**二，变量和基本类型**

**1，字和字节**

可寻址的最小内存块称为字节，大多数机器的字节由8比特构成。大多数计算机将内存中的每个字节和一个数字（地址）关联起来。

存储的基本单元称为字，通常由几个字节组成，一般是4或8字节。通常float以一个字（32比特）来表示，double以2个字（64比特）来表示，long double以3或4个字来表示。

**2，如何选择类型**

1，当明确知晓数值不可能为负时，选用无符号类型。

2，一般使用int进行运算，如果超过int范围请使用long long类型。

3，执行浮点数运算选用double，long double提供的精度在一般情况下没有必要。

**3，无符号数运算**

当一个算术表达式中即有无符号又有int值时，那个int值就会转换成无符号数。

当从无符号数中减去一个值时，不管这个值是不是无符号数，都必须确保结果不能是负数。

**4，引用和指针**

引用

引用为对象起了另外一个名字。

初始化后，引用将和对象绑定在一起，无法重新绑定到另一个对象。引用必须初始化。

引用并非对象，不能定义引用的引用，也不能定义指向引用的指针。

指针

指定空指针的方法：int \*p1 = nullptr; =0; = NULL;

指向指针的指针 int \*\*ppi = &pi;

指向指针的引用 int \*p; int \*&r = p;

面对一条复杂的指针或引用的声明语句时，从右向左阅读有助于弄清楚它的真实含义。

**5，const限定符**

const对象一旦创建后其值就不能再改变，所以const对象必须初始化。

默认情况下，const对象被设定为仅在文件内有效。当多个文件中出现了同名的const变量时，其实等同于在不同文件中分别定义了独立的变量。

如果想在多个文件之间共享const对象，必须在变量的定义之前添加extern关键字。

std::string isbn() const{}中的const关键字的作用是修改隐式this指针的类型。

这个成员函数不能改变调用它的对象的内容。这样使用const的成员函数称作常量成员函数。

构造函数不能被声明为const。

**6，顶层const和底层const**

顶层const表示指针本身是一个常量，底层const表示指针所指对象是一个常量。

常量指针的定义：int \*const curErr = &errNumb;

指向常量的指针：const int\* curErr = &errNumb;

指向常量对象的常量指针:const int \*const curErr = &errNumb;

当执行对象拷贝的操作时，顶层const会被忽略，而底层的const的限制却不能忽视。

**7，constexpr**

常量表达式是指不会改变并且在编译过程就能得到计算结果的表达式。

允许将变量声明为constexpr类型以便由编译器来验证变量的值是否是一个常量表达式。

一般来说，如果你认定变量是一个常量表达式，那就把它声明成constexpr类型。

**8，处理类型**

有两种方法用于定义类型别名，传统的方法使用关键字typedef，新方法是使用using。

人们往往会错误地尝试把类型别名替换成它本来的样子，以理解该语句的含义。

编译器推断出来的auto类型有时候和初始值的类型并不完全一样：

1，编译器会以引用对象的类型作为auto的类型；

2，auto一般会忽略掉顶层const，同时底层const则会保留下来。

3，如果要定义引用和顶层const，需要手动指明。

decltype的作用是选择并返回操作数的数据类型。在此过程中，并不实际计算表达式的值。

如果decltype使用的是加了一层或多层括号的变量，得到的将是引用类型。decltype((variable))的结果永远是引用。