**C++ 代码日志**

7.7

《C++ Primer Plus》第二章编程练习

7.8

转义字符、浮点数。

7.9

《C++ Primer Plus》 第三章编程练习1,2

《C++ Primer Plus》 第三章编程练习3,4,5,6,7

7.12

getline()函数，get()函数的用法，string的一些基本性质，基本操作（赋值，拼接和附加）。

7.13

string类的操作(size()方法， string类I/O)，结构体定义，结构体数组

7.14

联合体，枚举，指针

7.15

new开辟空间，delete释放空间，指针、数组和指针算术

new创建静态、动态数组，创建动态结构，类型组合(结构体指针，结构体指针数组，结构体数组指针)。

7.16

原版C创建的数组，vector类创建的数组，array类创建数组这三者的相似与区别。vector类与array类成员函数at()的用途。

《C++ Primer Plus》第四章编程练习1-7题。

7.17

《C++ Primer Plus》第四章编程练习8-10题。

7.18

for循环的一些用法

7.19

递增和递减运算符，前后缀表达式，递增/递减运算符和指针，逗号运算符，将输入的字符逆序打印（一种是从最后一个元素往前一个一个打印，一种是将最后一个和第一个元素位置互换，直到逆序完后打印出来）。

赋值运算符和等于运算符可能犯的错误，strcmp()函数的用法，while循环的基本用法。

7.21

二维数组、《C++ Primer Plus》第五章编程练习1-3题。

7.22

《C++ Primer Plus》第五章编程练习4-5题。

《C++ Primer Plus》第五章编程练习第6题（细节不完整）。

7.23

《C++ Primer Plus》第五章编程练习6-10题。（9-10含C语言实现）。

7.24

分支语句（统计空格字符，猜数字）

逻辑表达式（||、&&、！），用&&来设置取值范围，统计字符、空格、标点符号、数字个数。

7.25

？：运算符(求两数中的大数)，switch语句，将枚举量用作标签，continue和break语句，读取数字的循环。

演示cin输入非法时的现象，用户输入非数字输入，程序将拒绝，并要求用户重新继续输入数字

7.26

写入到文本文件中，读取文本文件，《C++ Primer Plus》第五章编程练习1-2题。

7.27

《C++ Primer Plus》第五章编程练习3-5题。

7.28

《C++ Primer Plus》第五章编程练习6-7题。

7.31

《C++ Primer Plus》第五章编程练习第9题

8.1

函数的声明，定义和调用

8.2

①函数中修改形参的值不会影响调用程序中的数据;

②计算彩票的中奖概率;

③函数和数组;

④将数组作为参数

函数调用sum\_arr(cookies, ArSize)将cookies数组第一个元素的地址和数组中的元素数目传递给sun\_arr()函数。

sum\_arr()函数将cookies的地址赋给指针变量arr。

⑤考虑要对房地产数组执行操作。两个基本操作分别是，将值读入到数组中和显示数组内容，再添加一个操作为：重新评估每种房地产的值。

⑥const的第一种用法 - 让指针指向一个常量对象，这样可以防止使用该指针来修改所指向的值

⑦const的第二种用法 - 将指针本身声明为常量，这样可以防止改变指针的位置。

⑧使用数组区间的函数 - 将指向数组起始处的指针作为一个参数，将数组长度作为第二个参数

8.3

①使用一个函数来计算特定的字符在字符串中出现的次数

②定义一个名为buildstr()的函数，该函数返回一个指针。该函数接受两个参数：一个字符和一个数字。函数使用new创建一个长度与数字参数相等的字符串，将每个元素都初始化为该字符。然后，返回指向新字符串的指针。

③将两个时间相加，首先将分钟成员相加。然后通过整数除法得到小时值，通过求模运算符得到剩余的分钟数。用sum函数来计算时间，使用show\_time()函数显示trave\_time结构的内容。

