（一）开头（称为预处理指令）

#include 不是必须的，但一旦程序中有该语句必须放在开始处。

stdio.h：standared input out 标准输入输出 h头文件

main()函数是C程序的起点，main()可以返回一个值，也可以不返回值，如无返回值前面要有void(void main)

\n=回车 换行。; 表语句结束。

void main() 返回空值 int main() 可设返回值

代码要成对敲，注意中英文符号。

注释：单行：以//开头，多行以/\*开头\*/结尾

（二）图框含义

圆角矩形：开始、结束

平行四边形：输入、输出

菱形： 判断

（三）编码规范

**排版规则**

1) 程序应采用缩进风格编写，每层缩进使用一个制表位（TAB），类定义、方法都应顶格书写；

2) 源程序建议使用英文书写，尽量不含有中文。每行不超过80字符。对于较长的语句（>80字符）要分成多行书写，长表达式要在低优先级操作符处划分新行，操作符放在新行之首，划分出的新行要进行适当的缩进，使排版整齐，语句可读；循环、判断等语句中若有较长的表达式或语句，则要进行适应的划分；

3) 左花括号要另起一行，不能跟在上一行的行末；

4) 一个变量定义占一行，一个语句占一行；

5) 对独立的程序块之间、变量说明之后必须加空行；

6) 在结构成员赋值等情况，等号对齐，最少留一个空格；

7) 若函数或过程中的参数较长，则要进行适当的划分。

8) 形参的排序风格：

Ø 最常使用的参数放在第一位；

Ø 输入参数列表应放在输出参数列表的左边；

Ø 将通用的参数放在特殊的参数的左边。

**2 命名规范**

**2.1 应用程序的命名**

“系统简称”+模块名称

**2.2 子模块的命名**

每个子模块的名字应该由描述模块功能的1-3以单词组成。每个单词的首字母应大写。在这些单词中可以使用一些较通用的缩写。

**2.3 变量的命名**

变量的命名的基本原则是使得变量的含义能够从名字中直接理解。可以用多个英文单词拼写而成，每个英文单词的首字母要大写，其中英文单词有缩写的可用缩写；变量的前缀表示该变量的类型；对于作用域跨越10行以上的变量名称不能少于4个字符，除循环变量，累加变量外不得使用I、j、k等名称的变量。变量分为取全局变量和局部变量，对于全局变量以加前缀“g\_”来区分。

使用有意义的英语单词,使用大小写分隔,每个单词的第一个字母为大写



考虑到习惯性和简洁性，对于按常规使用的局部变量允许采用极短的名字，如用n、i作为循环变量，p、q作为指针等。另外，要注意的是：全局变量在程序中不要定义太多，能用局部变量的就用局部变量。如果要使用相关的变量，建议采用类的方式或者结构的方式存放，以减少具体变量的个数。

**2.4 常量的命名**

常量所有的字母均为大写。并且单词之间使用下划线”\_”隔开。

**2.5 函数/过程的命名**

函数/过程名称应该尽量使用能够表达函数功能的英文名称，函数名称中应该禁止使用如同function1,function2等含义不清的名称。单词间应该使用大小写分隔。全局函数/过程名称以“g\_”前缀开始。

**2.6 接口命名**

接口名称要以大写字母开头。如果接口包含多个单词，每个单词的首字母大写，其他字母小写，如果，这些单词是缩略语（例如XML），也要首字母大写，其他字母小写(写为Xml)。

