**程序设计教学大纲**

第1节

2017年11月26日星期日

热身题：

有三个物体，圆柱体、圆锥体、球体，它们的上俯视图都是同一个圆，请编写一个程序，要求输入投影圆半径r和高h，求圆柱体的表面积和体积，圆锥的体积与球体的体积。

1. 数据类型与变量、常量的定义

1.数据类型

1. 整型int

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据类型** | **定义标识符** | **占字节数** | **数值范围** | **数值范围** |
| 短整型 | short [int] | 2(16位) | -32768～32767 | -215～215-1 |
| 整型 | [long] int | 4(32位) | -2147483648～2147483647 | -231～231-1 |
| 长整型 | long [int] | 4(32位) | -2147483648～2147483647 | -231～231-1 |
| 超长整型 | long long [int] | 8(64位) | -9223372036854775808～9223372036854775807 | -263～263-1 |
| 无符号整型 | unsigned [int] | 2(16位) | 0～65535 | 0～216-1 |
| 无符号短整型 | unsigned short [int] | 2(16位) | 0～65535 | 0～216-1 |
| 无符号长整型 | unsigned long [int] | 4(32位) | 0～4294967295 | 0～232-1 |
| 无符号超长整型 | unsigned long long | 8(64位) | 0～18446744073709551615 | 0～264-1 |

1. 浮点类型float double

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据类型** | **定义标识符** | **数值范围** | **占字节数** | **有效位数** |
| 单精度实型 | float | -3.4E-38～3.4E+38 | 4(32位) | 6～7位 |
| 双精度实型 | double | -1.7E+308～1.7E+308 | 8(64位) | 15～16位 |
| 长双精度实型 | long double | -3.4E+4932～1.1E+4932 | 16(128位) | 18～19位 |
| 布尔变量 | bool | 真true或假false之一 | 1(8位) |  |

1. 字符类型char string

字符常量有以下两种表示法：

1）普通表示形式

　　字符常量是由单个字符组成，所有字符采用ASCII编码，ASCII编码共有128个字符(表2-4)。在程序中，通常用一对单引号将单个字符括起来表示一个字符常量。如：'a'，'A'，'0'等。如字符A的序号是65，字符a的序号是97, 字符0的序号的48。

2）转义字符表示形式。

　　转义字符有三种用法：表示控制字符、表示特殊字符、表示所有字符。常用的转义字符如右表所示。

|  |  |
| --- | --- |
| **转义字符** | **含义** |
| '\n' | 换行 |
| '\t' | 水平制表 |
| '\b' | 退格 |
| '\r' | 回车（不换行） |
| '\0' | 空字符 |
| '\’' | 单引号 |
| '\”' | 双引号 |
| '\\' | 一个反斜杠字符 |
| '\ddd' | 1～3位八进制数所代表的字符 |
| '\xhh' | 1～2位十六进制数所代表的字符 |

\*ASCII码表见第1节教学大纲

1. 布尔类型bool
2. 数组类型
3. 结构体类型
4. 指针类型

2.**常量**是指在程序中使用的一些具体的数、字符。在程序运行过程中，其值不能被更改。如123,145.88,’m’,TRUE等。

（1）整型常量：如3、-5、0等。

　　 整型常量是表示整数的常量。有三种表示形式：

　　 1）十进制形式。如99、-1。

　　 2）八进制形式。以数字0打头，由0-7构成，如012，

表示八进制整数12，即12(8)。

　　 3）十六进制形式。以0x打头，如0x12A，表示十六进

制整数12A，即12A(16)。

（2）实型常量：如3.1、-6.1E+2（科学记数法）。

　　 （3）字符常量：用单引号括起来的字符，如 ‘k’ 、’5’ 、’%’。

\*注意：’a’与”a”表示的含义是不同的，’a’表示一个字符常量，”a”表示一个字符串。一个常量可以直接调用（如124,’A’），也可以给常量取个名字用一个标识符代表它，这就是符号常量。其语法格式为：

const 符号常量=常量字串;

例如：const double PI=3.1415926;

3.**变量**代表了一个存储单元，其中的值是可以改变的，因此称为变量。如游戏中玩家命的条数最初为3，当你死了一次，命减少一条，这里命的条数就是一个变量（或者说命的条数存储在一个存储单元中）。

