

第3讲 数据类型（中）

--变量、常量、运算符、表达式

黄永峰

2022-9-30

yfhuang@tsinghua.edu.cn

□ 对应教材的第2、4章内容

本讲内容

- 数据类型及其表示方法
- 运算符和表达式概念






2.3.3 实型变量

- ① 实型变量分为单精度（float）和双精度（double）
- ② float内存4个字节； double占内存8个字节
- ③ float型变量只能接受7位有效数字； double型变量只能接受15位有效数字

```

1  #define N 100
2  #include<stdio.h>
3  int main()
4  {
5      float fPayRate = 889.8;
6      unsigned int uMaxItems = 12;
7      double dPayYear = 0;
8      dPayYear = fPayRate * uMaxItems;
9      printf("dPayYear=%f", dPayYear);
10     printf("total_L=%f", total_L);
11     return 0;
12 }
```

错误列表

<div> <div>整个解决方案</div> <div>错误 2</div> <div>警告 3</div> <div>消息 0</div> <div>生成 + IntelliSense</div> </div>			
	代码	说明	项目
	E0020	未定义标识符 "total_L"	Project1
	C26451	算术溢出: 使用 4 字节值上的运算符 *, 然后将结果转换到 8 字节值。在调用运算符 * 之前将值强制转换为宽类型可避免溢出(io.2)。	Project1
	C2065	"total_L": 未声明的标识符	Project1
	C4477	"printf": 格式字符串"%f"需要类型"double"的参数, 但可变参数 1 拥有了类型"int"	Project1
	C4305	"初始化": 从"double"到"float"截断	Project1

2.3.3 实型变量

```

1  #define N 100
2  #include<stdio.h>
3  int main()
4  {
5      float fPayRate = 889.8;
6      unsigned int uMaxItems = 12;
7      double dPayYear = 0;
8      dPayYear = fPayRate * uMaxItems;
9      printf("dPayYear=%.1f", dPayYear);
10     //printf("total_L=%f", total_L);
11
12     return 0;
13 }
```

Microsoft Visual Studio 调试控制台

dPayYear=10677.6

D:\My_New_UserFiles\New_Desktop\test\
按任意键关闭此窗口. . .

2.3.3 实型变量的陷阱

```

1  #define N 100
2  #include <stdio.h>
3  int main()
4  {
5      float f = 123.456;
6      if (f == 123.456)
7          printf("Yes, f is equal to 123.456 indeed.");
8      else
9          printf("No, f is equal to %\n");
10     return 0;
11 }

```

Microsoft Visual Studio 调试控制台

No, f is equal to

D:\My_New_UserFiles\New_Desktop\test\Project1\x64\Debu
按任意键关闭此窗口. . .

```

1  #define N 100
2  #include <stdio.h>
3  int main()
4  {
5      double f = 123.456;
6      if (f == 123.456)
7          printf("Yes, f is equal to 123.456 indeed.");
8      else
9          printf("No, f is equal to 123.456");
10     return 0;
11 }

```

Microsoft Visual Studio 调试控制台

Yes, f is equal to 123.456 indeed.
D:\My_New_UserFiles\New_Desktop\test\Project1\x64\Debu
按任意键关闭此窗口. . .

```

1  #define N 100
2  #include <stdio.h>
3  #include <math.h>
4  int main()
5  {
6      float f = 123.456;
7      if (fabs (f- 123.456)<1e-5)
8          printf("Yes, f is equal to 123.456 indeed.");
9      else
10         printf("No, f is not equal to 123.456");
11     return 0;
12 }

```

Microsoft Visual Studio 调试控制台

Yes, f is equal to 123.456 indeed.
D:\My_New_UserFiles\New_Desktop\test\Project1\x64\Debu
按任意键关闭此窗口. . .

结论:

- ① f和123.456有类型差异
- ② 不同类型实型数有精度差异
- ③ 实型数之间正确的比较方法

有关浮点数的精度问题

123.456000的二进制，小数位为无穷位.....