**2.7 类的命名**

类名称要以大写字母开头；

类名称如果包含多个单词，每个单词的首字母要大写，其他字母小写；如果这些单词是缩略语（例如XML），也要首字母大写,其他字母小写（写作Xml）；

类名称应该是一个名词或名词短语；

类成员变量的命名规则与上述规则相同，但是要以“m\_”开始，表示其为成员变量（Member）；

类名称不能出现下划线。

**2.8 方法的命名**

方法名称以小写字母开头。

方法名称如果包含多个单词，除了第一个单词外，每个单词的首字母大写，其它字

母小写。如果这些单词是缩略语（例如XML），也要首字母大写，其它字母小写（写作Xml）。

方法名称应该是一个动词或动名词短语，意思是“完成什么功能”，“执行什么操作”。

**2.9 数据库的命名**

**2.9.1 表：**

采用“模块名简称+前缀+’\_’+表名”的命名规则。表名以能理解该表的内容为原则，

可由中文表示，也可由代表此表含义的英文字母组成；首字母大写；前缀代表此表类别。

**2.9.2 视图：**

采用“模块名+’\_’+视图名+’视图’”的命名规则，通常由8个以内汉字组成。

**2.9.3 存储过程：**

采用“Proc+模块名+’\_’+存储过程名”的命名规则。

**2.9.4 触发器：**

采用“模块名+’’+触发类型+’’+表名”的命名规则，如果有多个触发类型，则可以叠加在一起。

**2.9.5 字段：**

字段的命名以能理解该字段的含义为原则，通常由多个英文单词加前缀拼写而成，而组成字段名称的首字母应大写。单词有缩写的可用缩写。字段的前缀表示该字段的数据类型，其取值详见“数据类型”描述。原则上，字段的命名长度不超过18字节；描述字段的中文名称，用数据库创建工具设计数据库时，需要输入。

**3 参数的约定**

**3.1 输入参数的约定**

有些函数有输入参数，这些参数指由函数外部（调用者）输入，并在函数内部使用。在函数业务流程说明后跟输入参数说明区，用“输入参数”或“Input Parameters”标记。在参数名列表中的每个参数后增加该参数的注释。

**3.2 输出参数的约定**

有些函数有输出参数，这些参数指由函数外部（调用者）定义，在函数内部使用并返回给调用者的参数。在输入参数说明区后跟输出参数说明区，用“输出参数”或 “Output Parameters”标记。在参数名列表中的每个参数后增加该参数的注释。另外输出参数一般以指针或应用输出。

**3.3 返回值的约定**

每个函数均有返回值，除非操作非常简单。对于有不同状态的返回值，建议用long型的返回值，0为成功。对于出错类返回值，在同一层次的模块，用统一代码表示。在输出参数说明区后跟返回值说明区，用“返回值”或“Return values”标记。返回值说明，要说明各种不同类型返回值以及它们的含义。

**4 注释约定**

在软件中对每个文件头,自定义函数和变量,重要的处理过程都要有必要的注释。

**4.1 源程序头的注释和规范**

每个源程序（包括存储过程）必须有头部说明、版本说明和函数结构三个部分：

源程序头部说明

FileName:

Copy Right:

System:

Module:

Function:

See also:

Author:

Create Date: 本程序的外部名字（如 \*.prg, \*.cpp）

xxx公司 版权所有 版本信息

本文件所在的系统或工程的名字

本文件所在的功能模块名称

简要说明本程序的功能

相关详细设计文档号

编码人员

创建日期

源程序版本说明

Editor：

Version：

Edit Date： 修改人员

版本号

修改日期

**4.2 函数头的注释和规范**

Name:

Function:

Input:

Output:

Return:

Syntax:

Env:

Calling: 函数名称

简述函数或过程的功能

[参数1] － [说明…]

[参数2] － [说明…]

[参数1] － [说明…]

[参数2] － [说明…]

[返回码1] － [说明…]

[返回码2] － [说明…]

调用语法(可选)

环境要求和影响(可选的)

被调用的函数(可选的)

**4.3 变量注释**

直接在变量后面注明变量的用途和取值约定，例如：

640.png

**4.4 类型定义注释**

指类和记录等等定义的注释。在注释中标明定义的用途。

**4.5 区的注释**

同一个类的成员方法要求排列在一起，共同协作而实现同一个功能的函数和过程要求排列在一起。代码通常使用几个函数和过程来实现某一项功能，这时候需要使用区注释将这些具有共同目的的函数和过程标明出来。

使用整行的”\*”作为隔离行，让程序清晰可读。

一般删除的代码不建议直接删除,最好用“//”注释起来。

**4.6 代码中的注释**

在代码中要求注释的地方有:

