



دانشگاه ریاضی و آمار



نیمسال اول ۱۴۰۰-۱۴۰۱

مدرس: دکتر مجتبی رفیعی

مبانی کامپیوتر و برنامه سازی

جلسه ۲۳ : عملگرهای جاوا

نگارنده: زهرا محبوب خلجانی

۶ آذر ۱۴۰۰

فهرست مطالب

(۱) عملگرها

(۱.۱) عملگر افزایش و کاهش ..... ۲

(۲.۱) عملگرهای بیتی ..... ۳

(۳.۱) عملگرهای انتساب ..... ۵

(۲) اولویت عملگرها در جاوا ..... ۶

## (۱) عملگرها

(۱.۱) عملگر افزایش و کاهش: عملگر افزایش و کاهش که به صورت ++/-- نمایش داده می شود، مقدار متغیر را یک واحد افزایش و یا یک واحد کاهش می دهد.

نکته: عملگر افزایش و کاهش می تواند قبل و یا بعد متغیر قرار گیرد و نحوه تغییر آن متغیر را متفاوت سازد.

حالت اول: عملگر افزایش و کاهش بعد از متغیر قرار گیرد:

Int i=۱۰

System.out.print (i++);                      Result=۱۰

System.out.print (i);                          Result=۱۱

\* همانطور که مشخص شده ابتدا متغیر برگردانده میشود و سپس یک واحد به آن اضافه میشود.

حالت دوم: عملگر افزایش و کاهش قبل از متغیر قرار گیرد:

Int i=۱۰

System.out.print (++i);                      Result=۱۱

System.out.print (i);                          Result=۱۱

ابتدا یک واحد اضافه شده و سپس متغیر برگردانده شده است.

## (۲.۱) عملگرهای بیتی

• عملگر (&) and

• عملگر (|) or

• عملگر (^) xor

b1 b2	and
0 0	0
0 1	0
1 0	0
1 1	1

b1 b2	or
0 0	1
0 1	1
1 0	1
1 1	0

b1 b2	xor
0 0	0
0 1	1
1 0	1
1 1	0

- عملگر شیفت به راست ( $>>$ )
- عملگر شیفت به چپ ( $<<$ )

\*یک صفر از سمت راست اضافه میشود، معادل تقسیم عدد بر ۲ می باشد:

$$\begin{aligned}
 12 &>> 1 \\
 12 &= 1100 \\
 \text{Result : } 0110 &= 6
 \end{aligned}$$

\*یک صفر از سمت چپ اضافه میشود، معادل ضرب عدد در ۲ می باشد:

$$\begin{aligned}
 12 &<< 1 \\
 1100 &= 12 \\
 \text{Result : } 11000 &= 24
 \end{aligned}$$

- عملگر ( $\sim$ ) not

b	not
0	1
1	0

### ۳.۱) عملگرهای انتساب:

این عملگرها عبارتند از :

- تساوی (=)
- جمع تساوی (+=)
- تفریق مساوی (-=)
- ضرب مساوی (\*=)

### ۲) اولویت عملگرها در جاوا

اولویت ها به صورت زیر است :

• ++ --  
• / \* %  
•  
• - +

همه از چپ به راست

\* اگر بخواهیم خودمان اولویت تعیین کنیم از پرانتز استفاده می کنیم .

```

;Int num1=2
;Int num2=2
System.out.print ((++num1)+ 10*(++num1));      result = 43
System.out.print ((num1++) + (++num1));          result= 6

```

### مثال برای عملگرهای بیتی :

مثال ۱:

$$8 \& 12 = 8$$

$$(1000)_2 \& (1100)_2 = (1000)_2$$