0	0.456	1	.552	1	.984
1	<u>0.912</u>	1	<u>1.04</u>	1	<u>.968</u>
1	<u>1.824</u>	0	<u>0.208</u>	1	<u>.936</u>
1	<u>.648</u>	0	<u>0.416</u>	1	<u>.872</u>
0	<u>.296</u>	0	<u>0.832</u>	1	<u>.744</u>
1	<u>0.592</u>	1	<u>.664</u>	1	<u>.488</u>
0	<u>.184</u>	0	<u>.328</u>	0	<u>0.976</u>
0	<u>0.368</u>	1	<u>0.656</u>	1	<u>.952</u>
0	<u>0.736</u>	0	<u>.312</u>	1	<u>.904</u>
1	<u>.472</u>	0	<u>0.624</u>	1	<u>.808</u>
0	<u>0.944</u>	1	<u>.248</u>	1	<u>.616</u>
1	<u>.888</u>	0	<u>0.496</u>	1	<u>.232</u>
1	<u>776</u>	0	<u>0.992</u>		

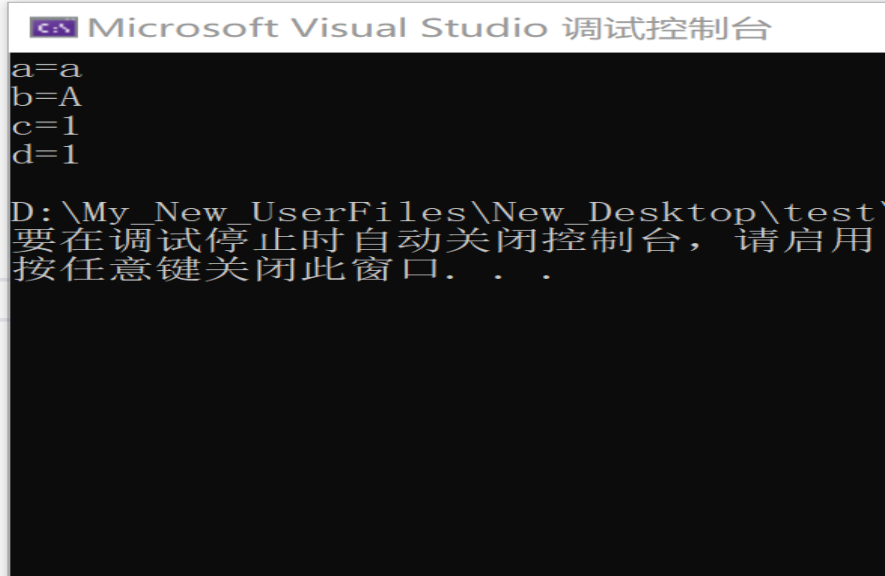
2.3.4 字符和布尔性变量

- 字符和布尔变量在内存中占一个字节；字符变量存放字符的ASCII码值；布尔变量存储1或0；
- 字符变量定义和初始化 `char x1, ch1='x' ;`
- 布尔变量定义和初始化 `bool c1, c2=true; c3=1;`

布尔型常量	实际取值
true	1
false	0

```

1  #define N 100
2  #include <stdio.h>
3  #include <stdbool.h>
4  int main()
5  {
6      char a = 'a';
7      char b = a - 32;
8      bool c = true;
9      bool d = 'a';
10     printf("a=%c\n", a);
11     printf("b=%c\n", b);
12     printf("c=%d\n", c);
13     printf("d=%d\n", d);
14     return 0;
15 }
```



Microsoft Visual Studio 调试控制台

```

a=a
b=A
c=1
d=1

D:\My_New_UserFiles\New_Desktop\test
要在调试停止时自动关闭控制台，请启用
按任意键关闭此窗口. . .
```

2.3.5 常变量

- ❑ 常变量：又称为只读变量（Read-only-variable）
- ❑ 是保护数据安全的机制，防止意外修改
- ❑ 定义形式：const 类型 变量名；例如：const int a=3; const float b=3.14*2;

```

1  #define N 100
2  #include <stdio.h>
3  #include <stdbool.h>
4  int main()
5  {
6      char a = 'a';
7      const char b = 0;
8      a = a - 32;
9      b = a - 32;
10     printf("a=%c\n", a);
11     printf("b=%c\n", b);
12     return 0;
13 }

```




错误列表

整个解决方案

❌ 错误 2

⚠ 警告 1

📄 消息 0

	代码	说明	项目
	E0137	表达式必须是可修改的左值	Project1
	C26451	算术溢出: 使用 4 字节值上的运算符 *, 然后将结果转换到 8 字节值。在调用运算符 * 之前将值强制转换为宽类型可避免溢出(io.2)。	Project1
	C2166	左值指定 const 对象	Project1

2.4 程序中的常量表示

- ▶ **常量**: 在编程时用来对变量赋值或初始化, 或直接参加表达式运算。例如, `int a, a=3; char c= 'a' ; a=a+4;`
- ▶ **类型**: 数值常量(整型常量, 实型常量)、字符常量和符号常量

常量表达		实际取值	常量类型
整型常量 十进制:	+123	123	int
	-378	-378	int
	-32271L	-32271	long int
	76542LU	76542	unsigned long int

