一、一些小问题

（1）一些C头文件包含了宏定义，指明了当前系统的限制信息(如：climits、cfloat等)

（2）Windows中，无法直接输出string类型，需要重载插入操作符或添加<string>头文件

---》实际上，本来只要定义string类型变量，就需要包含<string>头文件

1. c++类中，无法直接引入名字空间，但可以引入基类成员；成员函数可以直接引入名字空间。
2. cout.setf()设置输出格式：

---》输出布尔值字符串：cout.setf(ios::boolalpha); <==> cout << boolalpha;

---》设置输出浮点数的格式：cout.setf(ios\_base::fixed, ios\_base::floatfield);

---》设置宽度、间隔等

1. istream类提供了一个可以将istream对象(如cin)转换为bool值得函数：当cin出现在需要bool值的地方(如在while循环的测试条件中)时，改转换函数将被调用。读取成功为true（参考《C++ Prime》p 140）

（6）前++与后++：针对于内置类型和当代编译器而言，基本无区别；但C++允许针对类定义重载操作符，所以对类而言，前缀版本的效率比后缀版本高。

（7）逗号表达式：先计算第一个，后计算第二个...；逗号表达式的值是最后一部分的值

（8）C++建立别名两种方式：

---》（宏替换，不可定义指针类型）#define 别名 原类型

---》（更通用，指针类型的唯一选择）typedef 原类型 别名

1. 继承自C的字符函数库，包含头文件cctype(ctype.h)

---》用于检测字符类型和大小写转换的通用函数库（参考《C++ Prime》p 161）

===================================

二、C字符串输入问题

（1）由于不能通过键盘输入空字符，因此cin(以及C语言的scanf)需要用别的方法来确定字符串的结尾位置；cin使用空白(空格、制表符('\t')和换行符)来定字符串的界。

（2）每次读取一行字符串输入

---》有时候字符串输入需要含有空格(比如New York)，此时需要cin.get()/cin.getline()整行读取(面向行的输入和面向单词输入)

cin.get(name,20)/cin.get(name,20).get(): 读取到换行符，保留在输入序列(接下来读取)

cin.getline(name,20): 读取到换行符，丢弃

（3）混合输入字符串和数字

---》cin读取数字后，cin.get()读取回车键

（4）读取到空行(回车键生成的‘\n’)

---》get():设置失效位，阻断后续输入；cin.clear()可以恢复输入;

---》getline():解读为空字符赋值给字符串；

（5）输入的字符串比分配的空间长

---》getline():把余下的字符留在输入队列，设置失效位，并关闭后续输入

---》get():把余下的字符留在输入队列

---》5、6和17章探讨如何避免

注：string类型面向行输入时的不同

---------------------------------------

三、结构、联合、枚举

（1）可以定义时同时创建变量

（2）可以创建匿名联合，通过父级直接访问

（3）枚举类型、枚举变量、枚举量的区别；

（4）枚举取值范围？

---------------------------------------

1. new/delete操作符
2. 动态分配数组：程序运行时选择数组的长度 --- 动态绑定(动态联编)

---》用法类似变长数组，通过动态分配内存实现。

---------------------------------------

五、指针与数组

（1）指针与数组名两个区别： 指针可以修改指向； 数组名是常量；

（2）两种解引用方式： 指针方式（\*pn）； 数组方式（pn[0]）

---------------------------------------

六、递增/递减操作符和指针解引用顺序

（1）前缀递增、递减和解引用操作符优先级相同，从右到左结合

（2）后缀递增、递减优先相同，比解引用优先级高，从左到右结合

---------------------------------------

七、

===================================

C++面向对象：

1. 静态绑定和动态绑定（动态联编）：编译/运行时指定

---》new[]操作符动态创建数组

EOF文件结束符：

---》键盘输入模拟文件结尾符（EOF）：Ctrl+Z、Ctrl+D或回车键

---》检测文件结束符：cin.eof()/cin.fail()（参考《C++ Prime》p 138~139）

---》符号常量EOF在iostream中定义，通常为-1，区别与其他ASCII

C++逐字读取比较：

---》cin >> ch：跳过空白(空格、换行符合制表符)，并留在输入序列中

---》函数cin.get(ch)与ch = cin.get()的差别。（参考《C++ Prime》p 142）

---》cin.get(char)调用通过返回转换为false的bool值来指出已达到EOF，而cin.get()调用则通过返回EOF值来指出已达到EOF（参考《C++ Prime》p 142 总结）

针对数字的错误输入：

---》cin.clear()重置;删除错误输入;提示用户再输入

---》（参考《C++ Prime》p 168~170）