

همین خدا

برنامه‌نویسی پیچیده با C++
علیه هستیم

- Copy constructor
- function over loading

copy ctor:

(classname &obj)

همین نام به علاوه
خروجی ندارد
آرگومان ورودی

در حالت
بصورت خودکار، فراخوانی می‌شود
۱- ایجادش ضرورت ندارد
اولیه آن به یک شیء قبلی

Point P3 = P1;
P3. Point(P1)

۲- فراخوانی می‌شود به صورت call by value

compX (Point P1, Point P2) {
...
compX (P3, P4)
}

P1. Point(P3) ← Point P1 = P3;
P2. Point(P4) ← Point P2 = P4;

P3 = largerX (P1, P2);
Point tmp = P
P3 = tmp
destroy(tmp)

۳- بازگرداندن شیء در تابع
largerX () {
return P;
}

اند تابع cstr cpy را در یک ماس پیاده سازی کنیم، ما باید خودت را به cpy cstr
 ماس اضافه کند که در آن هم data members بیت به بیت در data members
 پیاده شود

Point (Point & obj) {

this->x = obj.x;
 this->y = obj.y;
 this->z = obj.z;

}

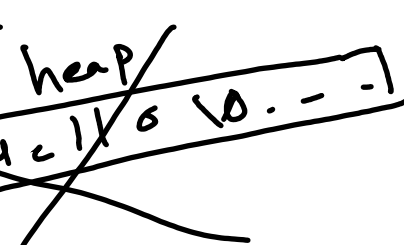
cstr
destr



0x0f...b2



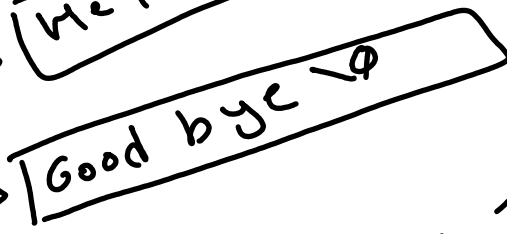
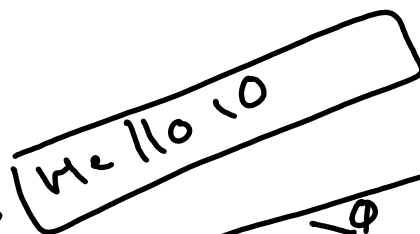
0x0f...b2



str1.s
2



0x0f...b2



String str2 = str1

در حالتی که در ماس خود attr خاص داریم
 که آن را به یک حافظه است دستور از یک
 سازی میخوان حافظه است نه سازی آن را که
 لازم است تابع cpy را خود را پیاده سازی کنیم.

function Overloading :

ایجاد چند تابع با نام یکسان که در تعداد یا نوع آرگومان های ورودی با یکدیگر تفاوت دارند. در حین فراخوانی تابع، کامپایلر براساس آرگومان های ورودی تصمیم می گیرد که کدام نسخه تابع اجرا شود.