常量表达		实际取值	常量类型
整型常量(十六进制)	0x20	32	int
	0xa0	160	int
	0XFF02	65282	int

2.4 常量表示—实型数的表示

□问：为什么程序员总是分不清万圣节和圣诞节？

□ 答：因为 Oct 31 == Dec 25！

	常量表达	实际取值	常量类型
浮点型常量 (小数方式)	0.	0.0	double
	.0	0.0	double
	2.0	2.0	double
	3.1416f	3.1416	float
	3.1415926536L	3.1415926536	long double
浮点型常量 (指数形式)	.314159E1	3.14159	double
	31.4159E-1	3.14159	double
	314.159E-2	3.14159	double

□ 注意：E表示的意义

2.4 程序中的常量表示-- 字符型常量的表示P36

(1) 单撇号表示法：用一对单撇号括起来的一个字符，如 ‘a’，‘c’，‘A’，‘?’ 等单撇号中的单个字符。但不能是单撇号或反斜杠：即 ‘ ’、‘\’ 都是不合法的

字符常量表达	实际取值
'A'	65
'<'	60
'/'	44
'9'	57
'+'	43
'?'	63
'z'	122
' '	32

2.4 程序中的常量表示-- 字符型常量的表示P36

(2) 转义字符表示法：用斜杠加一个字符或16进制数表示

字符形式	含 义	字符形式	含 义
\n	换行	\f	走纸换行
\t	横向跳格(跳到下一个输出区域)	\\	反斜杠字符\
\v	竖向跳格	\'	单引号 '
\b	退格	\ddd	3位八进制数所代表的字符
\r	回车	\xhh	2位十六进制数所代表的字符

□ 利用\ddd转义字符，可以表示任何ASCII码字符

□ 问题： '\101' == 'A' 吗？为什么呢



```

1  #define N 100
2  #include <stdio.h>
3  int main()
4  {
5      int i, j;
6      char c1, c2, c3;
7      i = 'A';
8      j = 'a';
9      c3 = j - 32;
10     c1 = '\101';
11     c2 = '\t';
12     printf("i=%d", i);
13     printf("%c", c2);
14     printf("j=%d,c3=%c\n", j, c3);
15     printf("c1=%c\n", c1);
16     return 0;
17 }

```

Microsoft Visual Studio 调

```

i=65      j=97, c3=A
c1=A

D:\My_New_UserFiles\New_
按任意键关闭此窗口. . .

```

提问：ch1=“x” 对吗？为啥呢

□ 结论：字符型数据和整型数据可以通用

2.4 程序中的常量表示-- 符号常量

```

1  #define PRICE 30
2  #include <stdio.h>
3  int main()
4  {
5      int num, total1, total2, total3;
6      num = 10;
7      total1 = num * PRICE;
8      total1 = (num) * PRICE;
9      total2 = (num + 1) * PRICE;
10     total3 = (num + 2) * PRICE;
11     printf("total1=%d\n", total1);
12     printf("total2=%d\n", total2);
13     printf("total3=%d\n", total3);
14     return 0;
15 }
```

Microsoft Visual Studio 调试窗口

```
total1=300
total2=330
total3=360
```

D:\My_New_UserFiles\New_D
按任意键关闭此窗口. . .

□ 提问：引入符号常量有什么好处？

阅读下列 C 程序： ↵

```
#include <stdio.h>↵
```

```
main()↵
```

```
{ long x, y, z;↵
```

```
    unsigned short y1;↵
```

```
    x=-0xffffL; y=-0xffL; z=-0xffffffffL;↵
```

```
    y1=1;↵
```

```
    printf("x=%6ld    y=%6ld    z=%6ld\n", x, y, z);↵
```

```
    printf("y=%u\n", y1);↵
```

```
}↵
```

该程序运行后，输出的结果为： ↵

```
x=-65535    y=-255    z=    1↵
```

```
|y1=1↵
```

□ 提问：常变量与符号常量有区别吗？



符号常量



常变量

- 符号常量在编译时被替换掉，不分配存储空间
- 常变量编译时分配存储空间，并检测类型，因此更安全
- 常变量是C++对C的改进，因此更好用

2.4 程序中的常量表示

-- 字符串常量的表示p36



- 字符串常量： 用双撇号括起来的多个字符序列, 如：“zhang san ”、“student” ， “x” ， ”hello” 等，字符串的字符数为字符串长度
- 系统在每一个字符串的末尾都自动加一个“字符串结束标志”，即 ‘\0’

字符串常量	实际显示
""	无内容，空串
"h"	h
"Hello World!\n"	Hello World!
"HOW ARE YOU?"	HOW ARE YOU?
"Good Morning!"	Good Morning!
"' Good' Morning!"	'Good' Morning!
"\"Good\" Morning!"	"Good" Morning!

