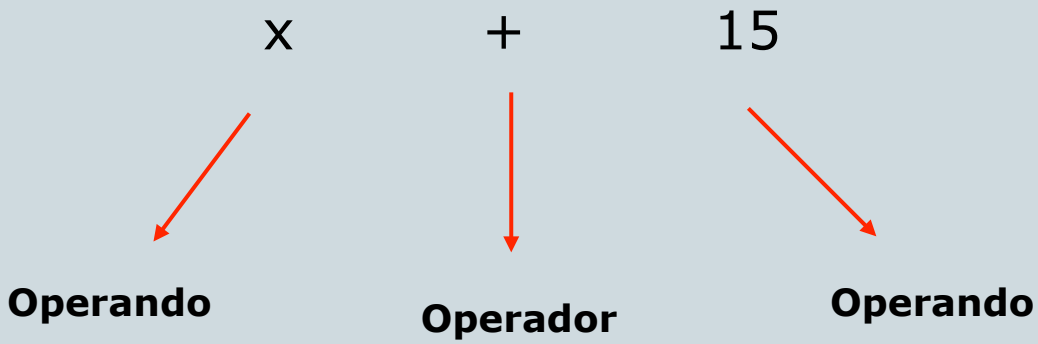
Algoritmos I

OPERADORES

Operadores e Expressões

- Uma expressão é composta por um ou mais operandos, que se combinam entre si mediante operadores, produzindo um resultado.

- Exemplo:



Expressões

- Nas linguagens é possível formar expressões utilizando variáveis constantes e operadores matemáticos: + (adição), - (subtração), / (divisão) e % (resto de divisão inteira, ou mod). Essas expressões são utilizadas onde é preciso utilizar um valor, que será o que resultará da expressão. Exemplo:
  - X + 30 //devolve x + 30.
  - X == 55 // verifica se são iguais e retorna 1 ou 0.
  - X = 10 //atribuição: retorna o valor atribuído.
- As expressões não são iguais a sentenças. As sentenças indicam ao compilador que realize alguma tarefa (e terminam com ;), enquanto as expressões indicam um cálculo. Uma sentença pode conter várias expressões.

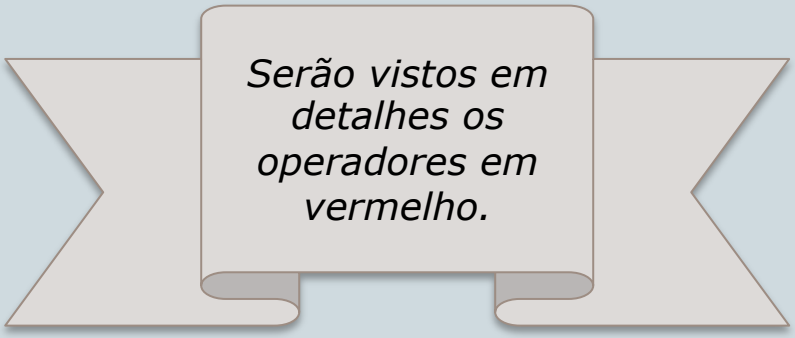
Operadores

- C/C++ são linguagens ricas em operadores. A classificação é feita em função do número de operadores sobre os quais atuam e a quantidade de operações que realizam.
- Os operadores se classificam em: *unitários* (atuam sobre 1 operando), *binários* (atuam sobre 2 operandos) ou *ternários* (atuam sobre 3 operandos). Também é chamado de aridade.
- Também existe a classificação conforme a posição do operador e dos operandos: *prefixo* (vai antes), *infixo* (vai no interior) e *sufixo* (vai atrás).
- Ao haver vários operandos existem as propriedades de precedência, associatividade e ordem de avaliação.

Operadores

- *Precedência (prioridade)*: indica a prioridade do operador mediante outros no cálculo de uma expressão.
- *Associatividade*: determina a ordem que operandos do mesmo tipo se associam na ausência de parênteses. Podem ser pela direita (D-E) – operadores binários, ternários e de atribuição, ou pela esquerda (E-D) – operadores binários. Em alguns operadores a associatividade não faz sentido, como no caso do operador sizeof().

- Segue abaixo uma classificação completa dos operadores:
  - Operadores de resolução de escopo.
  - **Operadores Aritméticos**.
  - Operadores de incremento e decremento.
  - Operadores de atribuição.
  - Operadores de atribuição composta.
  - **Operadores Relacionais**.
  - **Operadores Lógicos**.
  - Operadores de bits.
  - Operadores condicionais.
  - Operadores de endereço ou indireção.
  - Operadores de tamanho (sizeof).
  - Operadores de sequência ou de avaliação (vírgula).
  - Operadores de conversão.
  - Operadores de molde (“cast”).
  - Operadores de construção de tipo.
  - Operadores de memória dinâmica.



