

تقدم عملگرها، مشخص می‌کند که در محاسباتی که بیش از دو عملوند دارند، ابتدا کدام عملگر اثرش را اعمال کند. عملگرها در جاوا در محاسبات دارای حق تقدم هستند. به عنوان مثال :

```
number = 1 + 2 * 3 / 1;
```

اگر ما حق تقدم عملگرها را رعایت نکنیم و عبارت بالا را از سمت چپ به راست انجام دهیم نتیجه 9 خواهد شد (1+2=3 سپس 3×3=9 و در آخر 1/9=9). اما کامپایلر با توجه به تقدم عملگرها محاسبات را انجام می‌دهد. برای مثال عمل ضرب و تقسیم نسبت به جمع و تفریق تقدم دارند. بنابراین در مثال فوق ابتدا عدد 2 ضربدر 3 و سپس نتیجه آنها تقسیم بر 1 می‌شود که نتیجه 6 به دست می‌آید. در آخر عدد 6 با 1 جمع می‌شود و عدد 7 حاصل می‌شود. در جدول زیر تقدم برخی از عملگرهای جاوا آمده است :

تقدم	عملگر
بالاترین	++, −, (used as prefixes); +, − (unary)
	*, /, %
	+, −
	<<, >>
	<, >, <=, >=
	==, !=
	&
	^
	&&
	=, *=, /=, %=, +=, -=
پایین‌ترین	++, − (used as suffixes)

ابتدا عملگرهای با بالاترین و سپس عملگرهای با پایین‌ترین حق تقدم در محاسبات تأثیر می‌گذارند. به این نکته توجه کنید که تقدم عملگرها ++ و − به مکان قرارگیری آنها بستگی دارد (در سمت چپ یا راست عملوند باشند). به عنوان مثال :

```
int number = 3;

number1 = 3 + ++number; //results to 7
number2 = 3 + number++; //results to 6
```

در عبارت اول ابتدا به مقدار number یک واحد اضافه شده و 4 می‌شود و سپس مقدار جدید با عدد 3 جمع می‌شود و در نهایت عدد 7 به دست می‌آید. در عبارت دوم مقدار عددی 3 به مقدار number اضافه می‌شود و عدد 6 به دست می‌آید. سپس این مقدار در متغیر number2 قرار می‌گیرد. و در نهایت مقدار number به 4 افزایش می‌یابد. برای ایجاد خوانایی در تقدم عملگرها و انجام محاسباتی که در آنها از عملگرهای زیادی استفاده می‌شود از پرانتز استفاده می‌کنیم :

```
number = ( 1 + 2 ) * ( 3 / 4 ) % ( 5 - ( 6 * 7 ) );
```

در مثال بالا ابتدا هر کدام از عباراتی که داخل پرانتز هستند مورد محاسبه قرار می‌گیرند. به نکته‌ای در مورد عبارتی که در داخل پرانتز سوم قرار دارد توجه کنید. در این عبارت ابتدا مقدار داخلی‌ترین پرانتز مورد محاسبه قرار می‌گیرد یعنی مقدار 6 ضربدر 7 شده و سپس از 5 کم می‌شود. اگر دو یا چند عملگر با حق تقدم یکسان موجود باشد ابتدا باید هر کدام از عملگرها را که در ابتدای عبارت می‌آیند مورد ارزیابی قرار دهید. به عنوان مثال :

```
number = 3 * 2 + 8 / 4;
```

هر دو عملگر * و / دارای حق تقدم یکسانی هستند. بنابر این شما باید از چپ به راست آنها را در محاسبات تأثیر دهید. یعنی ابتدا 3 را ضربدر 2 می‌کنید و سپس عدد 8 را بر 4 تقسیم می‌کنید. در نهایت نتیجه دو عبارت را جمع کرده و در متغیر number قرار می‌دهید.