

# 成绩排序

例、输入n个学生的学号、姓名、考试成绩，找出最高分和最低分者，输出他们的信息

样例输入

```
5
1 Limei 87
2 Allen 64
3 Bob 79
4 Mike 91
5 Hanlei 67
```

样例输出

```
4 Mike 91
2 Allen 64
```

Q:如何存储n个学号、姓名、成绩?

学号: `int a[100];`

姓名: `char b[100];`

成绩: `int c[100];`

# 排序过程

过程描述：

选择一个排序算法(从小到大)

{

比较成绩

{

1.交换成绩

2.交换学号

3.交换姓名

.....

}

}

交换学生整体信息

Q: 有没有一种数据类型  
能存储学生所有的信息呢?

自定义数据类型

# 自定义数据类型

---

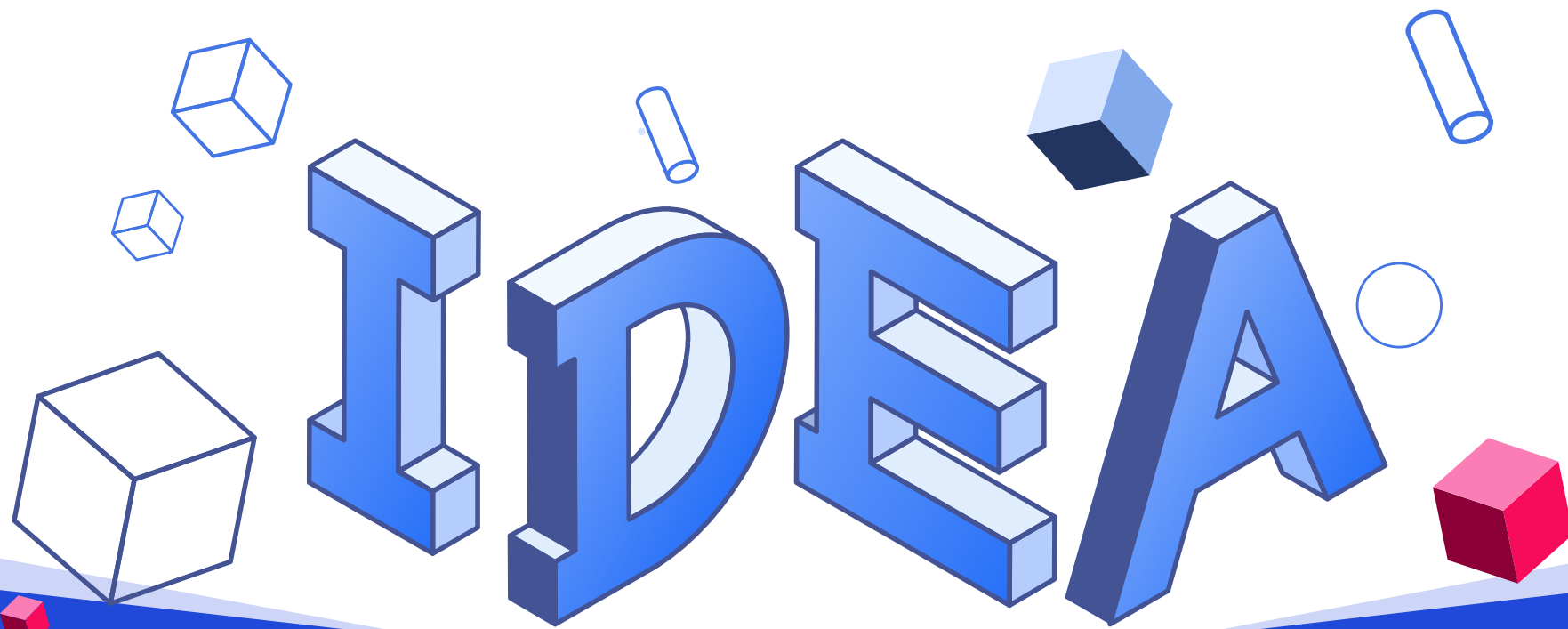
## 自定义数据类型

1. 结构体类型(struct)

2. 共用体类型(union)

3. 枚举类型(enum)

.....



