

عملگرهای بیتی

[illegible]

راست برای پر کردن بیتهایی است که عدد از نوع int نیاز دارد. به این نکته توجه کنید که اعداد باینری از سمت راست به چپ خوانده می‌شوند. عملگرهای

عملگر	نام	دسته	مثال
&	بیتی AND	Binary	$x = y \ \& \ z;$

$x = y \wedge z;$	Binary	XOR	بیتی XOR	\wedge
$x = \neg y;$	Unary	NOT	بیتی NOT	\neg
$x \&= y;$	Binary	AND Assignment	بیتی AND Assignment	$=\&$
$x = y;$	Binary	OR Assignment	بیتی OR Assignment	$= $
$x \wedge= y;$	Binary	XOR Assignment	بیتی XOR Assignment	$=\wedge$

0	0	0
---	---	---

در زیر نحوه استفاده از عملگر بیتی AND آمده است :

```
int result = 5 & 3;
```

همانطور که در مثال بالا مشاهده می‌کنید نتیجه عملکرد عملگر AND بر روی دو مقدار 5 و 3 عدد یک می‌شود. اجازه بدهید ببینیم که چگونه این نتیجه را به دست می‌آید:

[illegible]

عملگر بیتی (OR)

اگر مقادیر دو طرف عملگر بیتی، OR هر دو صفر باشند نتیجه صفر در غیر اینصورت 1 خواهد شد. جدول درستی این عملگر در زیر آمده است:

X

1	0	1
1	1	0
0	0	0

نتیجه عملگر بیتی OR در صورتی صفر است که عملوندهای دو طرف آن صفر باشند. اگر فقط یکی از دو عملوند یک باشد نتیجه یک خواهد شد. به مثال زیر توجه کنید:

```
int result = 7 | 9;
System.out.println(result);
```

15

[illegible]

با استفاده از جدول درستی عملگر بیتی OR می‌توان نتیجه استفاده از این عملگر را تشخیص داد. عدد ۱۱۱۱ باینری معادل عدد ۱۵ صحیح است.

 $\log(A) = \frac{1}{2} \log(\det(A))$

X	Y	X XOR Y
1	1	0
1	0	1
0	1	1
0	0	0

در صورتیکه عملوندهای دو طرف این عملگر هر دو صفر یا هر دو یک باشند نتیجه صفر در غیر اینصورت نتیجه یک می‌شود. در مثال زیر تأثیر عملگر بیتی

XOR را بر روی دو مقدار مشاهده می‌کنید:

```
int result = 5 ^ 7;
```

در زیر معادل باینری اعداد بالا (5 و 7) نشان داده شده است.

با نگاه کردن به جدول درستی عملگر بیتی XOR ، می‌توان فهمید که چرا نتیجه عدد 2 می‌شود.

عملگر بیتی NOT(~)

این عملگر یک عملگر یگانگی

	NOT X	X
NOT X	0	1
X	1	0

عشکر بیانی ۱۹۵۱ معادیر بیتها را محسوس می‌کند. در زیر چگونگی استفاده از این عشکر آمده است:

```
int result = 0;
System.out.println(result);
```

به نمایش باینری مثال بالا که در زیر نشان داده شده است توجه نمایید.

عملگر بی‌تی تغییر مکان (shift)

قبول می‌کنند. عملوند سمت حب این

عملگر	نام	دسته	مثال

$\mu = \gamma = 2$	2.000000	$\frac{1}{\gamma} \frac{1}{\gamma} \text{---} \frac{1}{\gamma} \frac{1}{\gamma} \text{---} \frac{1}{\gamma} \frac{1}{\gamma} \text{---} \frac{1}{\gamma} \frac{1}{\gamma}$	0.000000

این عملگر، بیت‌های عمود سمت چپ را به تعداد n مکان مشخص شده، توسط عمود سمت راست، به سمت چپ منتقل می‌کند. به عنوان مثال:

در مثال بالا، ما بیت‌های مقدار 10 را دو مکان به سمت چپ منتقل کرده‌ایم، حال بیا ببیند تأثیر این انتقال را بررسی کنیم:

10:	00000000000000000000000000000000

40:	00000000000000000000000000000000

[illegible]

عملی تغییر مکان به سمت راست می‌شود.

این عملکرد شبیه به عملکرد تغییر مکان به سمت چپ است با این تفاوت که بیت‌ها را به سمت راست جا به جا می‌کند.

```
System.out.println(result);
```

مورد بررسی قرار دهیم :

6: 00000000000000000000000000000000110
