

文件

主讲老师：党东



一、什么是文件？



- 文件是根据特定的目的而收集在一起的有关数据的集合。
- C++把每一个文件都看成是一个有序的字节流，每个文件都以文件结束标志结束，如果要操作某个文件，程序必须首先打开该文件。当一个文件被打开后，该文件就和一个流关联起来，这里的流实际上是一个字节序列。



二、文件的操作



■ C++语言提供了一批用于文件操作的标准函数，一般而言文件操作基本步骤如下：

- (1)打开文件，将文件指针指向文件，决定打开文件类型；
- (2)对文件进行读、写操作；
- (3)在使用完文件后，关闭文件。



三、文件的重定向



样例：求任意两个数的和。

输入文件名：sum.in 输出文件名：sum.out

方法一：

```
#include<fstream>
using namespace std;
ifstream cin("sum.in");
ofstream cout("sum.out");
int main(){
    int a,b;
    cin>>a>>b;
    cout<<a+b;
    return 0;
}
```

方法二：

```
#include<iostream>
#include<cstdio>
using namespace std;
int main(){
    freopen("sum.in","r",stdin);
    freopen("sum.out","w",stdout);
    int a,b;
    scanf("%d %d",&a,&b);
    printf("%d\n",a+b);
    fclose(stdin);
    fclose(stdout);
    return 0;
}
```


结构体

主讲老师：党东



为什么需要结构体？



巴蜀中學
BASHU SECONDARY SCHOOL

在实际问题中，一组数据往往具有不同的数据类型。例如，人口大普查时，我们需要记录每一位公民的姓名，年龄，性别，住址，身份证号码。

这些信息分别要用**整型**，**字符型**，**字符串型**来记录。为了解决问题，C++语言给出了另一种**构造数据类型**——“结构体”，它在数据存储方面相当于其他高级语言中的记录，但它有着面向对象的优势。



一、结构体(struct)定义和操作



定义结构体及结构体变量（两种形式）

(1)定义结构体类型的时候同时定义变量

```
struct 结构体类型名{    //其中struct是关键字  
    成员表;                //可以有多个成员  
    成员函数;              //可以有多个成员函数，也可以没有  
}结构体变量表;          //可以同时定义多个结构体变量，用“，”隔开
```

样例

```
struct student{    //定义类型名叫student的struct类型  
    string name;  
    int chinese,math;  
    int total;  
} a[110];          //同时定义了a数组变量
```

一、结构体(struct)定义和操作



(2)先定义结构体再定义结构体变量

struct 结构体类型名{

成员表;

成员函数;

};

结构体名 结构体变量表 //同样可以同时定义多个结构体变量

样例

```
struct student{  
    string name;  
    int chinese,math;  
    int total;  
};  
student a[110];
```