Ø 代码中的关键部分；

Ø 在使用特殊算法或者逻辑性较强的代码；

Ø 在修改或删除代码部分,需要加注释;修改/删除人,目的。

**5 变量的作用范围**

尽量做到缩小变量的作用范围，对于变量是指针的，应遵循以下约定：

Ø 在局部分配的空间在局部释放。

Ø 函数体内不能分配空间并将空间指针作为函数参数返回。

Ø 动态全局空间在程序结束时一定要释放。

Ø 所有动态分配的空间在对应层次的模块释放，并且用完马上释放。不重复释放相

同的指针。

**6 函数/过程的定义**

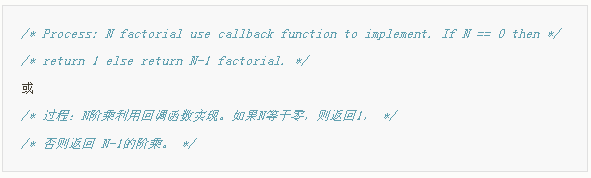
在函数的定义处应当增加本函数的功能描述的注释。用一句话描述清楚功能。可用英文或中文。功能注释格式要求所有代码一致。

**7 函数业务流程的定义**

在函数功能描述后，要增加函数的主要业务流程注释。可以用多行描述，以解释清楚业务流程为主。可用英文或中文。业务流程注释格式要求所有代码一致。

业务流程注释可以尽量详细，注释的长度可以与代码长度差不多，但是不要太长。

比如处理N阶乘的函数业务流程定义如下：

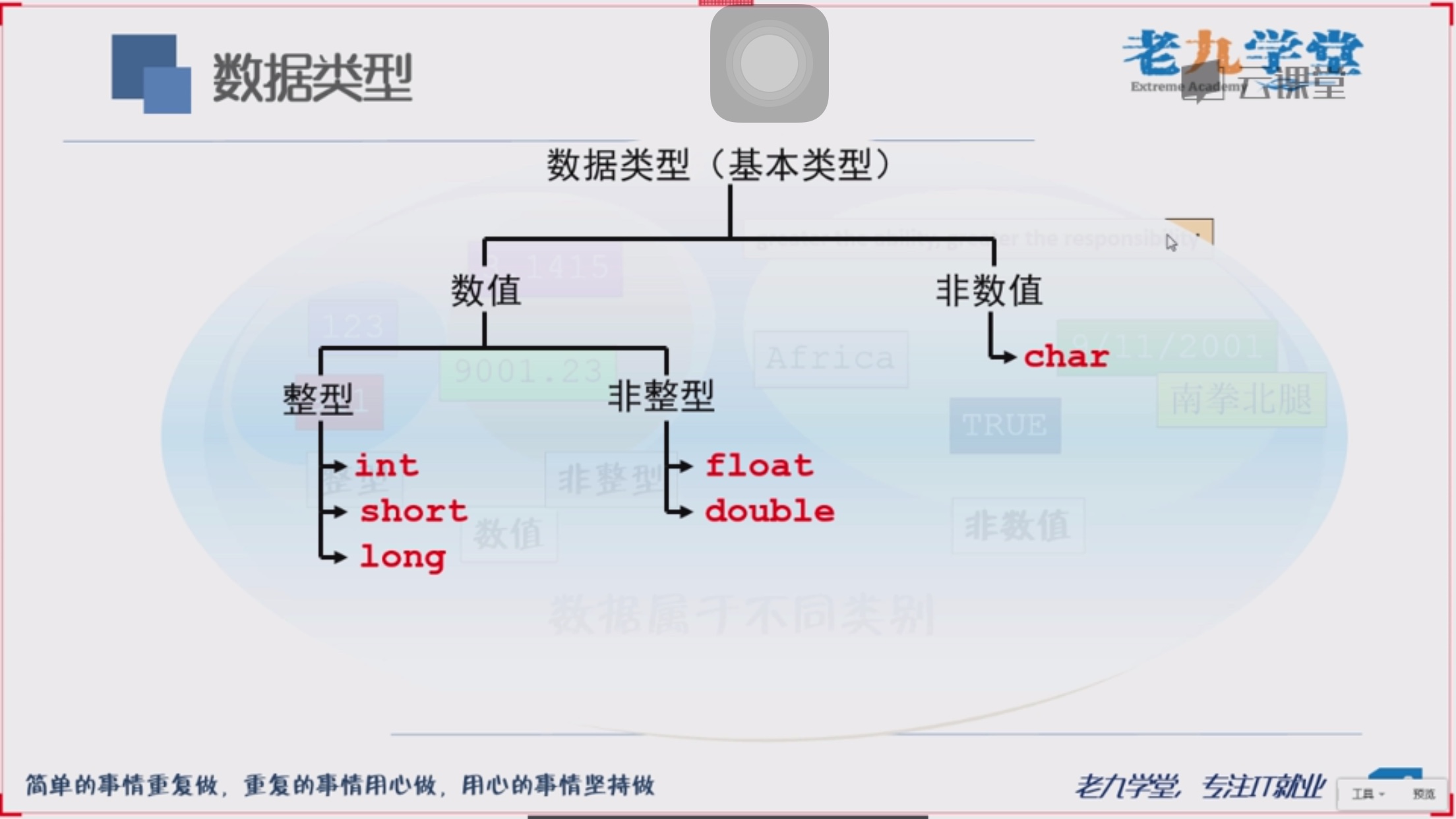


（四） 变量命名及数据类型

c只能由字母及下划线，且第一个字符必须为字母或下划线，不能包含除\_以外的任何特殊字符如：%#，空格

变量就是一块内存空间

数据类型





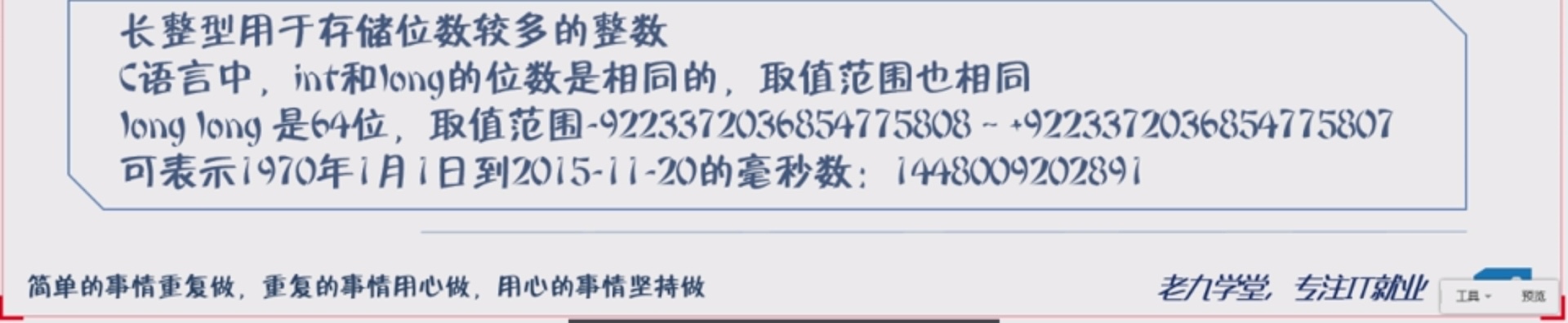
整型

用于存储整数，客户编号(9527)学生人数(25000)一天秒数(864000)

