

## محاسبات ریاضی در ++C

- در اغلب برنامه‌ها، عملیات‌های محاسباتی انجام می‌شود.
- به عنوان مثال برنامه‌ای که میانگین سه عدد را محاسبه می‌کند، ب.م.م. دو عدد را بدست می‌آورد و ... نیاز به اجرای عمل‌های محاسباتی (جمع، ضرب و ...) دارد.
- عملگرهای محاسباتی (ریاضی) در ++C

مثال	عمل	عملگر
$x + y$	جمع	+
$x - y$	تفریق	-
$x * y$	ضرب	*
$x / y$	تقسیم	/
$x \% y$	باقیمانده	%

عملوند سمت راست (عملگر) عملگر سمت چپ

- اگر هر دو عملوند یک عملگر ریاضی از یک نوع باشند، نتیجه نیز از همان نوع است



## محاسبات ریاضی در ++c

- اگر هر دو عملوند عملگر تقسیم از نوع صحیح (int) باشند، نتیجه نیز یک عدد صحیح است. پس

$$17 / 5 \rightarrow 3, \quad 7 / 4 \rightarrow 1$$

- اما

$$17.0 / 4 \rightarrow 4.25$$

- توجه کنید که قسمت اعشاری نتیجه‌ی تقسیم دو عدد صحیح بریده می‌شود (تغییر نوع داده می‌شود) و عدد گرد نمی‌شود.

- هر دو عملوند عملگر باقیمانده‌ی تقسیم باید حتماً عدد صحیح (از نوع int) باشند

$$7 \% 4 \rightarrow 3, \quad 17 \% 5 \rightarrow 2$$

- یک کاربرد عملگر باقیمانده تقسیم صحیح (%):

– عدد صحیح  $x$  زوج است اگر  $x \% 2$  صفر باشد و در غیر این صورت فرد است.

– عدد صحیح  $x$  مضربی از ۷ است (بر ۷ بخش پذیر است) اگر  $x \% 7$  صفر باشد و در غیر این صورت  $x$  بر ۷ بخش پذیر نیست.



## محاسبات ریاضی در ++c

- در ++C برای توان هیچ نماد یا عملگری در نظر گرفته نشده است.
- بنابراین برای محاسبه‌ی عبارت جبری

$$ax^2 + bx + c$$

- باید نوشت

$$a * x * x + b * x + c$$

- علاوه بر محاسبات ساده حسابی، در بسیاری از برنامه‌ها ممکن است نیاز به تابع‌های مثلثاتی، لگاریتمی، نمایی و ... باشد.
- سرفایل `<cmath>` شامل مجموعه‌ای از تابع‌های ریاضی مفید و پرکاربرد است که برنامه‌نویس را قادر می‌سازد محاسبات معمولی ریاضی را انجام دهد.
- برای استفاده از تابع‌های موجود در این سرفایل، باید دستور پیش‌پردازنده‌ی زیر در ابتدای برنامه قرار گیرد

```
#include <cmath>
```



# محاسبات ریاضی در ++c

مثال	شرح	تابع
$\text{ceil}(9.2) \rightarrow 10.0$ $\text{ceil}(-9.8) \rightarrow -9.0$	کوچکترین عدد صحیح بزرگتر یا مساوی $x$	$\text{ceil}(x)$
$\text{floor}(9.2) \rightarrow 9.0$ $\text{floor}(-9.8) \rightarrow -10.0$	بزرگترین عدد صحیح کوچکتر از $x$	$\text{floor}(x)$
$\text{fabs}(-8.76) \rightarrow 8.76$	قدر مطلق $x$	$\text{fabs}(x)$
$\text{exp}(1.0) \rightarrow 2.71828$	عدد نپر به توان $x$ (تابع نمایی)	$\text{exp}(x)$
$\text{log}(2.71828) \rightarrow 1.0$	لگاریتم طبیعی $x$ (در مبنای عدد نپر)	$\text{log}(x)$
$\text{log10}(100.0) \rightarrow 2.0$	لگاریتم در پایه ی ده $x$	$\text{log10}(x)$
$\text{pow}(9, 0.5) \rightarrow 3$ $\text{pow}(2, 7) \rightarrow 128$	$x$ به توان $y$	$\text{pow}(x, y)$
$\text{sqrt}(9.0) \rightarrow 3.0$	ریشه ی دوم $x$ ( $x$ باید نامنفی باشد)	$\text{sqrt}(x)$
$\text{sin}(0.0) \rightarrow 0.0$	سینوس $x$	$\text{sin}(x)$
$\text{cos}(0.0) \rightarrow 1.0$	کسینوس $x$	$\text{cos}(x)$
$\text{tan}(0.0) \rightarrow 0.0$	تانژانت $x$	$\text{tan}(x)$
$\text{fmod}(2.6, 1.2) \rightarrow 0.2$	باقیمانده ی تقسیم $x$ و $y$ به صورت اعشاری	$\text{fmod}(x, y)$