④用户输入x坐标和y坐标的值，通过rect\_to\_polar函数计算极坐标值，并通过show\_polar函数显示极坐标值。

8.4

①传递结构体的地址

②声明一个string对象数组，并将该数组传递给一个函数以显示其内容

③模板类array与数组的区别 - array对象进行传参是作为实参传递(主函数中与函数中地址不同)，而数组对象传参是传递首元素地址(主函数中与函数中地址相同)。

④函数与array对象

8.6

①演示递归的行为

②包含多个递归调用的递归

subdivide()函数使用一个字符串，该字符串除两端为|字符外，其他全部为空格。main函数使用循环调用subdivide()

函数6次，每次价格递归编号加1，并打印得到的字符串。这样，每行输出表示一层递归。

③演示使用函数指针。它两次调用estimate()函数，一次传递betsy()函数的地址，另一次则传递pam()函数的地址。在

第一种情况下，estimate()使用betsy()计算所需的小时数；在第二种情况下，estimate()使用pam()进行计算

④深入探索指针函数

⑤《C++ Primer Plus》编程练习第1题。

8.7

《C++ Primer Plus》编程练习第2-9题

8.8

《C++ Primer Plus》编程练习第10题（两种方式1.通过函数来调用函数；2.通过访问指针数组的内容来调用函数）

8.9

内联函数的演示

8.10

①引用变量 - &

②交换两个变量的值 - 函数使用三种传参方式

③使用两个函数来计算参数的立方，其中一个函数接受double类型的参数，另一个接受double引用。

④将引用用于结构

⑤使用string类，并演示不同的设计方案，其中一些是糟糕的。这个例子的基本思想是，创建一个函数，它将指定的字符串加入到另一个字符串的前面和后面。程序提供三个这样的函数，其中一个存在非常大的缺陷，可能导致程序崩溃甚至不能通过编译。

8.11

①程序通过调用同一个函数(只有函数调用参数不同)将数据写入文件和显示到屏幕上来说明这一点。该程序要求用户输入望远镜和一些目镜的焦距，然后计算并显示每个目镜的放大倍数。放大倍数等于物镜的焦距除以目镜的焦距加上0.5。

②默认参数的使用 - 设计一个left()函数，它将字符串和n作为参数，并返回该字符前n个字符。（方法一，会产生空间的浪费；方法二，不调用strlen()函数来计算字符长度）

③左值和右值的概念

8.12

①函数重载的例子 - left()函数，分别显示字符串的前n位和数字的前n位

②函数模板的例子 - 设计Swap()函数交换两个int类型的参数，和两个double类型的参数

③模板的重载 - 设计Swap()函数来交换int类型和double的参数，以及使用其重载来交换数组的内容

④显示具体化的例子 - 无法使用重载的模板来完成对结构体部分内容的交换(仅交换结构体的后两项，因此采用显示具体化；演示具体化优先于常规模板，而非模板函数优先于具体化和常规模板

⑤完全匹配和最佳匹配 - 有两个用来显示数组内容的模板定义。第一个定义(模板A)假设作为参数传递数组中包含了要显示的数据；第二个定义(模板B)假设数组元素为指针，指向要显示的数据

⑥关键字decltype

⑦另一种函数声明语法 - 后置返回类型

8.13

《C++ Primer Plus》第八章编程练习第1-3题

8.14

《C++ Primer Plus》第八章编程练习第4-7题

8.16

《C++ Primer Plus》第八章编程第1-6题修改以及改正

8.17

①单独编译，分别编写头文件，主函数文件和函数定义文件

②自动变量只在包含它们的函数或代码块可见

8.19

①演示如何使用关键字extern来重新声明以前定义过的外部变量，以及如何使用C++的作用域解析运算符来访问被隐藏的外部变量。

②程序除定义remote\_access()外，该文件还使用extern关键字来与第一个文件共享tom。接下来，该文件定义一个名为dick的静态变量。static限定符使该变量被限制在这个文件内，并覆盖相应的全局定义。然后，该文件定义了一个名为harry的外部变量，这不会与第一个文件中的harry发生冲突，因为后者链接性为内部的。随后，remote\_access()函数显示这3个变量的地址，以便于将它们与第一个文件中相应变量的地址进行比较。