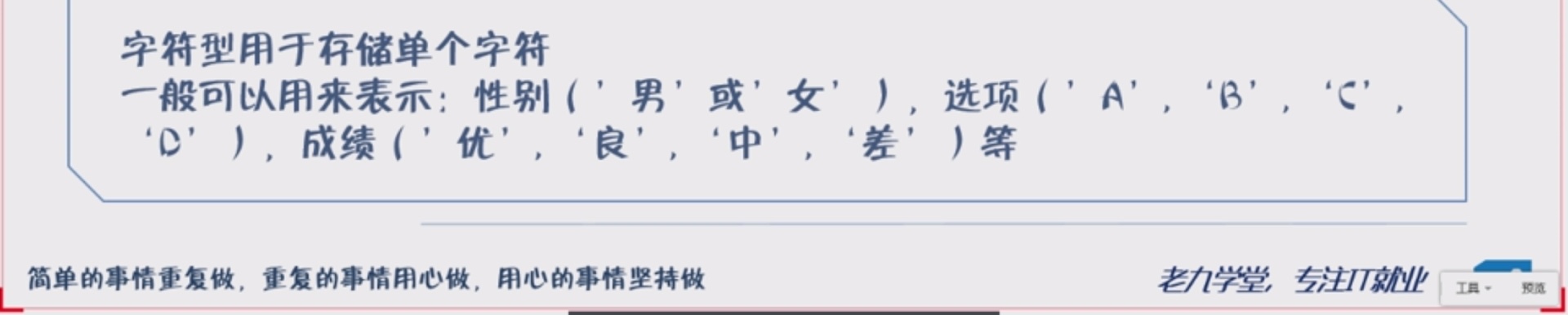
短整型

存储位数较少的整数，年龄，一年天数

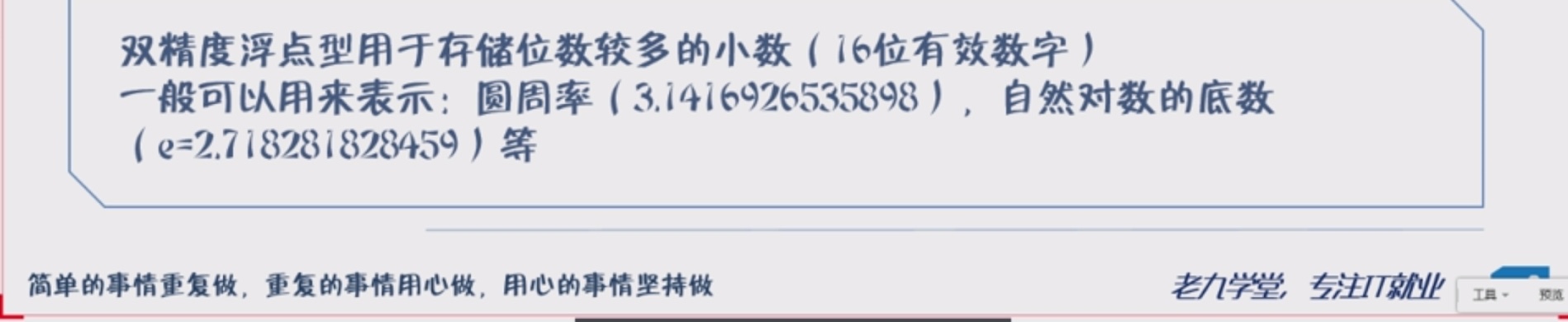
长整型



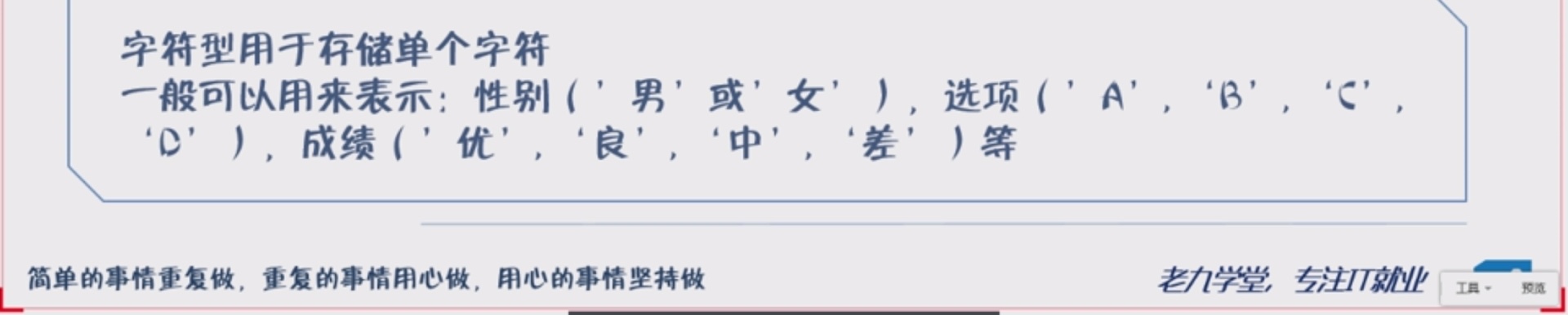
单精度



双精度



字符型