一个程序中可能要使用到若干个变量，为了区别不同的变量，必须给每个变量（存储单元）取一个名（称为变量名），该变量（存贮单元）存储的值称为变量的值，变量中能够存储值的类型为变量的类型。例如游戏中用于存储“命”的变量，在游戏程序中的存储命的变量名可取为life，它的类型为整型，游戏初始时这个变量的值为3。

\*注意：变量名应遵循自定义标识符的命名规则，并建议使用“见名知义”的原则，即用一些有意义的单词作为变量名。在C++语言，变量名大小写有区别。建议变量名的长度不要超过8个字符。例如：month、\_age、s2为合法的标识符；m.k.jack、a<=b、9y为不合法的标识符。

定义变量的语法格式为：

　　　　　数据类型 变量表

　 　例如：int i=5,j,k; //定义I,j,k为整型变量，

//i赋初值为5,j,k的初值未知。

　　 Char a,b,c; //定义a,b,c为字符变量

　　 float x,y,z; //定义x,y,z为实型变量

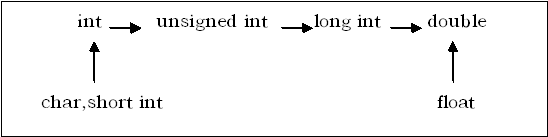
　　 C++语言允许在定义变量的同时为变量赋初值。

4.变量类型转换

C++语言中，不同数据类型的运算对象进行混合运算，或者需要将一个表达式的结果转换成期望的类型时，就需要依据数据类型转换规则进行转换。

（1）混合运算时的类型转换规则

整型、实型、字符型数据间可以混合运算。在这种情况下，需要将不一致的数据类型转换成一致的数据类型，然后进行运算。为了保证运算精度，系统在运算时的转换规则是将存储长度较短的运算对象转成存储长度较长的类型，然后再进行处理。这种转换是系统自动进行的，具体见表2-5所示。



（2）当赋值运算符两侧的数据类型不同时，需进行类型转换，这种转换是系统自动进行的，转换规则如下：

　　1）float、double型赋值给int型：直接截断小数。

　　例如，“int i=f+0.6;”，f的值为4.0，右边算术表达式运算后的结果为4.6的double型数据，根据上述转换原则，直接舍弃小数，所以i的值为4。

　　2)int、char型赋值给float、double型：补足有效位以进行数据类型转换。

　　例如：“float f=4;”，float为7位有效数字，所以f的值为4.0000000。

　　3）char型(1字节)赋给int型(4字节)：数值赋给int型的低8位，其它位补0。

　　4）long int型赋值给int型：long int型截断低字节给int型。

　　5）int型赋值给long int型：赋给long int型的低16位，如果int型的最高位是0，则long int的高16位全为0；如果int型的最高位是1，则long int型的高16位全为1（称为“符号扩展”）。

　　6）unsigned int型赋值给int型：直接传送数值。

　　7）非unsigned int型赋值给位数相同的unsigned int型：直接传送数值。

1. 运算符与运算符的优先级
2. 算术运算符——用于各类数值运算。包括加(+)、减(-)、乘(\*)、除(/)、求余(或称模运算，%)、自增(++)、自减(--)共七种。

5%2=1

求余的运算符“%”也称为模运算符，是双目运算符，两个操作数都是整型数。A%b的值就是a除以b的余数，5%2余数为1。其操作对象只能是整型数，而其他四种运算符对int,float,double,char都适用。

自增、自减运算符用来对一个操作数进行加1或减1运算，其结果仍然赋予该操作数，而且参加运算的操作数必须是变量，而不能是常量或表达式。

1）自增运算符。如，x++表示在使用x之后，使x的值加1，即x=x+1；++x表示使用x之前，先使x的值加1，即x=x+1。

2）自减运算符。如，x—表示在使用x之后，使x值减1，即x=x-1；--x表示使用x之前，先使x的值减1，即x=x-1。

2.关系运算符——用于比较运算。包括大于(>)、小于(<)、等于(==)、大于等于(>=)、小于等于(<=)和不等于(!=)六种。

关系运算符运算的结果是整型，值只有两种：0或1，0代表关系不成立，1代表关系成立。

3.逻辑运算符——用于逻辑运算。包括与(&&)、或(||)、非(!)三种。

a>b && c>d等价于(a>b) && (c>d)

!b==c||d<a等价于((!b)==c)||(d<a)

a+b>c && x+y<b等价于((a+b)>c) && ((x+y)<b)

