2，变量和基本类型

1，基本内置类型

参考文档：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，默认的浮点型字面值是一个double。long double提供的精度在一般情况下没有必要，而且运行时消耗不容忽视。

整型字面值后缀ll或LL表示long long，浮点型后缀l表示long double

除去布尔型和扩展的字符型之外，其他类型可分为带符号和无符号的两种。

unsigned是unsigned int的缩写。

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，我们都必须确保结果不能是负数。

2，变量

声明使得名字为程序所知，而定义负责创建于名字关联的实体。

如果想声明一个变量而非定义它，就在变量名前添加关键字extern，而且不要显示地初始化变量。

变量能且只能被定义一次，但是可以被多次声明。如果要在多个文件中使用同一个变量，就必须将声明和定义分离。

3，复合类型

C++两种重要的复合类型：引用和指针。

引用为对象起了另外一个名字。

引用必须初始化，一旦初始化完成，无法令引用重新绑定到另外一个对象。

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。

指针是对象，所以存在对指针的引用。int \*p; int \*&r = p;（从右向左阅读）

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

void\*是一种特殊的指针，可用于存放任意对象的地址，不能直接操作\*void指针所指的对象。

4，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。

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。

指针本身是不是常量以及指针所指的是不是常量就是两个独立的问题。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;

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。

const int number = 1;//这是一个常量表达式

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。

5 处理类型

auto类型说明符，用它就能让编译器替我们去分析表达式所属的类型。

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。

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。

decltype的作用是选择并返回操作数的数据类型。

如果decltype使用的是加了一层或多层括号的变量，编译器就会把它当成是一个表达式，得到的将是引用类型。

decltype((variable))的结果永远是引用，而decltype(variable)结果只有当variable本身就是一个引用时才是引用。

6，自定义数据结果

确保头文件多次包含仍能安全工作的常用技术是预处理器。

预处理命令：#define #ifdef #ifndef #endif

#ifndef SALES\_DATA\_H

#define SALES\_DATA\_H

XXXXXX

#endif

头文件即使没有被包含在任何其他头文件中，也应该设置保护符。头文件保护符很简单，程序员只要习惯性加上就可以了。

使用#pragma once可以获得相同的效果。但是，这种方式不支持跨平台！