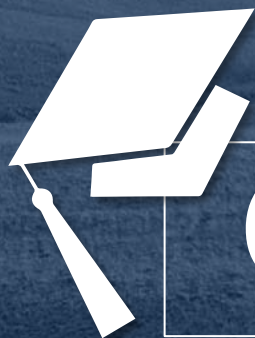


章节复习 | 结构体 | 共用体 | 思维导图



C语言程序设计第七章

——结构体和共用体

主讲人：李伦彬



CONTENTS

章节复习

结构体

共用体

思维导图

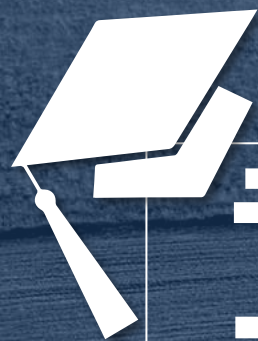


章节复习

结构体

共用体

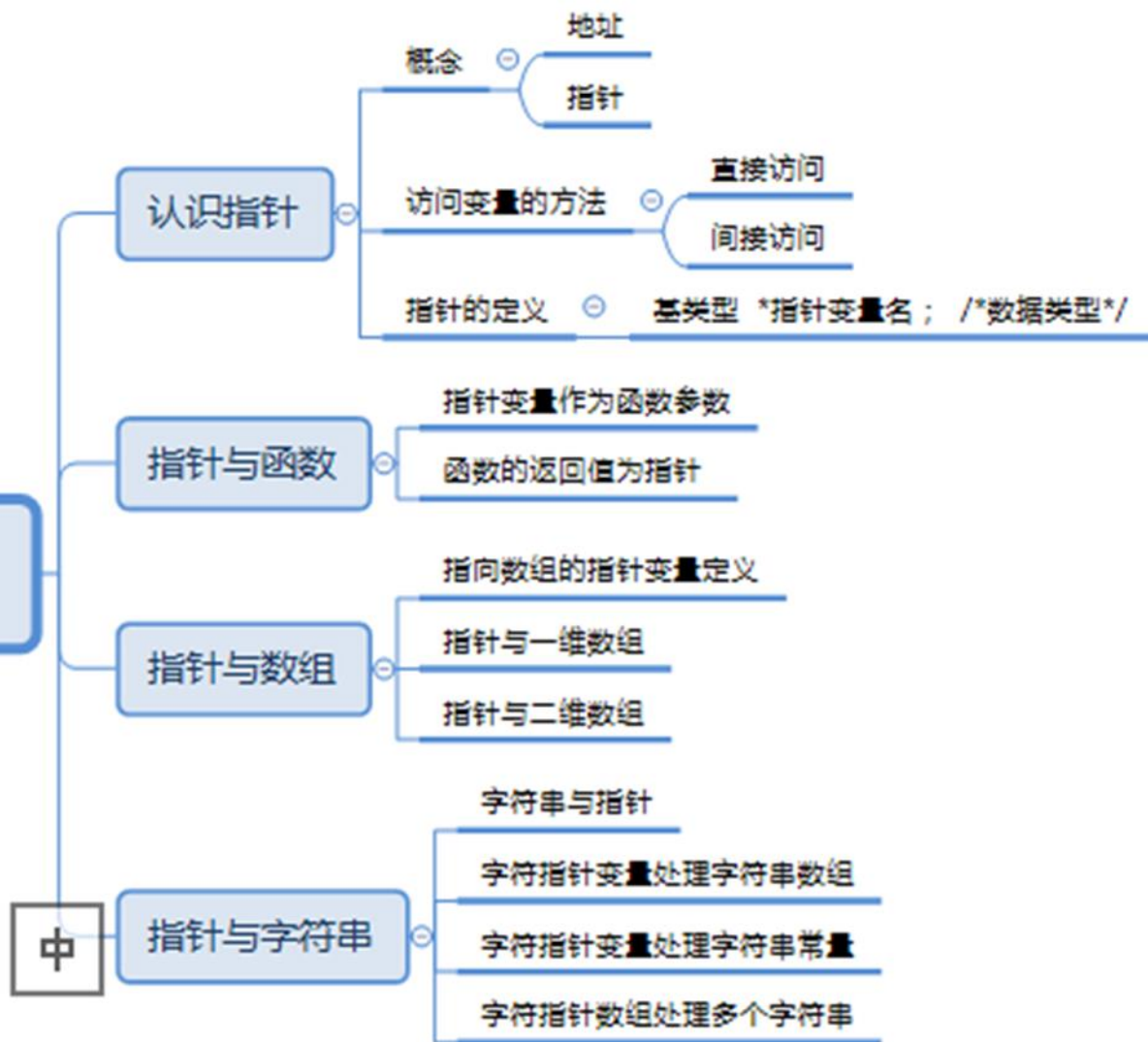
思维导图



章节复习



Chapter06-地址和指针

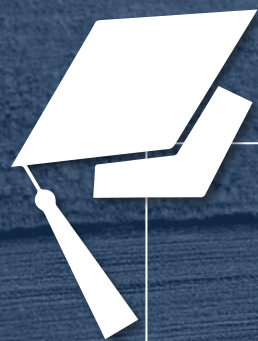


章节复习

结构体

共用体

思维导图



结构体





本节内容

结构体概述

结构体定义

结构体变量初始化



- ❖ 数据的基本类型：整、实、字符。
- ❖ 数组是构造类型：每个元素为同一类型
- ❖ 有些问题仅用基本类型和数组来描述，无法反映其内在联系，如学生情况：

num	name	sex	age	score	addr
整	字符串	字符	整	实	字符串
11001	Zhang xin	m	19	96.5	Shang hai
12001	Wang li	f	20	98.5	Bei jing

结构体概述

- 定义：由**相互关联的不同数据类型**的数据组成的有机整体。
- 结构体是一种**构造数据类型**。与其他高级语言中的“记录”类似。
- 用途：为处理复杂的数据结构提供了手段。为函数间传递不同类型的参数提供了便利。



结构体类型的定义

注意：这只是声明一种数据类型并没有定义变量。

`struct`是关键字，不能省略

；不能省略

`struct` [结构体名]

类型标识符
类型标识符
.....

成员名1;
成员名2;
.....

合法标识符可省略
(无名结构体)

成员类型可以是任何数据类型



结构体类型的定义示例

```
1 struct student
2 { int num;
3   char name[20];
4   char sex;
5   int age;
6   float score;
7   char addr[30];
8 };
9
```

结构体类型**定义**仅描述结构体的组成,
不分配内存空间

结构体类型的变量定义

(1) 先声明结构体类型，再定义结构体变量

```

struct student
{ int num;
  char name[20];
  char sex;
  int age;
  float score;
  char addr[30];
};
struct student stu1,stu2;
```

定义结构体
类型

```

struct 结构体名
{ 类型标识符 成员名;
  类型标识符 成员名;
  .....
};
struct 结构体名 变量名表列;
```

定义结构体**变量**



结构体类型的变量定义

(2) 声明结构体类型的同时定义结构体变量

```

struct student
{
    int num;
    char name[20];
    char sex;
    int age;
    float score;
    char addr[30];
} stu1,stu2 ;
    
```

```

struct      结构体名
{ 类型标识符      成员名;
  类型标识符      成员名;
  .....
} 变量名表列;
    
```

