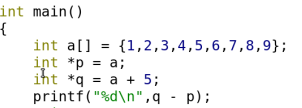
2018.8.10

上午：

一：指针：是一种数据类型

**1.指针**

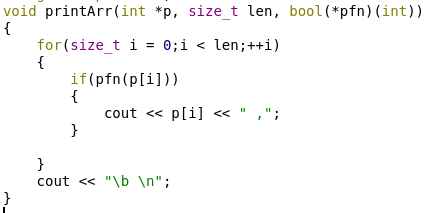


结果是（int型）5，求偏移量【偏移几个基类型】

**2.指针数组**

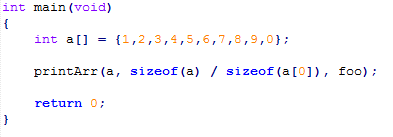
**3.函数指针**

便利函数：



回调函数：

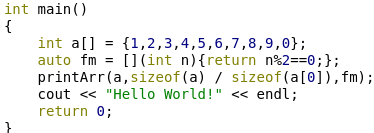




***---------------------------------------------------------------------------------------------------------------------***

***C++中：lamda表达式：[capture捕获外部变量] (argment参数) { body函数体 };***

***返回的是函数指针***



方式二：



**[capture捕获外部变量] (argment参数) { body函数体 };**

值捕获 ，引用捕获；【捕获就不可以作为函数的指针使用】【传参才可以作为函数指针】





[=]捕获前面的所有变量值

[&]捕获前面的所有变量本身

int i = 100;

int j = 11;

int k =10100; Int I =



捕获后lamada不可以再做函数指针；

返回值可以手写：【本身可以自己判断】



4.指向类的非静态成员指针

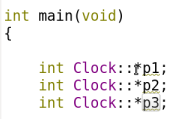
指向数据成员指针：

Int hour;

Char a;

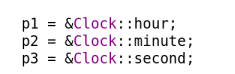
Int minute;

Int second;



目前为野指针：

赋值：



但是没有定义类，指针里装的是什么：

P1=null p2=0x8 p3=0xc【相当于对象的首字节偏移量】



打印出22:；

5. 指向函数成员指针

void show()

{

cout << hour << ":" << minute << ":" << this->second << endl;

}

野指针

使用：



【&，：：，函数名】

【&不可省略】

先创建类：



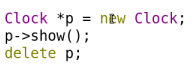
有疑问？？？？

6.动态空间分配



不是clock的对象；只是一个12字节空间

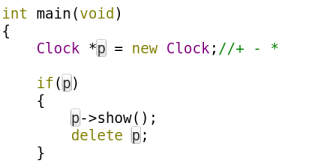
取而代之的是new；

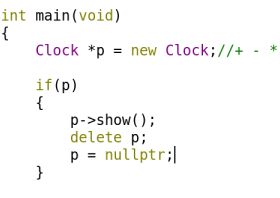


这个会调用构造函数，析构函数；new是个运算符

Malloc是函数

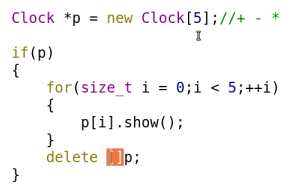
如果失败new会返回nullptr





这类对象需要程序员自己销毁

一次性new多个对象



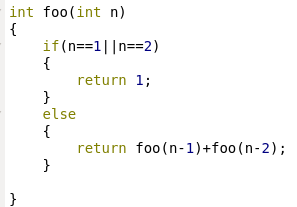
[]表示析构多个，

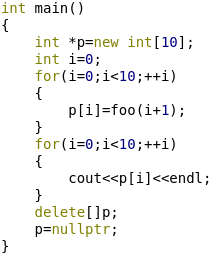
再加一句 p=nullptr；

带参构造：



7.例题：new数组 往里写斐波拉切数列在打印出来





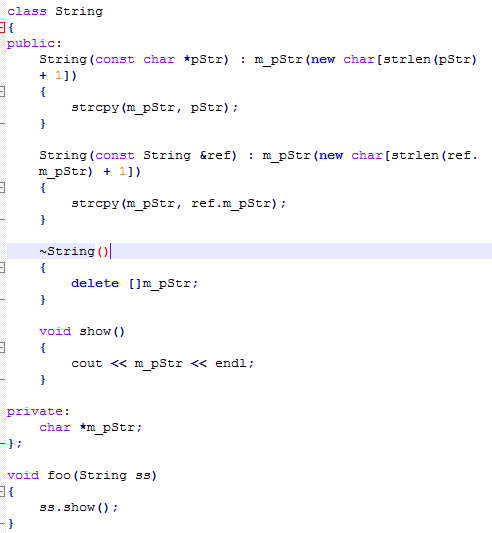
下午：

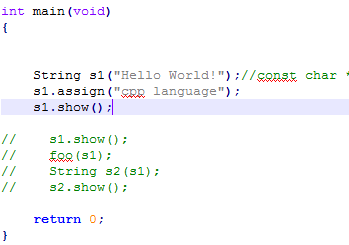
一：动态内存分配

1. 深复制与浅复制

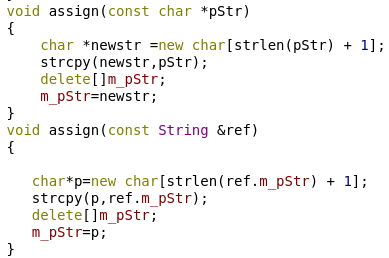
复制构造函数复制字符串需要深复制；

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

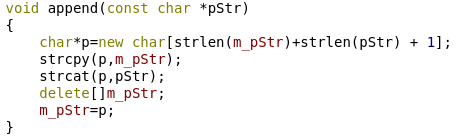




1. 替换函数：

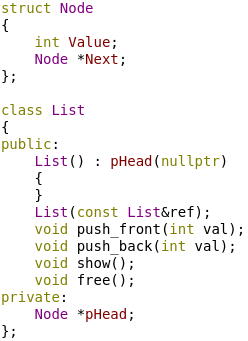


1. 追加

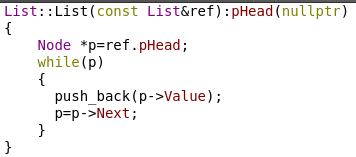


二：链表类：

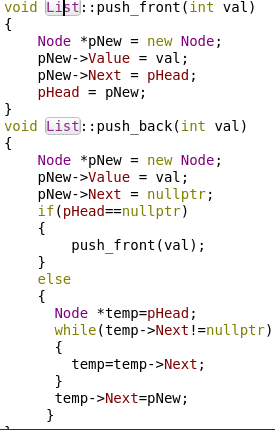
抽象：节点类，链表里是头，



复制构造：



头插尾插：



打印和销毁：