在定义结构体变量时注意，结构体变量名和结构体名不能相同。在定义结构体时，系统对其不分配实际内存。只有定义结构体变量时，系统才为其分配内存。

二、结构体变量的特点



(1) 结构体变量可以**整体操作**，例如：

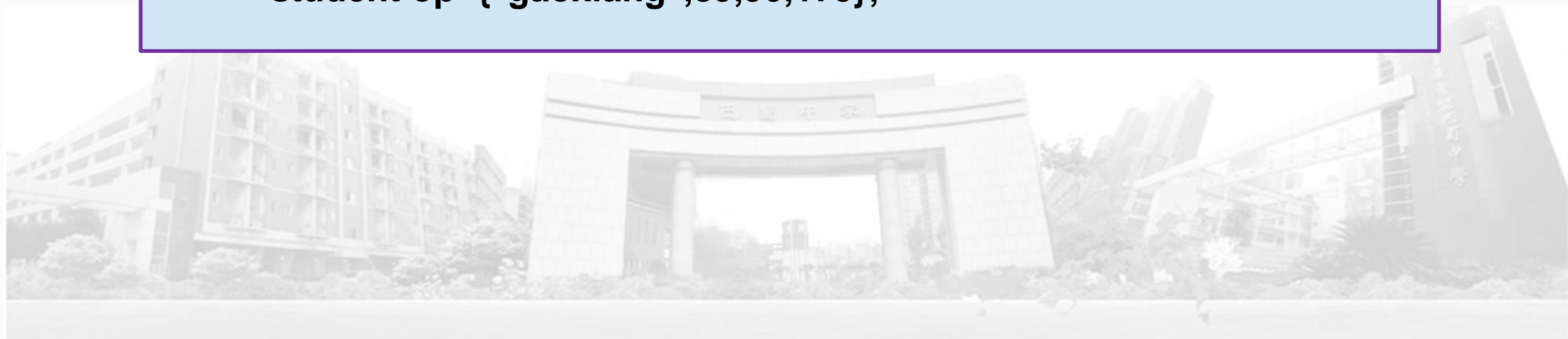
```
swap(a[j],a[j+1]);
```

(2) 结构体变量的成员**访问**也很方便、清晰，例如：

```
cin>>a[i].name;
```

(3) 结构体变量的初始化和数组的**初始化**类似，例如：

```
student op={"gaoxiang",89,90,179};
```



三、结构体成员的调用



结构体变量与各个成员之间引用的一般形式为：

结构体变量名 . 成员名

对于上面定义的结构体变量，我们可以这样操作：

```
cin>>a[i].name;           //一般情况下不能写cin>>a[i];
```

```
    a[i].total=a[i].chinese+a[i].math; //就像用整型变量一样
```

实际上结构体成员的操作与该成员类型所具有的操作是一致的。

成员运算符 “.”在存取成员数值时使用，其优先级最高，并具有左结合性。在处理包含结构体的结构体时，可记作：

```
strua.strub.membb
```

这说明结构体变量strua有结构体成员strub；结构体变量strub有成员membb。

【例1】成绩统计



【问题描述】 输入N个学生的姓名和语文、数学的得分，按总分从高到低输出，分数相同的按输入先后输出。

【输入格式】

第1行，有一个整数N，N的范围是[1...100]；下面有N行，每行一个姓名，2个整数。姓名由不超过10个的小写字母组成，整数范围是[0...100]。

【输出格式】

总分排序后的名单，共N行，每行格式：姓名 语文 数学 总分。

【输入样例】

```
4
gaoxiang 78 96
wangxi 70 99
liujia 90 87
zhangjin 78 91
```

【输出样例】

```
liujia 90 87 177
gaoxiang 78 96 174
wangxi 70 99 169
zhangjin 78 91 169
```

【例1】成绩统计



【分析】 由于姓名是字符串，分数是整数，如果用数组保存，则要两个数组，比如：

```
string name[100];
```

```
int score[100][3];
```

这种方法不利于把一个学生的信息当成一个整体处理。

下面程序中通过使用结构（struct）类型的方法来解决这个问题。

【参考代码】

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
struct student{
```

```
    string name;
```

```
    int chinese,math;
```

```
    int total;
```

```
};
```

```
student a[110];
```

//定义一个struct的类型，类型名叫：student

//定义一个数组a，每个元素是student类型

【例1】成绩统计



【参考代码】

```
int n;
int main()
{
    cin>>n;
    for (int i=0; i<n; i++)    //对结构体中成员的赋值、取值。
    {
        cin>>a[i].name;
        cin>>a[i].chinese >>a[i].math;
        a[i].total=a[i].chinese+a[i].math;
    }
    for (int i=n-1; i>0; i--)
        for (int j=0; j<i; j++)    //冒泡排序
            if (a[j].total<a[j+1].total)
                swap(a[j],a[j+1]);
    for (int i=0; i<n; i++)    //输出
        cout<<a[i].name<<' '<<a[i].chinese<<' '<<a[i].math<<' '<<a[i].total<<endl;
    return 0;
}
```


【延伸】利用STL排序



【参考代码】//如果总分相同时，按语文成绩由大到小排序

```
bool cmp(student a, student b) {  
    if(a.total > b.total) return true; //多关键词排序时做多次判断即可  
    else if(a.total < b.total) return false;  
    else return a.chinese > b.chinese;  
}  
  
int n;  
int main()  
{  
    cin>>n;  
    for (int i=0; i<n; i++) //对结构体中成员的赋值、取值。  
    {  
        cin>>a[i].name;  
        cin>>a[i].chinese >>a[i].math;  
        a[i].total=a[i].chinese+a[i].math;  
    }  
    sort(a,a+n,cmp);  
    for (int i=0; i<n; i++) //输出  
        cout<<a[i].name<<' '<<a[i].chinese<<' '<<a[i].math<<' '<<a[i].total<<endl;  
    return 0;  
}
```

1103-- 【2005提高】谁拿了最多奖学金

1186 -- 【2007普及】奖学金