③静态局部变量的用法演示 - 该程序演示了循环统计用户输入的字符的个数，且用循环处理了用户输入的字符数大于给定的字符数组的问题

④使用常规new运算符和定位new运算符创建动态分配的数组。该程序说明了常规new运算符和定位new运算符之间的一些重要差别。

8.22

①名称空间的示例，该示例说明了名称空间的一些特性。

②《C++ Primer Plus》第九章编程练习1-3题

9.2

①演示类的声明，实现类成员函数以及使用类

②构造函数的显式和隐式使用

两种方式定义构造函数（1.在声明部分为形参赋予默认值 2.重载的方式定义构造函数）

③该程序显示了构造函数和析构函数

④const成员变量

9.3

①this指针的用法

②该程序对个数组元素进行初始化，显示它们的内容，并找出这些元素中总值最高的一个。由于topval()每次值检查两个对象，因此程序使用for循环来检查整个数组。另外，它使用stock指针来跟踪值最高的元素。

③测试stack类 - 该程序模拟了售货员的行为--使用栈的后进先出方式，从购物筐的最上面开始处理购物订单

④《C++ Primer Plus》第十章编程练习1-3题

9.4

《C++ Primer Plus》第十章编程练习3-7题

9.5

《C++ Primer Plus》第十章编程练习第8题

《C++ Primer Plus》第十章编程练习2、4、6、8修改

9.6

①采用一个使用方法来处理假发的Time类。首先使用一个名为Sum()的常规方法

②该程序将调用operator+方法的两种表达法。在运算符表示法中，运算符左侧的对象（这里为coding）是调用对象，运算符右边的对象（这里为fixing）是作为参数被传递的对象。

③运算符重载限制的演示

④-和\*运算符的重载演示，并演示解决\*运算符在运用时出现的问题（传递的实参在\*的左边这种情况）第一种，在类外声明一个\*运算符的重载函数，并在函数中调用类的成员函数；第二种通过创建友元函数来解决

9.7

①常用的友元：重载<<运算符

②使用类来描述二维矢量

\* 用大小（长度）和方向（角度）描述矢量

\* 用分量x和y来表示矢量

此程序实现了对一个矢量的一些操作。首先将矢量分为直角坐标系表示和极坐标表示；通过枚举创建两个常量来表示这两种模式；在设计这个类时，用户修改矢量的一种表示模式，对象将自动更新另一种表示；同时，使用了运算符重载和友元函数的方法将矢量操作实现。

该程序演示如何使用Vector类：

该程序模拟了著名的醉鬼走路问题。其意思是，将一个人领到街灯柱下。这个人开始走动，但每一步的方向都是随机的（与前一步不同）。这个问题的一种表述是，这个人走到离灯柱50英尺处需要多少步，从矢量的角度看，这相当于不断将方向随机的矢量相加，直到长度超多50尺。该程序允许用户选择行走距离和步长。该程序用一个变量来表示位置（一个矢量），并报告到达指定距离处（用两种格式表示）所需的步数。可以看到，行走着前进得相当慢。虽然走了1000步，每步的距离为2英尺，但距离起点可能只有50英尺。这个程序将行走者所走的净距离（这里为50英尺）除以步数，来指出这种行走方式的低效性。随机改变方向使得平均值远远小于步长。为了随机选择方向，该程序使用了标准库函数rand()、srand()和time()。

9.8

①使用类的构造函数初始化了一些Stonewt对象，并处理类型转换

②演示使用转换函数 - 隐式调用转换函数和显示调用转换函数

③《C++ Primer Plus》第十一章编程练习第1-3题

9.9

《C++ Primer Plus》第十一章编程练习第4-7题

9.10

《C++ Primer Plus》第十一章编程练习第2-6题修改

9.13

①演示stringbad的构造函数析构函数是何时以及如何运行的。该程序演示了一个因复制构造函数调用析构函数而导致程序的错误。

错误原因1：复制构造函数的隐式调用

也可能的错误原因：重载的赋值运算符的隐式调用而导致

②该程序通过一个小程序来使用这个类，该程序允许输入几个字符串。程序首先提示用户输入，然后将用户输入的字符串存储到String对象中，并显示它们，最后指出哪个字符串最短，哪个字符串按字母顺序排在前面。