　逻辑运算的值也为“真”和“假”两种，用“1”和“0 ”来表示。

|  |  |  |  |
| --- | --- | --- | --- |
| 条件A | 条件B | 结果（与） | 结果（或） |
| 真 | 真 | 真 | 真 |
| 真 | 假 | 假 | 真 |
| 假 | 真 | 假 | 真 |
| 假 | 假 | 假 | 假 |

真的“非”结果是假，假的“非”结果是真。

4.位操作运算符——参与运算的量，按二进制位进行运算。包括位与(&)、位或(|)、位非(~)、位异或(^)、左移(<<)、右移(>>)六种。

|  |  |  |  |
| --- | --- | --- | --- |
| 运算符 | 含 义 | 说明 | 例子 |
| & | 按位与 | 把参与运算的两个数对应的二进制位相与，只有对应的二进制均为1时，结果的对应位才为1，否则为0。 | 9&5中9可以写成(00001001),5可以写成(00000101)，那么9&5的运算结果为0000 0001，输出结果是1。 |
| | | 按位或 | 把参与运算的两个数对应的二进制位相或，也就是只要对应的两个二进制位有一个为1时，其结果就为1。 | 9|5相当于00001001|00000101，运算结果是00001101，输出结果是13。 |
| ^ | 按位异或 | 把参与运算的两个数对应的二进制位相异或，当对应的二进制位上的数据字不相同时，结果对应为1时，否则为0。 | 1^1=0，1^0=1，0^0=0，0^1=1  9^5相当于00001001^00000101，运算结果是00001100，输出结果是12。 |
| ～ | 取反 | 把运算数的各个二进制位按位求反。 | ～9相当于～(0000 1001)，运算结果为1111 0110。 |
| << | 左移 | 把“<<”左边的运算数的各二进制位向左移若干位，“<<”右边的数是指定移动的位数，高位丢弃，低位补0。 | A<<4指把a的各二进位向左移动4位，如a=00000011(十进制为3)，左移4位后为00110000(十进制48)。 |
| >> | 右移 | 把“>>”左边的运算数的各二进制位全部右移若干位，“>>”右边的数是指定移动的位数。 | 设a=15，a>>2表示把00001111右移为0000 0011(十进制为3)。 |

5.赋值运算符——用于赋值运算，分为简单赋值(=)、复合算术赋值(+=,-=,\*=,/=,%=)和复合位运算赋值(&=,|=,^=,>>=,<<=)三类共十一种。

6.条件运算符——这是一个三目运算符，用于条件求值(?:)。

7.逗号运算符——用于把若干表达式组合成一个表达式(，)。

8.指针运算符——用于取内容(\*)和取地址(&)二种运算。

9.求字节数运算符——用于计算数据类型所占的字节数(sizeof)。

10.特殊运算符——有括号()，下标[]，成员(->，.)等几种。

11.运算符的优先级

算术运算符、关系运算符、逻辑运算符和赋值运算符的优先级如下：

赋值运算符 < 逻辑运算符 < 关系运算符 < 算术运算符

低 高

　　关系运算符的结合性为：自左至右。

　　根据以上优先级和结合性，计算出以下表达式的结果（假设a=3，b=2，c=1）

a>b 表达式为真，所以表达式的值为1

(a>b)==c 表达式为真，所以表达式的值为1

b+c<a 表达式为假，所以表达式的值为0

d=a>b a>b为真，所以d的值为1

f=a>b>c a>b为真，结果1，1>c为假，所以f的值0

1. 常用的函数

|  |  |  |  |
| --- | --- | --- | --- |
| 函数名 | 格式 | 功能说明 | 例子 |
| 绝对值函数 | abs(x) | 求一个数x的绝对值 | abs(-5)=5 |
| 自然数指数函数 | exp(x) | 求实数x的自然指数ex | exp(1)=2.718282 | |
| 向下取整 | floor(x) | 求不大于实数x的最大整数 | floor(3.14)=3 | |
| 向上取整 | ceil(x) | 求不小于实数x的最小整数 | ceil(3.14)=4 | |
| 自然对数函数 | log(x) | 求实数x的自然数对数 | log(1)=0 | |
| 指数函数 | pow(x,y) | 计算xy,结果为双精度实数 | pow(2,3)=8 | |
| 随机函数 | rand() | 产生0到RAND-MAX之间的随机整数 |  | |
| 平方根值函数 | sqrt(x) | 求实数x的平方根 | sqrt(25)=5 | |
| 变量空间大小 | Sizeof(x) | 求变量x占用空间大小，字节输出 | Sizeof(int x)=4 | |