

Table of Contents

.....Prevent exceptions from leaving destructors \

Item 10: Have assignment operators return a reference to *this

یکی از جالب توجه‌ترین ویژگی‌های **assignment** این است که شما می‌توانید زنجیره‌ای از آن‌ها را داشته باشید.

```
int x,y,z;  
x=y=z=15;
```

همچنین در نظر داشته باشید که انتساب یک عمل راست به چپ هست، بنابراین انتساب‌های بالا به صورت زیر **parse** خواهند شد:

```
x=(y=(z=15));
```

در اینجا ۱۵ به **z** انتساب داده می‌شود، سپس نتیجه این انتساب به **y** انتساب می‌شود و نتیجه‌ی آن هم به **x** انتساب داده می‌شود.

راهی که این انتساب پیاده‌سازی شده این است که انتساب یک رفرنس به آرگومان سمت چپ برمیگرداند، و این رویه رو هم شما بهتره برای پیاده‌سازی کلاس‌هاتون در نظر بگیرید.

```
class Widget{  
public:  
    Widget& operator=(const Widget& rhs) //return type is a refrence to the current class  
    {  
        return *this; //return the left-hand object  
    }  
};
```

چنین رویکردی برای همه‌ی اپراتورهای انتساب قابل تعمیم هست، نه فقط اپراتوری که در بالا دیدیم.

```
class Widget{  
public:  
    Widget& operator+=(const Widget& rhs) //the convention applies to +=,-=,*=,etc.  
    {  
        return *this;  
    }  
    Widget& operator=(int rhs)  
    {
```

```
        return *this;  
    }  
};
```

در C++ این یک اجماع نظر است، کدی که از این رویه طبیعت نکند نیز کامپایل خواهد شد. اگر چه، این رویکرد برای همه‌ی تایپ‌های **built-in** رعایت شده، برای **stl** ها نیز همینطور هست. مگر این که شما دلیل بهتری برای یک رویکرد متفاوت داشته باشید.