9.14

①使用new定位运算符合常规new运算符给对象分配内存，其中定义的类的构造函数和析构函数都会显示一些信息。但该程序会出现一些问题，比如调用了4个构造函数，但却只有2个析构函数，而且将不同内容定位到了同一个buffer空间

②修改placenew1中出现的问题

③ATM模拟

9.15

《C++ Primer Plus》第十三章编程练习第1-2题

9.19

《C++ Primer Plus》第十三章编程练习第3-6题

9.20

《C++ Primer Plus》第十三章编程练习第1-2题修改

9.21

①一个简单的基类

②编写一个派生类

③虚函数的用例，以及多态的公有继承

9.22

①使用了一个Brass对象和一个BrassPlus对象来测试类定义

②usebrass1z中，方法是通过对象（而不是指针或引用）调用的，没有使用虚方法的特性。以下程序使用了虚方法。假设要同时管理Brass和BrassPlus账户，如果能使用同一个数组来保存Brass和BrassPlus对象，将很有帮助，但这是不可能的。数组中所有元素类型必须相同，而Brass和BrassPlus是不同的类型。然而，可以创建指向Brass的指针数组。这样，每个元素的类型都相同，但由于使用的是公有继承模型，因此，Brass指针既可以指向Brass对象，也可以指向BrassPlus对象。因此，可以使用一个数组来表示多种类型的对象。这就是多态性，下面程序演示了这个例子

9.23

①抽象基类

②继承和动态内存分配

使用动态内存分配和友元继承示例

第一种情况：派生类不使用new。

第二种情况：派生类使用new。

9.24

①《C++ Primer Plus》第十三章编程练习第1-2题

9.25

①《C++ Primer Plus》第十三章编程练习第3-4题

9.26

①《C++ Primer Plus》第十三章编程练习第3-4题修改补充

9.27

包含对象成员的类

Student类中包含了string类对象和valarray对象，并可以通过公有的方法访问string和valarray类的方法

9.28

①私有继承，在studentc的基础上进行修改

②使用using重新定义访问方法

③

9.29

①该程序编写了一个Worker的抽象基类和其派生类Waiter和Singer

②多重继承

Worker为抽象虚基类，Waiter,Singer为Worker的派生类，SingingWaiter是Waiter，Singer的派生类

③类模板的定义(必须将所有模板信息放在一个头文件中，并在要使用这些模板的文件中包含该头文件)以及如何实例化类模板

10.8

①深入探讨模板类1:正确使用指针栈

②该程序像以前介绍的模拟那样使用rand()、srand()、time()来生成随机数，这里是随机生成0和1，来模拟掷硬币的结果

③该程序使用ArrayTP模板创建了一维数组，来分别保存这10个组(每组包含5个数)的总数和平均值。方法调用cout.width(2)以两个字符的宽度来显示下一个条目(如果整个数字的宽度不超过两个字符

④使用多个类型参数，该程序演示了创建并使用Pair模板来保存两个不同的值。其中，方法first() const和second()const报告存储的值，由于这两个方法返回Pair数据成员的引用，因此让您能够通过赋值重新设置存储的值