定义结构体**类型**

定义结构体**变量**



结构体类型的变量定义

(3) 直接定义结构体类型变量

```

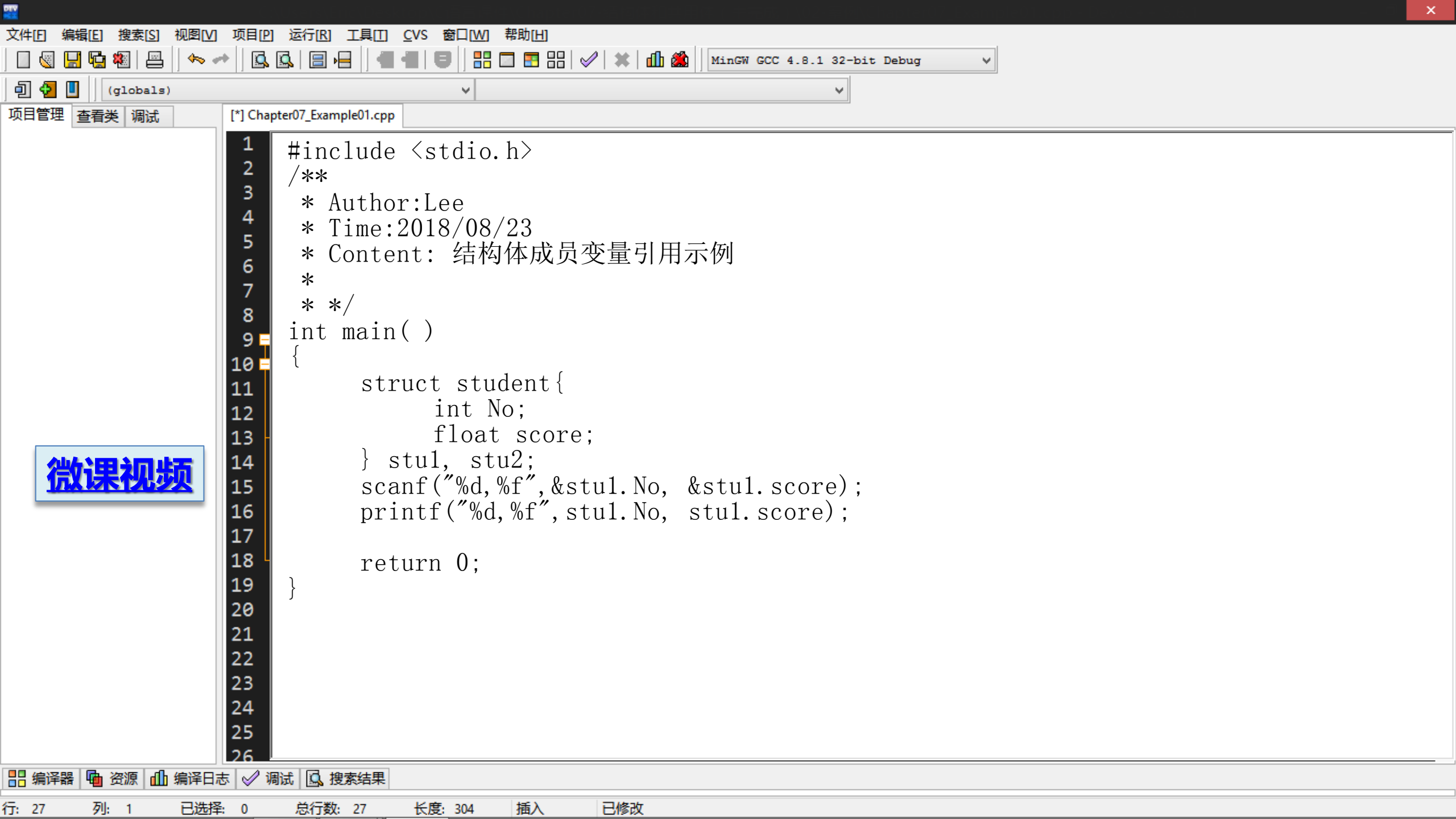
struct
{ int num;
  char name[20];
  char sex;
  int age;
  float score;
  char addr[30];
} stu1,stu2 ;
    
```

```

struct
{ 类型标识符      成员名;
  类型标识符      成员名;
  .....
} 变量名表列;
    
```

