编译：高级语言翻译为机器码——编译原理、汇编（微机原理）；先知道按键

缺少课程支撑，课时远远不够（30学时对别人64学时），各种课都是压半学完

编辑就是打字（vs2012的编辑功能等）

编译为机器码是为了CPU

调试：反复调整到需要的结果，慢慢熟练的结果

第十周机考！

历年不及格率低于3%，坚持认真学就能会，不要放弃

自己电话/邮件找老师/助教（5个，在办公室隔壁，随时找，罗姆楼5-104/108，耐心）

老师需要预约，助教晚上基本都在实验室

算法自己思考；操作问助教，当面手把手！

变量定义：类型标识符 变量名列表；

变量的初始化：赋初值

定义变量（变量名）就是在内存中开一个空间（地址）

访问变量：通过变量名（直接访问）或变量地址（间接访问——指针）

变量大小决定空间开的大小

最好的取名规则（良好的素养但不强求）——匈牙利命名法： int iCount（i表示整型）

小写字母表示类型，完整单词表示内容

用语义取名！

标识符：

float：单精度实

double：双精度实

数值、类型相一致

变量没有新的数据写入，就总保存这个数据（一次写、多次读）

存储空间（对应变量）=大小+地址

C语言用来作控制语句使用的单词是关键字、预处理命令，不能用来作变量名（熟了就知道）

整型4种：基本型、短整型、长整型、无符号型（包含三种）；与环境相关（sizeof测试）

调试可以把语句注释掉

没有赋初值就是随机值，会报错（但是报错位置不一定是那一行）

printf：普通字符原样输出，控制字符控制变量输出格式

预编译#define N 1001 给出符号常量

溢出会把后面的字节放进去，前面的不管了（会有错误结果）

#include<limits.h>

printf(\t)输出制表符（tab）\n换行（enter）

SCHAR\_MIN：signed char的最小数

SCHAR\_MAX：signed char 的最大数

INT\_MIN; INT\_MAX ULONG\_MIN之类的一类，都是已经定义的变量了（limits.h）

记住！

数据类型决定编码方式（ASCII、补码、偏移码……）

数据类型决定空间大小（2、4、8……）

**第三讲：数据类型（中）**

实型变量：单精度float 4byte 双精度double 8byte（和长双精度）

偏移码表示（符号、尾数、指数（含符号））

存储空间范围表示**精度**和**大小**范围

float 7位有效数字；double 15位左右有效数字

等于号是赋值，以左值类型为准，右边的数会转化为左边的类型

按照错误列表报错提示改程序即可，一遍很难不出错

实型常量默认double，所以float a=889.8会出警告（“截断”）

报错：警告（可运行，随便改不改）、错误（不可运行，必须改）

精度问题（存储误差）导致实型数比较出问题

单精度实型数比较大小的正确方式：

float f=123.456;

fabs(f-123.456)<1e-5

这样才能认为相等

双精度就没问题（没有截断的问题）

原因：十进制有限小数可能在二进制中是无限小数

字符型：char （内存1字节，存放ASCII）

布尔型：bool （true=1，false=0，内存1字节/1位）；输入非0值都认为是true

字符型存入二进制和十进制整数就没有差别了

在计算机内部各个类型只有长度的差别

符号常量：（代码改数方便，增加语义）

#define PRICE 100

预编译阶段符号常量的替换

常变量：只读变量（符号常量但是有数据类型，保证安全性）

const 类型 变量名;

定义了就不能再改了（会报错）

有些代码多写了（废代码）不影响运行

常量：对变量赋值或初始化（数值、字符、符号），

前面注 0x/后面注L/LU之类的改变类型/数据，不然就是十进制默认类型

整型默认int

0.=.0都是double，实型默认double，注f/L；E表示10的幂（大小写不影响）

字符常量：加单引号，和字符串常量不一样（双引号）！单撇或反斜杠是不合法的

转义字符：表示特殊控制字符（用一个字符表示一个意思）

\n换行 \t横向跳格 \f打印机走纸换行 [\\反斜杠](file:///\\反斜杠) \v竖向跳格 \’单引号 \b退格

\r回车（先回车还是先换行每个系统还不一样）

ASCII表示：\ddd三位八进制（替换d） \xhh二位16进制（替换h）

‘\101’表示’A’

常量直接赋值：c=’a’;即可，用默认的类型

字符型数据和整型可以通用