

## 极客班第一期 C++ 专业期末综合测试题

### 说明：

1. 极客班第一期学员可通过项目实践（样本项目或者自选项目）或者此试题来参加期末考评。
2. 答案提交截止日期为2015年10月18日 23:30。
3. 请将答案提交到自己在极客班github的C++ Homework资源库里。

地址：<https://github.com/GeekBand/GeekBand-CPP-1501-Homework>

- 请在以自己学号命名的文件夹中，创建文件夹“C++综合测试”，并将作业文档（PDF格式）和代码通过pull request的方式提交。
  - 请在作业文档中注明题目和对应的代码文件夹（部分题目所需的关键代码可以写在作业文档里），方便老师批改。
4. 此试题分数（或项目实践）占整个专业考评的70%；课程测验（平时作业），占20%；课堂优秀表现（分享、讨论、创新），占10%。

## 面向对象与设计

1. 设计一个多边形 Polygon，其由若干个点连成线。为其实现构造函数，拷贝构造函数，复制操作符，析构函数。

```
class Shape{  
  
    int no; //形状编号  
  
};
```

```
class Point{  
    int x;  
    int y;  
};  
  
class Polygon: public Shape{  
    Point* points;  
};
```

2. 为 Point 类设计一个数据绑定机制，当其坐标 x 或 y 被更改时，可以通知外界其更改的过程（即从旧值改为新值）。将更改过程打印在控制台上。考虑使用松耦合设计（即未来有可能将更改过程显示在任何界面上）。

```
class Point {  
    int x;  
    int y;  
}
```

# STL与泛型编程

1. 给定一个 vector : `v1 = [0, 0, 30, 20, 0, 0, 0, 0, 10, 0]` , 希望通过 `not_equal_to` 算法找到到不为零的元素 , 并复制到另一个 vector: `v2`

2. 为以下 `Programmer` 对象提供一个基于 `Id` 并且升序的仿函数

`ProgrammerIdGreater` , 使得 `Programmer` 对象可以在 `set` 中以 `Id` 排序存放

- (1) 将 `Programmer(1, L"Scott Meyers")`、`Programmer(2, L"Martin Fowler")`、`Programmer(3, L"Bill Gates")`、`Programmer(4, L"P.J. Plaught")`、`Programmer(5, L"Stanley B. Lippman")`、`Programmer(6, L"Andrei Alexandrescu")`插入到一个 `set` 中 ;
- (2) 通过 `for_each` 遍历 `set` , 并且使用 `Programmer` 对象的 `Print` 方法打印出对象 , 结果应该如下所示 ;

```
[6] : Andrei Alexandrescu
[5] : Stanley B. Lippman
[4] : P.J. Plaught
[3] : Bill Gates
[2] : Martin Fowler
[1] : Scott Meyers
```

- (3) 在 `set` 中查找 `Id` 为 3、名称为 “Bill Gates” 的对象 ;
- (4) 如果(2)中找到目标对象 , 则将其名称改成 “David Vandevoorde” , 再通过 `for_each` 重新打印 `set` , 查看是否真的修改成功了 ;
- (5) 重新定义一个 `set`(命名为 `set2`) , 将 `Programmer` 在 `set2` 中排序方式改为通过 `Name` 来排序 , 为此还需重新定义一个仿函数用于比较 `Name` , 请写出该仿函数 , 名字可能类似 `ProgrammerNameComparer`;
- (6) 通过 `for_each` 重打印 `set2` , 验证 `set2` 中的元素是否是按照名字来排序的。

```

struct Programmer
{
    Programmer(const int id, const std::wstring name)
        : Id(id), Name(name)
    {
    }

    void Print() const
    {
        std::wcout << L "[" << Id << L "] : " << Name << std::endl;
    }

    int Id;
    std::wstring Name;
};

struct ProgrammerIdGreater { ... }

```

# 算法与系统设计

## 1. Moving Average

给定一移动窗口，计算在这个窗口内的平均数，举例：

```
MovingAverage ma = new MovingAverage(2);
```

```
ma.next(1) return 1
```

```
ma.next(3) return 2
```

```
ma.next(4) return 3.5
```

设计这个class，并给出测试用例

## 2. Total Difference Strings

给一个string列表，判断有多少个不同的string，返回个数

相同的定义：字符串长度相等并从左到右，或从右往左是同样的字符

abc和cba为视为相同

举例：

abc

cba

abc

Aaa

返回不同的个数：2

### 3. Binary Tree Print

给出一个二叉树，打印所有从root到叶子节点的路径

```
      3
     / \
    1   5
   / \
  2  6 7
   \
    8
[3,1,2],[3,5,6,8], [3,5,7]
```

2) bonus question: print in beautiful order, indent needed

```
      3
     /
    1
   /
  2
-----
      3
     /
    5
   /
  6
 /
8
-----
3
5
7
```

### 4. Design Shopping Cart

如何设计一个购物车，可以从商品列表中添加商品，修改数量，生成订单，如果商品数量在1亿以上，如何设计架构保证安全稳定的，  
(multi-tier, MVC, SOA)

bonus：如何设计抢购页面, suppose we have limited products and time