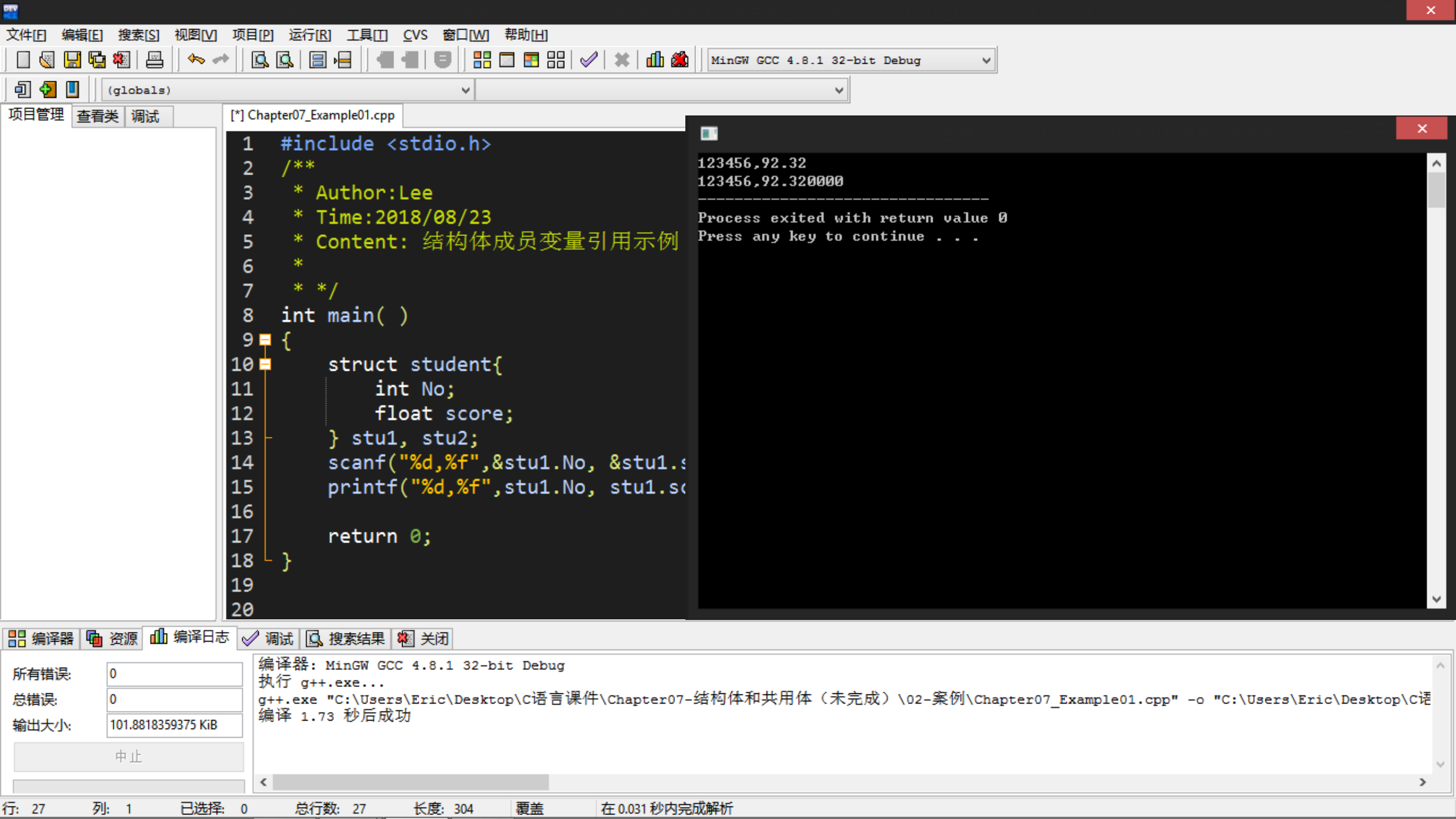
用无名结构体直接定义变量只能一次





微课视频

```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/23
5  * Content: 结构体成员变量引用示例
6  *
7  * */
8 int main( )
9 {
10     struct student{
11         int No;
12         float score;
13     } stu1, stu2;
14     scanf("%d,%f",&stu1.No, &stu1.score);
15     printf("%d,%f",stu1.No, stu1.score);
16
17     return 0;
18 }
19
20
21
22
23
24
25
26
```



```
1 #include <stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/23
5  * Content: 结构体成员变量引用示例
6  *
7  * */
8 int main( )
9 {
10     struct student{
11         int No;
12         float score;
13     } stu1, stu2;
14     scanf("%d,%f",&stu1.No, &stu1.score);
15     printf("%d,%f",stu1.No, stu1.score);
16
17     return 0;
18 }
19
20
```

```
123456,92.32
123456,92.320000
-----
Process exited with return value 0
Press any key to continue . . .
```

所有错误:	0
总错误:	0
输出大小:	101.8818359375 KiB
中止	

编译器: MinGW GCC 4.8.1 32-bit Debug
执行 g++.exe...
g++.exe "C:\Users\Eric\Desktop\C语言课件\Chapter07-结构体和共用体 (未完成)\02-案例\Chapter07_Example01.cpp" -o "C:\Users\Eric\Desktop\C语言课件\Chapter07_Example01.exe" -g -m32 -x86_64-w64-mingw32
编译 1.73 秒后成功

结构体类型变量的初始化

形式一

```
struct 结构体名
{
    类型标识符 成员名1;
    类型标识符 成员名2;
    .....
};
struct 结构体名 结构体变量 = {初始数据};
```


结构体类型变量的初始化

形式一示例

```
1 struct stu
2 { int num;
3   char name[20];
4   int age;
5   char addr[30];
6 };
7 struct stu a={112,"Wang Lin", 19, "200 Beijing Road"};
```