⑤演示模板的具体化:隐式实例化，显示实例化，显示具体化，部分具体化

⑥演示一个简短的模板类示例，该模板类将另一个模板类和模板函数作为其成员

10.9

①将模板用作参数

②非模板友元的演示

③模板类的约束模板友元函数 - 修改前一个例子，使友元函数本身成为模板。具体的说，为约束模板友元做准备，来使类的每一个具体化都获得一个与友元匹配的具体化。

④模板类的非约束模板友元函数 - 通过在类内部声明模板，可以创建非约束友元函数，即每个函数具体化都是每个类具体化的友元

10.10

《C++ Priemr Plus》第十四章第1-2题

10.11

《C++ Priemr Plus》第十四章第3题

10.12

《C++ Priemr Plus》第十四章第4题

10.13

《C++ Priemr Plus》第十四章第5题

10.18

《C++ Priemr Plus》第十四章第1-2题修正

10.19

友元类的演示

定义一个Tv类.可以用一组状态成员来表示电视机：

\*开关

\*频道设置

\*音量设置

\*有线电视或天线调节模式

\*TV调谐或A/V输入

遥控器的控制能力与电视机内置的控制功能相同，它的很多方法都可以通过Tv方法来实现。

10.20

①在Tv类和Remote类中，实际上直接访问Tv成员的Remote方法是Remote::set\_chan()方法，其他方法都不直接访问，因此无需将整个Remote声明为友元类，只需将set\_chan()方法声明为友元函数即可。但由于两个类循环依赖，因此需要前置声明来解决此问题

②嵌套循环的演示，使用队列模板，将原来队列里的结构体改为类

10.21

①异常

该程序用hmean()函数来求调和平均数的值，函数有两个参数，在运行阶段，若两个参数互为相反数:

在没有异常处理时，输入两个相反的数程序会输出无穷大的浮点数来表示，cout将这种值表示为inf

经过异常处理，利用abort()函数终止程序。

②返回错误码的方式来解决程序运行时出现的异常，这种方法可以在用户输入错误时提示错误，然后再继续往下让用户重新输入新的值

③C++异常处理机制：

处理异常的3个组成部分：

\*引发异常 （throw关键字）

\*使用处理程序捕获异常 （catch关键字）

\*使用try块 (try关键字)

④将对象用作异常类型

编写针对函数hmean()和gmean()引发的异常的两个类

10.22

①栈解退的演示。增加一个求调和平均数，平方根，两数平均数这三个数的平均值来演示。

②利用编程练习第二题演示exception类

③演示bad\_alloc和new。捕获到异常后，程序将显示继承的what()方法返回的消息(该消息随实现而异)，然后终止

10.23

异常，类和继承

在本程序中使用这些类：首先试图超越LabeledSales对象Sales2中数组的末尾，然后试图超越Sales对象sales1中数组的末尾。这些尝试是在两个try块中进行的。

10.24

①RTTI中dynamic\_cast的用法

②RTTI中的typeid运算符和type\_info类

③const\_cast的演示与用途

④static\_cast第一种用法

⑤利用static\_cast使类进行向上或向下的类型转换

⑥《C++ Primer Plus》第十五章编程练习第1题

10.25

《C++ Primer Plus》第十五章编程练习第2、3题

10.26

①《C++ Primer Plus》第十五章编程练习第4题

②《C++ Primer Plus》第十五章编程练习第3题修改

10.27

①演示C++11之前规定的7个string的构造函数

②此程序是从文件中读取字符串的简短示例，它假设文件中包含用冒号字符分隔的字符串，并使用指定分界符的getline()方法。然后，显示字符串并给它们编号，每个字符串占一行。

③本程序创建了一个非图形版本的拼字游戏。该游戏将一系列的单词存储在一个string对象数组中，然后随机选择一个单词，让人猜测单词的字母。如果猜错6次，玩家就输了。该程序使用find()函数来检查玩家的猜测，使用+=运算符创建一个string对象来记录玩家的错误猜测。为了记录玩家猜对的情况，程序创建了一个单词，其长度与被猜单词相同，但包含的是连字符。玩家猜对字符时，将用该字符替换相应的连字符。

④演示capacity()和reserve()方法

10.28

①演示如何使用全部三种智能指针。每个智能指针都放在一个代码块内，这样离开代码块时指针将过期。Report类使用方法报告对象的创建和销毁。

②演示不适合使用auto\_ptr的示例

③该程序创建了两个vector对象----一个是int规范，另一个是string规范，它们都包含5个元素

⑥演示了vector容器中的size()，begin()，ebd()，push\_back()，erase()，insert()和swap()的用法。为简化数据处理，将title和rating组合成了一个Review结构，并使用FillReview()和ShowReview()函数来输入和输出Review对象。