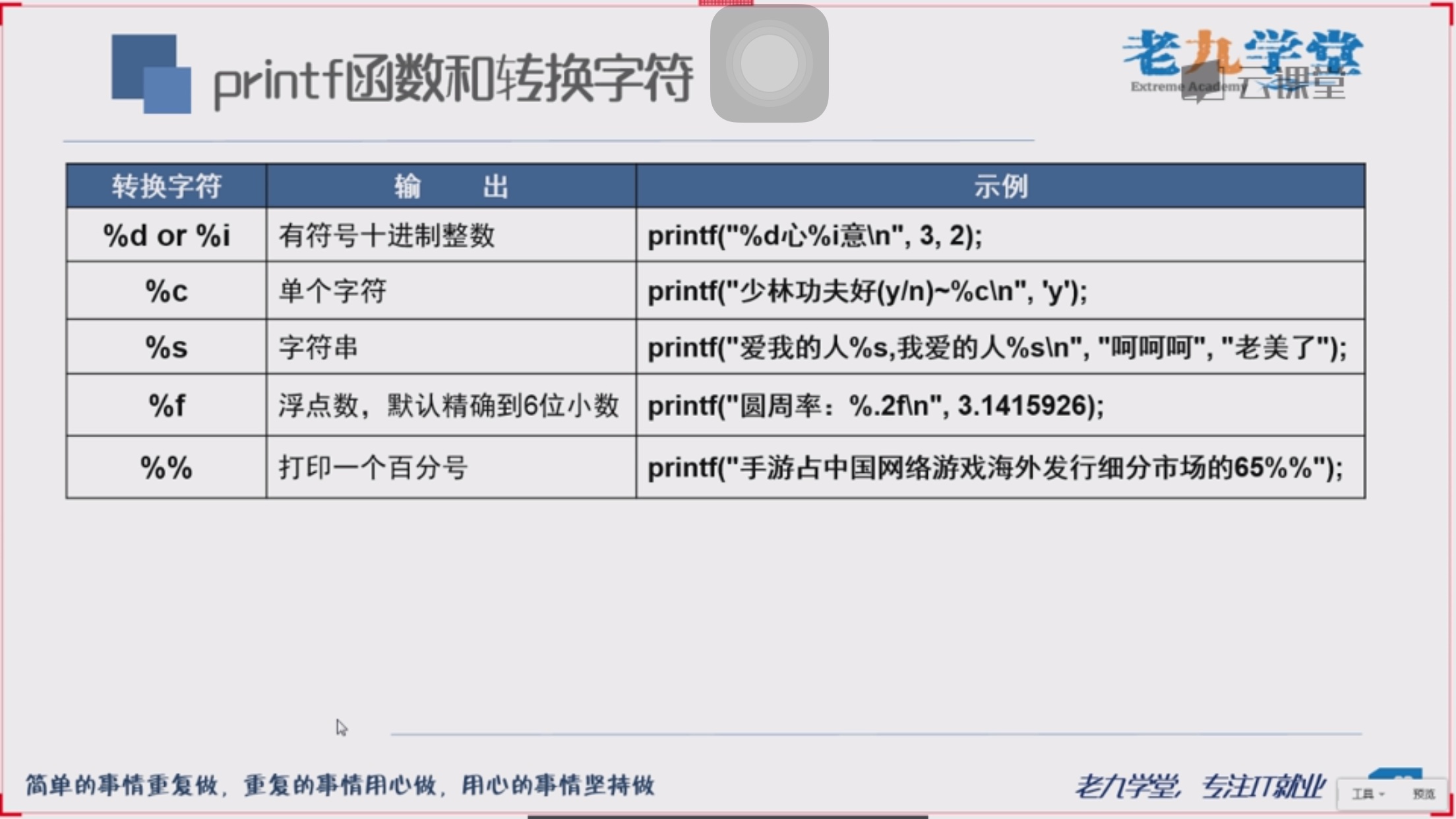


一条语句可以声明多个不同类型的变量

变量名不要重复

float小数点后有7位有效数字eg：float = 40.0f。double 16位不用加f，加L就变为long double

（五） 输出函数和转换字符



转义字符



蜂鸣有的系统无用，



自动类型转换