结构体类型变量的初始化

形式二

```
struct 结构体名  
{ 类型标识符 成员名1;  
  类型标识符 成员名2;  
  .....  
} 结构体变量 = {初始数据};
```

结构体类型变量的初始化

形式二示例

```
1 struct stu
2 { int num;
3   char name[20];
4   int age;
5   char addr[30];
6 } a={112,"Wang Lin", 19, "200 Beijing Road"};
7
```



结构体类型变量的初始化

形式三

```
Struct  
{  类型标识符    成员名1;  
   类型标识符    成员名2;  
   .....  
}  结构体变量 = {初始数据};
```

结构体类型变量的初始化

形式三示例

```
1 struct
2 { int num;
3   char name[20];
4   char sex;
5   int age;
6   char addr[30];
7 }stu1={112,"Wang Lin",'M',19, "200 Beijing Road"};
8
```





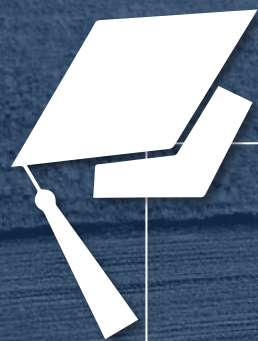
练习巩固

设有以下说明语句

```
struct stu
{
    int a;
    float b;
} stutype;
```

则下面的叙述不正确的是：

- A. struct是声明结构体类型的关键字
 - B. struct stu是用户定义的结构体类型
 - C. stutype是用户定义的结构体类型名
 - D. a和b都是结构体成员名
- C. 结构体变量名



共用体





本节内容

共用体概述

共用体的定义和引用

共用体的特点



共用体概述

□ 使几个不同的变量**共占同一段内存**的结构称为“共用体”类型的结构。

定义共用体类型变量的一般形式为：

```
union 共用体名  
{  
    成员表列  
} 变量表列；
```



共用体类型定义、变量定义、初始化

例如:

```
union data
```

```
{ int i;
```

```
  char ch;
```

```
  float f;
```

```
} a={100};
```

或

```
union data
```

```
{ int i;
```

```
  char ch;
```

```
  float f;
```

```
};union data a={100};
```

