

# C到C++2

## 一、new和delete

### 1、知识点介绍

1、在C++中用来向系统申请堆区的内容空间，new等价于C语言中的malloc,delete等价于C语言中的free

### 2、new使用

1、申请单个内存

```
int *p=new int;
```

2、申请单个内存初始化

```
int *p=new int(10);
```

3、申请多个连续的空间

```
int *p=new int[10];
```

### 3、new的特性

1、new是运算符

2、new可以分配两种内存，一种是单个内存，一种是连续的内存

3、new只有分配单个内存的时候可以同时给该内存赋值，分配连续多个内存是无法同时赋值的

4、如果要给连续的内存同时赋值用memset赋值，赋值为0

### 4、delete使用

1、释放单个内存

```
delete p;
```

```
p=NULL;
```

2、释放连续内存

```
delete []p;
```

```
p=NULL;
```

### 5、delete的特性

1、delete 是运算符

2、delete可以释放两种内存，一种是单个内存，一种是连续内存

3、delete释放内存时，指针的必须是指向内存首地址

4、释放完内存之后要记得给指针赋值NULL操作，不要产生野指针

## 6、动态内存操作中常见的问题

- 1、没有给指针分配内存，就对该指针解引用赋值

```
int*p; *p=10;
```

- 2、给指针分配内存成功，但是没有初始化就使用了

```
int*p=new int; printf("%d\n",*p);
```

- 3、内存分配成功，也初始化了但是操作越界

```
int*p=new int[5]; *(p+7)=123;
```

这种操作通常情况下，因该都可以正常执行，但是这样很有大的问题，因为你不知道越界到的地方是否有在使用的数据，如果有那么这种越界操作就会改变哪里的数据，出现一些问题

- 4、忘记释放内存，造成内存泄漏

有申请内存那么就要记得使用完之后释放内存，如果不释放可以用的堆区内存就会变少

- 5、释放内存之后，继续使用

```
int *p=new int; *p=10; delete p; *p=20;
```

## 二、命名空间

### 1、知识点介绍：

- 1、命名空间是用来组织和重用代码的编译单元。

- 2、在编写代码时写的程序不可能所有的标识符都没有重名现象，在多人协同开发时更加不可控，尤其对于库来说问题更加严重。为了解决重名现象，通过命名空间来避免冲突。

### 2、命名空间的定义：

namespace 命名空间标识符

```
{  
...命名空间成员  
}
```

花括号只有两种情况：1、定义域（必须加分号）；2、作用域（可加可不加）

### 3、注意：

- 1、命名空间标识符必须满足标识符的命名规则和命名规范，习惯名字唯一，通常以开发团队的名字（项目名）来命名
- 2、命名空间可以在全局，也可以在局部（命名空间接受嵌套定义），但不能在函数内和类中定义
- 3、命名空间的花括号是作用域
- 4、注意命名污染，尽量规避同名的出现，如果两个命名空间同名，那么就会合并两个命名空间

## 4、命名空间访问：

作用域运算符 `::`：  
可以理解为， 什么的什么 `::` 就是那个的

- 1、**using 声明**：`using 命名空间名称::空间成员名称;`  
放开一个成员的可见性，这一个成员可以在空间外可见(建议大家使用这种，比较容易避免出现命名污染)
- 2、**using 指示**：`using namespace 命名空间名;`  
放开该命名空间的所有权限(所有成员都在空间外可见)，适用于该空间的大部分成员都需要经常被使用
- 3、**命名空间名称::** 空间成员名称，直接访问空间下的某一个成员

## 5、命名空间取别名：

- 1、`namespace 别名=命名空间名`
- 2、当命名空间标识符过长或不太方便记忆，可通过取别名的方式来表示该命名空间，别名的操作等价于原命名空间

## 6、命名空间成员的声明及定义：

<code>namespace A</code>	<code>void A::fun()</code>
<code>{</code>	<code>{</code>
<code>void fun();</code>	函数功能;
<code>}</code>	<code>}</code>
声明	命名空间外实现

# 三、cin和cout

## 1、知识点介绍

`cin`的作用和`scanf`一样  
`cout`的作用和`printf`一样  
他们在是具体使用的时候有些区别，C++不需要记格式占位符

## 2、使用前准备

- 1、需要包含头文件`<iostream>`
- 2、需要声明命名空间`std`中的权限
- 3、C++的头文件，不帶`.h`,如果你要包含C语言的头文件正常写，C++兼容C，或者`#include <cstdio>`

## 3、cin与cout

示例：

```
int x,y,z;  
cin>>x>>y>>z;//从键盘对xyz赋值 输入的时候每个值中间用空格隔开  
cout<<x<<y<<z;//输出xyz的值  
cout<<endl;//输出一个换行  
cout<<"这里是你要输出的文字！！"<<"\n"; //转义字符还是可以用在双引号中写上即可
```

## 四、string的基本使用

### 1、知识点介绍

- 1、string是c++中的字符串。类似于c语言中的字符数组。
- 2、string是系统提供的一个类（大概先了解一下，暂时不需要掌握类的概念）

### 2、string的基本使用

- 1、需要包含头文件<string>和声明命名空间std中的权限
- 2、赋值： `string str1="qwertyu",str2; str2 = str1;`
- 3、求长度： `int len = str1.length();`
- 4、清空字符串： `str1.clear();`
- 5、判断字符串是否为空： `if(str1.empty() == NULL)`
- 6、比较字符串是否相同： `if(str1 == str2)`
- 7、得到字符串中某一个位置的字符：  
`char c = str1[3];`  
`char c1 = str1.at(3);`  
注意：这2个3指的是字符串中下标(从0开始)为3的字符，不能越界

## 补充：

- 1、三名运算符，在C语言中，返回的是一个常量，不能被赋值的，C++返回的是变量
- 2、C++的函数必须要写返回值类型
- 3、在全局下面，C++不允许`int a;int a=10;`这种二义性操作
- 4、不要返回局部变量或临时变量的地址
- 5、C语言用const修饰的变量不是不能用来当作定义数组时，数组大小，C++的时真的，可以用

## 作业：

使用命名空间写好一下3个功能（要求分文件编写）：

- 1、能完成对int类型的一维数组的排序
  - 2、能倒序输出一个字符串
  - 3、能在一个int类型的一维数组中找到最大的值
- 以上三个功能，分别写成3个函数，数据可以提前定义好  
提交运行结果截图和代码截图发给2829114166（QQ邮箱）  
格式：期号+名字+作业名字 （2012+易木+第二节课作业）