问题： “x”= ‘x’吗，为什么呢？

H	e	l	l	o	\n	
72	101	108	108	111	10	0

```
1  #include<stdio.h>
2  int main()
3  {
4      int x;
5      char y;
6      x = 65;
7      y = 'B';
8      printf("x=%c\n", x);
9      printf("y=%c\n", y);
10     printf("y=%d\n", y);
11     return 0;
12 }
```

Microsoft Visual Studio 调试控制台

```
x=A
y=B
y=66
```

```
D:\My_New_UserFiles\New_Desktop\test\Pr
按任意键关闭此窗口. . .
```

2.4 程序中的常量表示-- 字符串常量的表示

```

1  #include<stdio.h>
2  int main()
3  {
4      printf("中国人民万岁\n");
5      printf("中国人民万岁\n");
6      printf("how are you\n");
7      printf("how\t are\t you\n");
8      printf("how\t\t are\t\t you\n");
9      printf("abc\n");
10     printf("abc\n");
11     printf("abc\n");
12     return 0;
13 }
```

Microsoft Visual Studio 调试控制台

```

中国人民万岁
中国人民万岁\nhow are you
how      are      you
how\tare\t you
abc
abc\nabc\
```

D:\My_New_UserFiles\New_Desktop\tes
要在调试停止时自动关闭控制台，请启用
按任意键关闭此窗口. . .

2.5 运算符和表达式p63

- **运算符**: C++丰富运算符(共44个); 按功能可分为13类

算术运算符	+, -, *, /, %
关系运算符	>, <, ==, >=, <=, !=
逻辑运算符	!, &&,
位运算符	<<, >>, ~, , ^, &
赋值运算符	=
条件运算符	? :
逗号运算符	,
指针运算符	*, &
求字节数运算符	Sizeof, 例:sizeof(int)或sizeof(变量)
强制类型转换运算符	(类型)
分量运算符	.(点)、→
下标运算符	[]
其他	函数调用运算符()

2.5 运算符和表达式p63

● 表达式：用运算符将常量、变量、函数连接起来构成的计算式

● 注意

(1) 参加表达式运算的量必须有确切的值，表达式经过运算后，也一定会得到一个确切的值。

(2) 计算表达式的值要注意三个方面问题

- a、运算符的结合性
- b、运算符的优先等级
- c、数据类型的转换

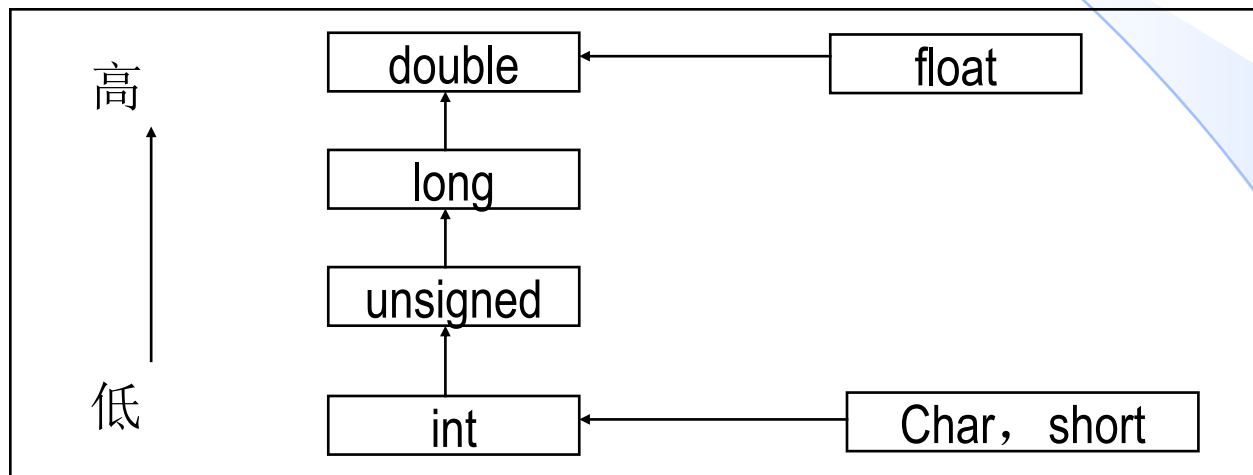


□ 请问：何为C++语句？如：`a=3;`

2.5 运算符和表达式 --类型转换

(1) 不同类型的混合运算的自动转换规则

- ◆ 不同的数据类型要先转换成同一类型后，才可运算
- ◆ 自动转换的特点：临时性和逐步性



(2) 为程序清晰, 建议使用强制转化, 如: (int) a;

- ◆ 强转很灵活, 你必须知道你在做什么!!!
- ◆ 经常用强转来解决很多warning.