注：在定义的同时只能用**第一个成员的类型**的值进行初始化

Union data a={100, ' a' , 1.23} 错误



共用体概述

共用体和结构体的比较：

结构体变量所占内存长度是各成员占的**内存长度之和**。每个成员分别占有其自己的内存单元。

共用体变量所占的内存长度等于**最长的成员的长度**。

共用体变量的引用方式

只有先定义了共用体变量才能引用它，而且不能引用共用体变量，而只能引用共用体变量中的成员。

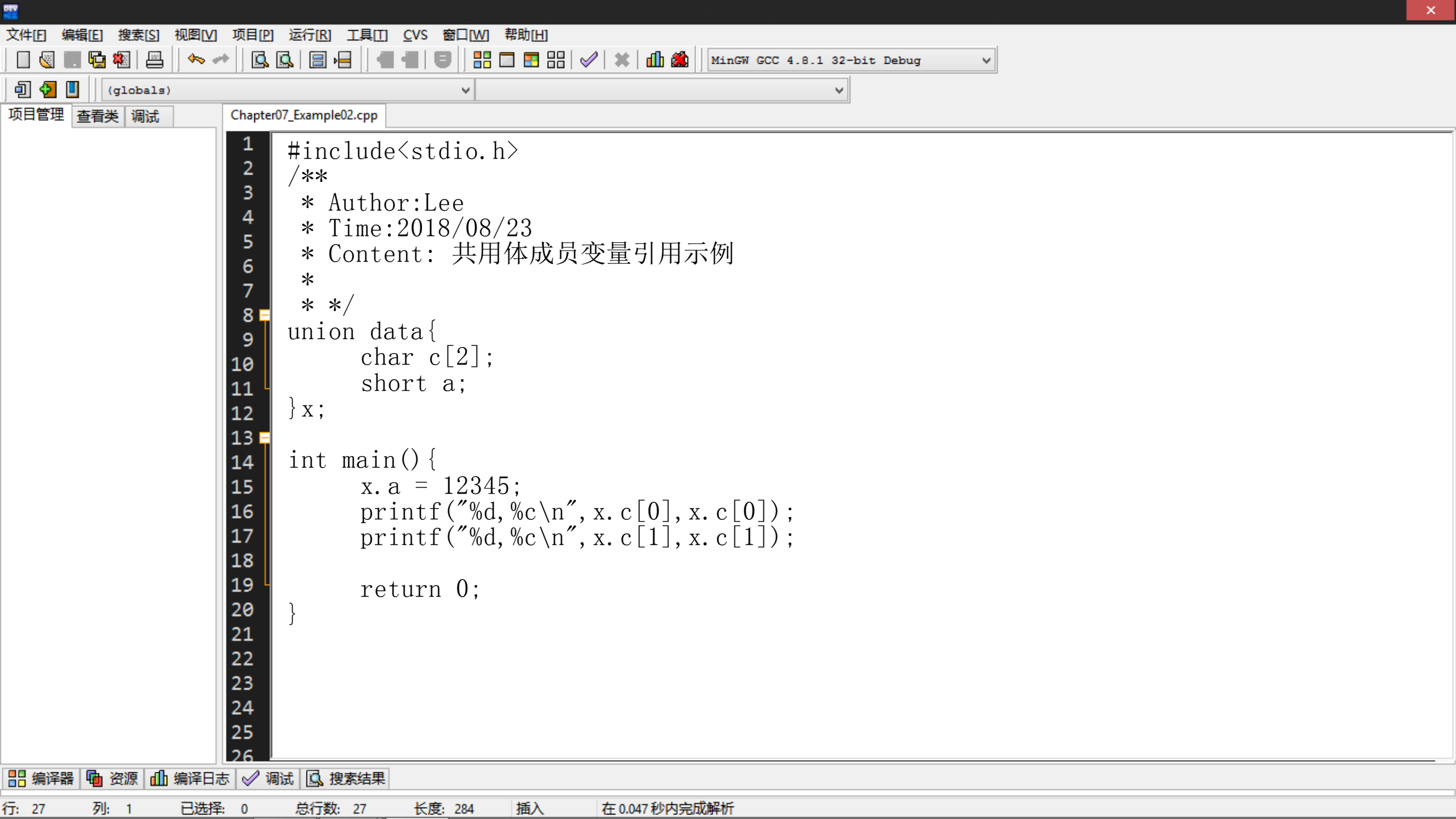
共用体特点

- (1) 同一个内存段可以用来存放几种不同类型的成员，但在**每一瞬时只能存放其中一种**，而不是同时存放几种。
- (2) 共用体变量中起作用的成员是**最后一次存放的成员**，在存入一个新的成员后原有的成员就失去作用。
- (3) 共用体变量地址和它的各成员的地址都是**同一地址**。
- (4) 不能对共用体变量名赋值，也不能企图引用变量名来得到一个值。
- (5) 不能把共用体变量作为函数参数，也不能使函数带回共用体变量，但可以使用**指向共用体变量的指针**。
- (6) 共用体类型可以出现在结构体类型定义中，也可以定义共用体数组。反之，结构体也可以出现在共用体类型定义中，数组也可以作为共用体的成员。

共用体变量的引用

3种形式:

- ① 共用体变量. 成员名
- ② (*p) . 成员名
- ③ p ->成员名



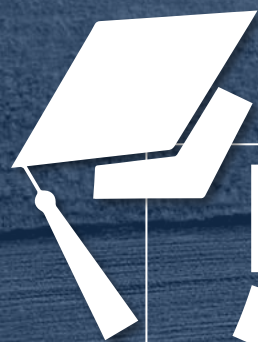
```
1 #include<stdio.h>
2 /**
3  * Author:Lee
4  * Time:2018/08/23
5  * Content: 共用体成员变量引用示例
6  *
7  * */
8 union data{
9     char c[2];
10    short a;
11 }x;
12
13
14 int main() {
15     x.a = 12345;
16     printf("%d,%c\n", x.c[0], x.c[0]);
17     printf("%d,%c\n", x.c[1], x.c[1]);
18
19     return 0;
20 }
21
22
23
24
25
26
```