# پرانتر

- استفاده از پرانتز در عبارتهای C++ تقریباً مشابه استفاده از پرانتز در عبارتهای جبری است.

$$a * (b + c) \rightarrow a * b + a * c$$

$$x * (y - (z - 3) / t) \rightarrow$$

$$x * y - z * z / t - 3 * x / t$$

- می توان از پرانتزهای زائد برای روشنی بیشتر عبارت مد نظر استفاده کرد.

$$y = (a * x * x) + (b * x) + c$$

- هر پرانتز باز شده حتما باید بسته شود در غیر این صورت خطای زمان کامپایل رخ خواهد داد.

- بنابراین تعداد پرانتزهای باز شده و بسته شده در یک دستور باید برابر باشند.



# اولویت (تقدم) عملگرها

- اولویت عملگرها در C++ تقریباً مشابه قوانین جبر است:

عملگر(ها)	عمل(ها)	ترتیب ارزیابی (اولویت) عملگر(ها)
( )	پرانتز	در مرحله‌ی نخست ارزیابی می‌شود. اگر چند پرانتز تو در تو باشند، داخلی‌ترین پرانتز پیش از همه ارزیابی می‌شود. اگر در یک سطح چند پرانتز باشند (تو در تو نباشند)، پرانتزها از چپ به راست ارزیابی می‌شوند.
* / %	ضرب تقسیم باقیمانده	در مرحله‌ی دوم ارزیابی می‌شوند. اگر در یک سطح چند تا از این عملگرها باشد، از چپ به راست ارزیابی می‌شوند.
+ -	جمع تفریق	در مرحله‌ی آخر ارزیابی می‌شوند. اگر در یک سطح چند تا از این عملگرها باشد، از چپ به راست ارزیابی می‌شوند.

- اولویت تقدم عملگرها در عبارت زیر:

Algebra:  $z = pr \% q + w/x - y$

C++:  $z = p * r \% q + w / x - y;$

6

1

2

4

3

5



# اولویت عملگرها (تقدم) عملگرها

- اولویت عملگرها را در عبارت  $y = 2 * 5 * 5 + 3 * 5 + 7;$  تعیین کنید.

Step 1.  $y = 2 * 5 * 5 + 3 * 5 + 7;$  (Leftmost multiplication)

$2 * 5$  is 10

Step 2.  $y = 10 * 5 + 3 * 5 + 7;$  (Leftmost multiplication)

$10 * 5$  is 50

Step 3.  $y = 50 + 3 * 5 + 7;$  (Multiplication before addition)

$3 * 5$  is 15

Step 4.  $y = 50 + 15 + 7;$  (Leftmost addition)

$50 + 15$  is 65

Step 5.  $y = 65 + 7;$  (Last addition)

$65 + 7$  is 72

Step 6.  $y = 72$  (Last operation—place 72 in y)



## تمرین

- ترتیب اولویت عملگرها را در هریک از عبارتهای زیر مشخص کنید.

$$x = 7 + 3 * 6 / 2 - 1;$$

$$x = 2 \% 2 + 2 * 2 - 2 / 2;$$

$$x = ( 3 * 9 * ( 3 + ( 9 * 3 / ( 3 ) ) ) ) );$$

- برنامه‌ای بنویسید که دو عدد صحیح را از خروجی دریافت و مشخص کند که آیا عدد اول بر عدد دوم بخش پذیر است یا نه.
- برنامه‌ای بنویسید که شعاع یک دایره را از ورودی بخواند، قطر، محیط و مساحت دایره را در خروجی چاپ کند. از تقریب  $\pi \cong 3.14159$  استفاده کنید.
- برنامه‌ای بنویسید که با دریافت یک عدد اعشاری مقدار تابع‌های مثلثاتی سینوس، کسینوس و تانژانت را به ازای آن محاسبه کند (مقدار عدد به صورت رادیان در نظر گرفته می‌شود).

