以用typedef简化struct Complex变量的定义形式



Complex就是struct Complex

在定义结构体的同时定义结构体变量



❖**struct student 是一个自定义的数据类型**❖**struct student s1；是一个结构体变量**❖**struct student stu[100]; 是一个结构体数组**

6. 在定义结构体变量的时候可以按照成员的顺序和类型对成员整体赋初值。（struct Complex c1 = {1.4, 2.5};）

或者在定义结构体变量后，单独给成员变量赋值。

 struct Complex c1 ;

c1.real = 1.4;

c1.imag = 2.5;

同类型结构体变量可相互赋值(不涉及到指针成员)

struct Complex c1, c2 ;

c1.real = 1.4;

c1.imag = 2.5;

c2 = c1;

7. 即使两种结构体类型的成员列表完全相同，他们对应的变量也不可以使用 = 进行赋值。

8. 若是普通结构体变量，则访问其成员的方式为：

– 结构体变量名.成员名

若是通过结构体指针访问结构体对象，则形式为：

– 结构体指针名->成员名

– （\*结构体指针名）.成员名

结构体砸内存中所占空间与每一个变量及其类型有关（成员个数 每个成员所占字节数+空隙）

10. 结构所占的实际空间一般是按照编译器默认对齐方式（对

齐系数）来进行内存分配的

成员的对齐步骤：

1 判断结构体成员的大小是否小于等于对齐系数，小于等于执行到2，否则到3；

2 判断预存放结构体成员的首地址相对于结构体首地址的偏移量是否是本成员大小的整数倍，若是则存放本成员，否则到4；

3 判断预存放结构体成员的首地址相对于结构体首地址的偏移量是否是对齐系数的整数倍，若是则存放本成员，否则到4；

4 在本成员和上一个成员之间填充一定的空字节以达到整数倍的要求再存放本成员。

整体的对齐步骤：

1 判断结构体成员中最大成员的大小是否小于等于对齐系数，小于等于执行到2，否则到3；

2 判断结构体各成员成员对齐后长度之和是否是最大成员大小的整数倍，是则不进行整体对齐，否则到4；

3 判断结构体各成员成员对齐后长度之后是否是对齐系数的整数倍，是则不进行整体对齐，否则到4；

4 在最后一个成员后填充一定的空字节以达到整数倍的要求，整体对齐完成

当结构体中某成员为数组时，则在处理结构体的对齐时，该成员数组在求整数倍关系时使用数组类型的大小。

当结构体中某成员为结构体时，则在处理结构体的对齐时，该成员结构体在求整数倍关系时使用该成员结构体中最大成员的大小。