章节复习

结构体

共用体

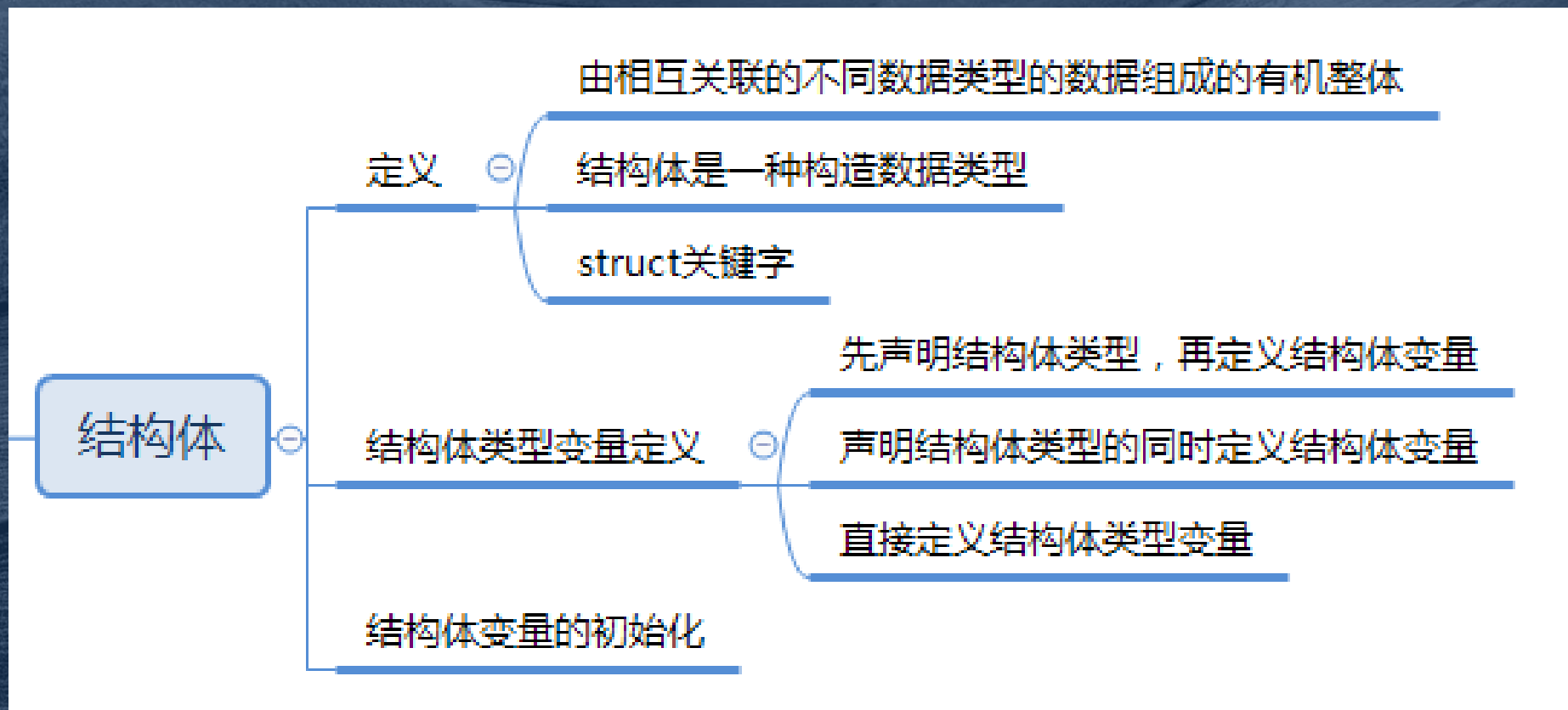
思维导图



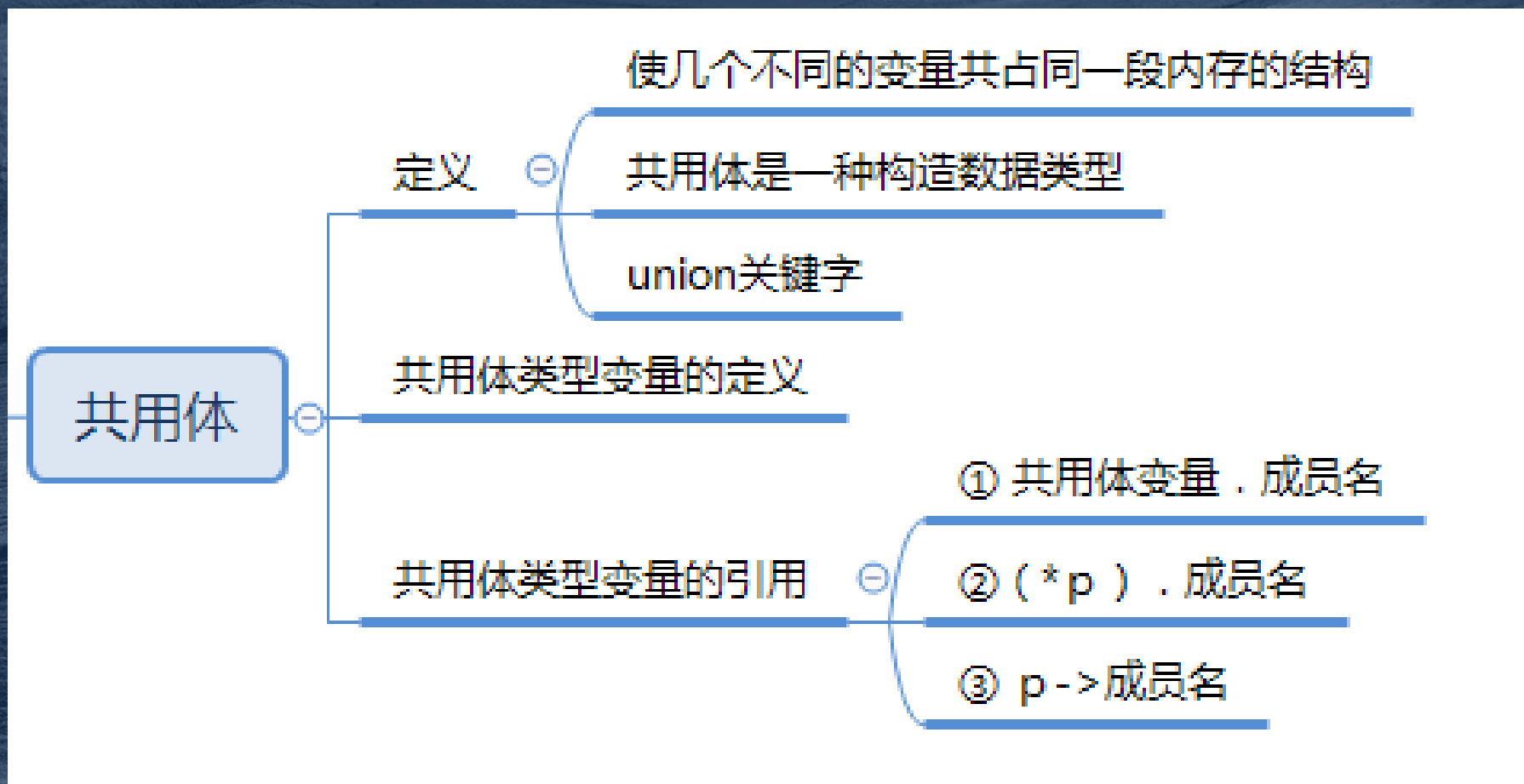
思维导图



结构体部分思维导图



共用体部分思维导图



教学评价

完成本章的作业和练习题后，请在QQ群中根据自己对本章知识的掌握情况，进行投票，我将根据同学们的投票，及时了解同学们对本章的掌握情况，并做出教学调整。

< 返回

投票

发布投票

进行中



★李伦彬135552966

24秒前

完成本章作业后，认真复习和总结，你掌握本章知识了吗？

选项预览

共0票

- A. 已掌握，可独立完成作业。
- B. 基本掌握，参考课件和微视频，可完成作业...
- C. 没有掌握。

立即投票



练习题库-笔试

FSCapture

练习题库-上机

Dev-Cpp编程工具

自测系统

ZoomIt

Xmind-8



课件下载网址：

<http://www.jsjx.com/index.aspx>

[推荐资料1 - C语言经典编程282例](#)

[推荐资料2 - 华为C语言编程规范](#)

[推荐资料3 - 嗨翻C](#)

[推荐资料4 - C&C++ API.chm](#)



程序社团-技术交流群 : 540653133



首页 成员 设置 [编辑资料](#)


PGIMA
Programming Association

Programming Association
540653133

[发消息](#)

群介绍 本群创建于2017/10/17: 群主很懒,什么也没有留下

群标签 [行业交流](#) [IT/互联网](#) [程序设计](#) [IT](#) [计算机](#)

群主/管理员

成员分布 (136/200)

66%	13%	84%	16%
男-91人	黑河-19人	90后-115人	单身-23人

本群星级  该群一切正常



课件下载网址：

<http://jsjxy.hhhxy.cn/info/1074/2602.htm>

课件下载二维码：





感谢聆听

下一章内容：文件

主讲人：李伦彬

请同学们提前预习课本相关知识点