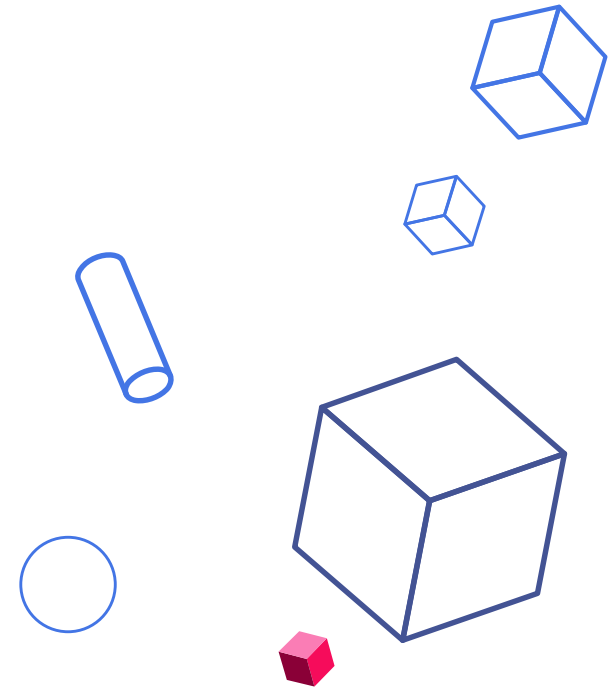
# 信息学

# 结构体与sort排序

西南大学附属中学校

信息奥赛教练组

# 01 struct结构体

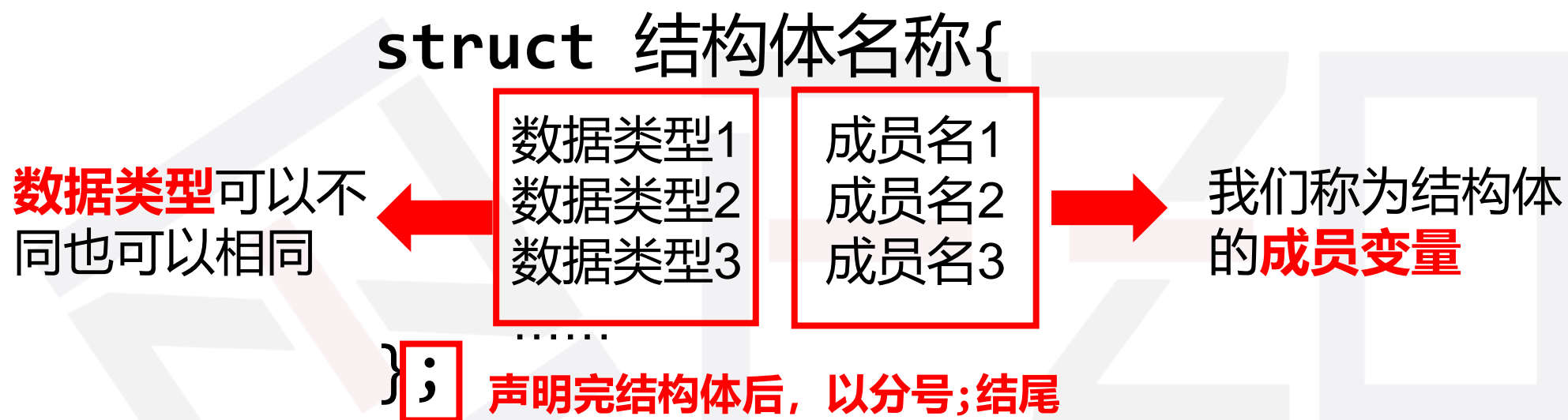


# 结构体(struct)

将若干相同或不同的数据类型存放在一起，成为一个整体

| 西 | 大 | 附 | 中 | 信 | 息 | 学 | 竞 | 赛 |  
High School Affiliated to Southwest University

# 声明结构体



# 例子

**例** 声明一个学生信息的结构体

```
struct Student{  
    int  snum;      //学号  
    char name[100]; //姓名  
    int  score;     //成绩  
}; .....
```

Student		
snum	name[]	score



# 结构体变量的定义

---

一般数据类型变量的定义： 数据类型 变量名；

结构体类型变量的定义： **结构体名称** 结构体变量名；

# 结构体变量的定义

以学生信息为例：

1. 定义一个学生的信息

```
Student a; { int snum; //学号  
              char name[100]; //姓名  
              int score; //成绩
```

2. 定义一组学生的信息

```
Student a[105]; { a[0] { int snum; //学号  
                  a[1] { char name[100]; //姓名  
                  a[2] { int score; //成绩  
                  ...  
                  a[3]  
                  ...  
                  a[n]
```

# 结构体变量的访问

## 1.成员访问

结构体变量名 . 成员变量



成员访问符

例：输入一个学生的信息

```
Student a;
```

```
cin>>a.snum>>a.name>>a.score;
```

```
scanf("%d %s %d",&a.snum,a.name,&a.score);
```

# 结构体变量的访问

## 2.结构体整体访问

结构体变量赋值:

```
Student a;  
a={1, "Limei", 87};
```

```
struct Student{  
    int  snum;      //学号  
    char name[100]; //姓名  
    int  score;     //成绩  
};.....
```