10.29

①演示STL中的for\_each(),random\_shuffle()和sort()函数。其中sort()函数有两种形式

②演示STL提供的一些预定义的迭代器。copy()算法，ostream\_iterator模板，reverse\_iterator

11.6

①演示关联容器set的set\_union, set\_intersection, set\_difference, lower\_bound, upper\_bound方法

②演示关联容器set的set\_union, set\_intersection, set\_difference, lower\_bound, upper\_bound方法

③演示函数符这种技术,包含remove\_if()方法

11.7

①演示STL中的算法next\_permutation()和sort()

②演示容器中的remove()方法，和STL中的remove()函数，以及erase()函数

③该程序让用户输入单词。希望最后得到一个按输入顺序排列的单词列表、一个按字母顺序排列的的单词列表(忽略大小写)，并纪录每个单词被输入的次数。出于简化的目的，假设输入中不包含数字和标点符号。

④演示transform()的用法

⑤该程序演示了vector和valarray的一些区别

⑥演示slice的一些特性

⑦一个简单的initializer\_list使用示例。

11.8

《C++ Primer Plus》第十六章编程练习第1-2题。

11.9

《C++ Primer Plus》第十六章编程练习第3-7题。

11.10

《C++ Primer Plus》第十六章编程练习第8-10题。

11.11

①《C++ Primer Plus》第十六章编程练习第1-10题修正。

②演示write()方法

11.12

①演示默认输出的情况，它在每个值后面都显示一个冒号(：)，以便可以知道每种情况下的字段宽度。该程序使用表达式1.0/9.0来生成一个无穷小数，以便能够知道打印了多少位

②以3种不同的计数系统显示了一个整数的值以及其平方。

③演示width()成员函数是如何工作的。

④演示fil()的用法

11.13

①演示precision()方法

②演示setf(ios\_base::showpoint)。

③演示setf()中的一些格式常量

④演示使用两个参数的setf()

⑤演示包含在头文件iomanip中的方法

⑥演示cin如何检查输入

⑦演示I/O和异常

⑧演示getline(), get()和ignore()

⑨采用两种方式来读取并显示输入中#字符（不包括）之前的内容。第一种方法读取字符，然后使用putback()将它插回到输入中。第二种方法在读取之前使用peek()查看下一个字符。

10. 使用peek()来确定是否读取了整行。如果一行中只有部分内容被加入到输入数组中，程序将删除余下内容。

11.15

①该程序要求输入文件名，然后创建一个名为输入名的文件，将一些信息写入到该文件中，然后关闭文件。关闭文件将刷新缓冲区，从而确保文件被更新。然后，程序打开该文件，读取并显示其内容。

②该程序结合使用命令行技术和文件流技术，来计算命令行上列出的文件包含的字符数。

11.17

①在文件尾追加数据。

②使用read()和write()方法来读取和创建二进制文件。

③运用seekg()和seekp()实现文件的随机存取

11.18

①内核格式化

②《C++ Primer Plus》第十七章编程练习第1-3题

11.19

《C++ Primer Plus》第十七章编程练习第4-6题

11.20

①《C++ Primer Plus》第十七章编程练习第7题

②《C++ Primer Plus》第十七章编程练习第1-7题修正

11.21

①演示有关右值引用的要点

②定义并使用了Useless类，这个类动态分配内存，并包含常规复制构造函数和移动构造函数，其中移动构造函数使用了移动语义和右值引用。为演示流程，构造函数和析构函数都比较啰嗦，同时Useless类还使用了一个静态变量来跟踪对象数量。另外，省略了一些重要方法，如赋值运算符。

③在Useless中添加了啰嗦的赋值运算符，并让以前啰嗦的构造函数和析构函数保持沉默，演示move()函数。