

# 2020春季学期C++整理2.0

## 一.内联函数

内联函数是在函数调用中出现的一种机制, 关键字 `inline`, 编译器看到 `inline` 后, (一般)为该函数创建一段代码, 以便在后面每次调用都用一段代码来替换, **内联函数的初衷是为了空间换取时间的更快**

内联函数的声明:

- 1.内联函数可以在一开始仅声明一次
- 2.内联函数**必须在调用之前被声明或定义**, 因为内联函数的代码必须在替换之前已经生成被替换的代码
- 3.在实现的时候可以不用加 `inline`

**问题: 内联函数和普通函数有啥区别?**

内联函数和普通函数最大的区别在于内部的实现方面, **当普通函数在被调用时, 系统首先跳跃到该函数的入口地址, 执行函数体, 执行完成后, 再返回到函数调用的地方, 函数始终只有一个拷贝; 而内联函数则不需要进行一个寻址的过程**, 当执行到内联函数时, 此函数展开 (很类似宏的使用), 如果在 N 处调用了此内联函数, 则此函数就会有N个代码段的拷贝。

代码样例:

```
1 inline int max(int a,int b)
2 {
3     return a > b;
4 }
```

## 二.函数重载

一个或者多个函数, **名字相同, 然而参数个数和参数类型不相同**, 这就叫做函数的重载。

例如下列三个函数:

```
int max(double a,double b)
```

```
int max(int a,int b)
```

```
int max(int a,int b,int c)
```

他们之间就是一个重载关系。

实际应用中, 编译器会通过调用语句中的实际参数个数和类型来判断应该调用那个函数。

**注意: 调用时应该避免二义性!!**

比如:

```
1 max(3.4,2.5); //ok
2 max(2,3); //ok
3 max(1,5,4); //ok
4 max(3,3.4); //二义性!!!
```

## 三.函数的缺省参数

c++中，定义函数时可以让最右边的连续若干个参数有缺省值，那么调用函数的时候，若相应位置不写参数，参数就是缺省值。

例如：

```
1 void func(int x1,int x2 = 2,int x3 = 3)
2 {
3     .....
4 }
5 func(10 );//ok
6 func(10,8);//ok 可以对缺省值进行修改
7 func(10, , 8);//error 只能最右边的连续参数缺省
```

## 四.类和对象

类构成了实现C++面向程序设计对象的基础。类用来定义对象的特征和行为，类是C++的基本封装单元。

**C++的类是一种数据类型**但是和其他数据类型不同的是，组成这种类型的不仅可以有数据，还可以有对数据进行操作的函数。

基本格式：

```
1 class 类的名称
2 {
3     数据成员+ 成员函数
4 };
```

数据成员用来描述对象的属性，但是在**类中不允许对数据成员进行初始化**

成员函数用来描述对象的行为，与普通函数一样，**它可以重载，可以使用默认参数，还可以声明内联函数。**

**类成员的可访问范围：**

```
1 class classname
2 {
3     private://私有属性和函数
4     public://公有属性和函数
5     protected://保护属性和函数
6 };
```

**注意：如果某个成员之前没有加上关键字，系统默认为 private 型。**

在类的成员函数内部，能够访问：

- 当前对象的全部属性和函数。
- 同类其他对象的全部属性和函数。

在类的成员函数意外：

- 只能访问该类对象的公有成员。

```
1 class classname
```

```

2  {
3      private:
4          char name[30];
5      public:
6          int val;
7          void getname(char * name);
8      void getval(classname x);
9  };
10 void classname :: getname(char * name)
11 {
12     strcpy(name, "kyrie"); //OK
13 }
14 void classname :: getval(classname x)
15 {
16     cout << x.name; //ok访问同类对象私有成员
17     cout << x.val;
18 }

```

## 五.关于class和struct:

来看一下CSDN的大佬写的blog:

1、最主要的区别，**默认的继承访问权限**。**struct是public的，class是private的。**

2、关于使用大括号初始化

class和struct如果定义了构造函数的话，都不能用大括号进行初始化

**如果没有定义构造函数，struct可以用大括号初始化。**

**如果没有定义构造函数，且所有成员变量全是public的话，可以用大括号初始化。**

3、关于默认访问权限

**class中默认的成员访问权限是private的，而struct中则是public的。**

4、关于继承方式

class继承默认是private继承，而struct继承默认是public继承。

5、其实二者在C++中的区别并不大

(1) 都可以有成员函数；包括各类构造函数，析构函数，重载的运算符，友元类，友元结构，友元函数，虚函数，纯虚函数，静态函数；

(2) 都可以有一大堆public/private/protected修饰符在里边；

(3) 虽然这种风格不再被提倡，但语法上二者都可以使用大括号的方式初始化：A a = {1, 2, 3};不管A是个struct还是个class，前提是这个类/结构足够简单，比如所有的成员都是public的，所有的成员都是简单类型，没有显式声明的构造函数。

(4) 都可以进行复杂的继承甚至多重继承，一个struct可以继承自一个class，反之亦可；一个struct可以同时继承5个class和5个struct，虽然这样做不太好。

原文链接：<https://blog.csdn.net/zzucsliang/article/details/43876199>

## 六.题解:

主要是那一道编写函数题想了好久，最后想起来可以用static来保护变量。其次感觉main函数里的东西不难懂。

## AC代码:

```
1  #include <iostream>
2  #include <iomanip>
3  using namespace std;
4  double GetDoubleFromString(char * str)
5  {
6      static char* p;//保护p不会随循环改变
7      if(str != NULL) p = str; //关键在这里!! 这句话实现的循环
8      double num = 0;
9      while(*p && !(*p >= '0'&& *p <= '9')) p++;
10     // cout << *p << endl;
11     if(*p == '\0') return -1;//判断是否读取完毕
12     while(*p >= '0'&&*p <= '9'){//整数累加
13         num = num * 10.0 + *p - '0';
14         p++;
15     }
16     if(*p == '.'){//小数处理
17         int l = 10;
18         p++;
19         while(*p >= '0'&&*p <= '9'){
20             num += (double)(*p - '0')/l*1.0;
21             l *= 10;
22             p++;
23         }
24     }
25     return num;
26 }
27
28 int main()
29 {
30     char line[300];
31     while(cin.getline(line,280)) {//getline函数读取一整行
32         double n;
33         n = GetDoubleFromString(line);
34         while( n > 0) {
35             cout << fixed << setprecision(6) << n << endl;
36             n = GetDoubleFromString(NULL);//继续循环
37         }
38     }
39     return 0;
40 }
```