无需——访问每个成员变量,  
按顺序——匹配进行赋值

# 结构体变量的访问

## 2.结构体整体访问

两个结构体类型相同的结构体变量可以互相赋值

- 1 Student a;  
Student b;  
a=b;
- 2 Student a[10];  
a[1]=a[2];

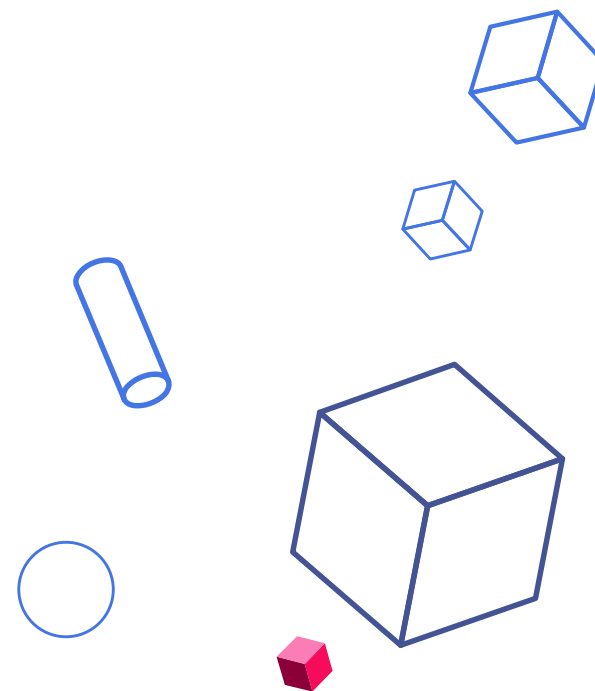
整体访问仅限赋值操作，**输入输出还需采用成员访问的方式**

# 小结

---

- 1.结构体能将多个有联系的变量，结合成一个整体
- 2.结构体能方便我们进行一些操作，比如交换、更新操作
- 3.结构体变量的定义：结构体名称 结构体变量名;
- 4.结构体变量有成员访问和整体访问两种方式，注意整体访问适用的情景

## 02 STL之sort()函数



# 什么是STL?

---

**STL**是Standard Template Library的简称，中文名标准模板库

STL提供了大约100个实现算法的模版函数，如果我们能够掌握，直接调用相应的算法模板可以大大简化我们的代码

例如排序算法：

**排序函数：sort()** 包含在  `#include <algorithm>`



# sort()函数

调用格式

sort(首地址 , 尾地址 , 比较函数)

可以不传入  
默认从小到大

例:排序以下这个数组

```
int a[10]={9,6,3,8,5,2,7,4,1,0};
```

```
sort(a,a+10);
```

排序结果: 0 1 2 3 4 5 6 7 8 9 → 从小排到大

只排了a[0]-a[9]之间的数

可以发现: 排序的区间其实是一个左闭右开区间, [0,10)

## 小练习

---

### 1.排序a[1]~a[10]之间的数字

```
sort(a+1, a+11);
```

### 2.排序a[5]~a[100]之间的数字

```
sort(a+5, a+101);
```

sort函数默认从小排到大，如何让sort**从大排到小**呢？

# cmp比较函数

比较函数的名称可以自己定义，这里函数名为cmp

比较函数里面要写明sort排序的方法

**格式**

```
int cmp(参数1,参数2){
```

排序方法

```
}
```

参数类型与排序的数据类型一致

**cmp函数的返回值只返回0或1：**

- 1.当返回0时，意味着不满足条件，两个参数位置交换
- 2.当返回1时，意味着满足条件，不交换

# 数组排序

---

//情况一：数组排序

```
int a[100];
```

```
int cmp1(int x,int y)//int为排序数组的数据类型
```

```
{
```

```
    return x>y;//从大到小排列
```

```
    //return x<y;//默认的从小到大排列
```

```
}
```

```
sort(a,a+100,cmp1);
```

# 结构体排序

---

## 例、按学生的成绩从大排到小

//情况二：结构体排序

```
Student stu[100];
```

```
int cmp2(Student a, Student b ) //参数传入结构体变量
```

```
{
```

```
    return a.score>b.score; //按照成绩从大到小排列
```

```
}
```

```
sort(stu, stu+100, cmp2);
```

**如果题目要求成绩相同,按照学号从小排到大怎么办?**

## 结构排序(二)

---

//情况二：结构体排序

```
Student stu[100];
```

```
int cmp2(Student a, Student b) //参数传入结构体变量
```

```
{
```

```
    if(a.score==b.score) return a.snum<b.snum; //成绩相同,  
    按照学号从小到大排列
```

```
    return a.score>b.score; //按照成绩从大到小排列
```

```
}
```

```
sort(stu, stu+100, cmp2);
```

# Thanks

## For Your Watching