2.5 运算符和表达式 --类型转换实例

```

1  #include<stdio.h>
2  int main()
3  {
4      int a = 3;
5      float b = 3.14;
6      char c = 'a';
7      printf("a=%d\r", a);
8      printf("b=%f\n", b);
9      a = a + b;
10     printf("a=%d\n", a);
11     a = a + b + c;
12     printf("a=%d\n", a);
13     return 0;
14 }
  
```

选择Microsoft Visual Studio 调试控制台

b=3.140000

a=6

a=106

D:\My_New_UserFiles\New_Desktop\test
按任意键关闭此窗口. . .

本讲重点小结

- 变量定义的含义
- 常量的表示
- 各种变量定义方法.
- 运算符优先级、结合性
- 表达式计算时类型转换

另外，教材的2.1节可以不作重点看

下一讲主要内容

- 运算符
- C输出和输入
- 参考教材的第2章，希望大家做好预习

第2次 实验练习

1. **本次实验练习必做题共3题**，要求上机完成。在第5周末之前提交。提交内容：代码和执行结果

2. 题目：

第1题：参考上次作业，使用运算符sizeof()来测试你电脑（如VS2012）编译器环境下short , int, long, unsigned short, unsigned int, unsigned long, char, Bool, double和float等类型的字节数；同时，设计一种方法测试float类型能保留的小数位数长度。将上述结果在屏幕打印出来。

第2题：参考和修改下列代码：要求通过键盘动态输入不同整数值给a和b；输出与参考代码不一样特殊图案。说明：ASCII 字符共有256个，不同字符，图形不一样。参考代码程序源代码：

```
#include "stdio.h"
main()
{
    char a=176,b=219;
    printf( "%c%c%c%c%c\n" , b, a, a, a, b) ;
    printf( "%c%c%c%c%c\n" , a, b, a, b, a) ;
    printf( "%c%c%c%c%c\n" , a, a, b, a, a) ;
    printf( "%c%c%c%c%c\n" , a, b, a, b, a) ;
    printf( "%c%c%c%c%c\n" , b, a, a, a, b) ;}
```

第3题：【问题描述】某农场主要牧农在6工月内交税费10万元，否则就被开除。可怜牧农家里穷得叮当响，除了两只小羊外，一无所有。他看着家里仅有的两只羊道：“羊呀羊，你要是能每个月为咱家生一对羊该多好啊！这样我就能拿你们去卖钱！半年后我或许就可缴纳税费。大家编程帮牧农算算：6个月是否可筹齐税费？编程打印每个月的“月份、羊数量和钱”。

【假设】牧农家的两只羊是2018年1月1号生的，要在下个月的才能发育成熟生子；每一对发育成熟的羊在接下来的每个月都能生出一对小羊（一雌一雄）；羊永不死亡；每只羊买10元。在上述假设下，6个月后，牧农家会有多少只羊？能买多少钱？

【提示】结合高中数学（繁殖模型允许不同），梳理出第1至6月份的羊数量的计算公式

选做题

『题目要求』如果你保存你同桌的手机号码，请同学帮忙设计一个存储11位手机号码软件。主要考虑用什么数据类型？试考虑多种不同方案。并根据所设定的数据类型，声明变量来存储手机号码，然后使用该变量类型在屏幕上正确地输出号码。如：存储的手机号码是13520865205，则最后输出结果应该是：13520865205

上帝是个程序员

问：上帝真能控制生活中发生的每一个细节吗？

答：如果他使用调试程序（Debugger）的话就可以，但一步步跟踪每一个变量肯定是件很烦的事，而且你也会控告他侵犯隐私。

问：为什么上帝会允许罪恶存在？

答：他原来以为在上一个版本中已经去掉所有的罪恶Bug。

问：上帝真的无所不知吗？

答：是的，他经常这么想，但也经常被自己昨天写的程序吓一跳。

问：上帝真的在七天里创建了整个世界吗？

答：他在头六天里没日没夜的工作，仅靠可口可乐和糖块过活，第七天等他回到家时发现已经被女朋友抛弃了。

问：我死后去哪儿？

答：备份磁盘。

问：有人声称听见了上帝的声音，是真的吗？

答：其实他只是收到了E-mail而已。