Prioridade de Operadores

- A tabela a seguir mostra a prioridade entre os operadores. O grupo 1, por exemplo, possui maior prioridade que o grupo 2. Seguem outras regras:
  - Se dois operandores são aplicados no mesmo operando, o operador com maior prioridade é aplicado primeiro.
  - Todos os operadores do mesmo grupo possuem prioridade de associatividade iguais.
  - Se dois operadores possuem prioridade igual, primeiro se aplica o operador com prioridade mais alta.
  - Os parênteses possuem prioridade máxima, e alteram qualquer ordem.

Prioridade de Operadores

Prioridade	Operadores	Associatividade
1	:: (Resolução de Escopo) * -> (Direções) [] (Indexação Vetorial) () (Chamada de Funções)	E-D
2	++ -- (Incremento e Decremento) ~ (Manipulação de bits) ! (Lógico) - + (Aritméticos – negativo ou positivo) & * (Endereço) sizeof (tamanho de bytes)	D-E
3	.* ->*	E-D
4	* / % (Aritméticos)	E-D
5	+ - (Aritméticos)	E-D
6	<< >> (Deslocamento de bits)	E-D
7	< <= > >= (Relacionais)	E-D
8	== != (Relacionais)	E-D

Prioridade de Operadores

Prioridade	Operadores	Associatividade
9	& (Manipulação de bits)	E-D
10	^ (Manipulação de bits)	E-D
11	|| (Manipulação de bits)	E-D
12	&& ou and (Lógico)	E-D
13	|| ou or (Lógico)	E-D
14	?: (Expressão Condicional)	D-E
15	= *= /= %= += -= <<= >>= &= ||= ^= (Atribuição)	D-E
16	, (Vírgula)	E-D

Operadores Aritméticos

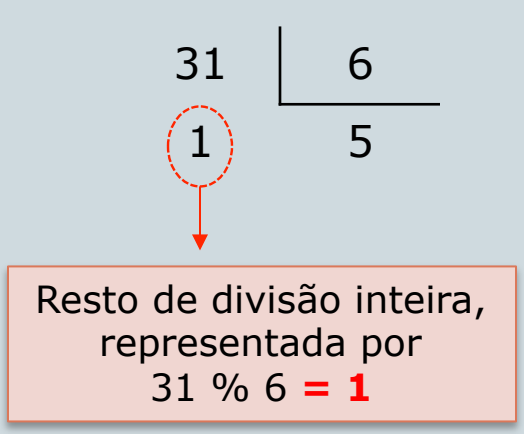
- Realizam operações aritméticas básicas.
- Prioridade de avaliação:
  - 1. Parênteses (mudam a ordem de prioridade).

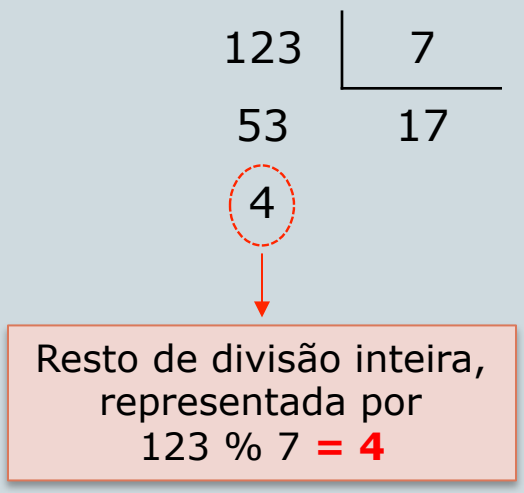
Operador	Operação	Prioridade
+, -	+25, - 6.475	2
*, /, %	5*5 é 25	3
	25/5 é 5	
	25%6 é 1	
+, -	2+3 é 5	4
	2-3 é -1	

- Quando de mesma prioridade, a associatividade é da esquerda para a direita.

Operadores Aritméticos

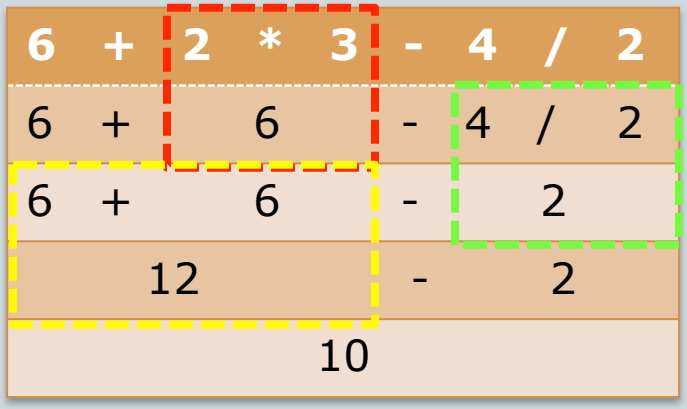
- O que faz o operador %?
  - Este não é um operador de porcentagem, como na calculadora. É um operador de resto de divisão inteira. Exemplo:

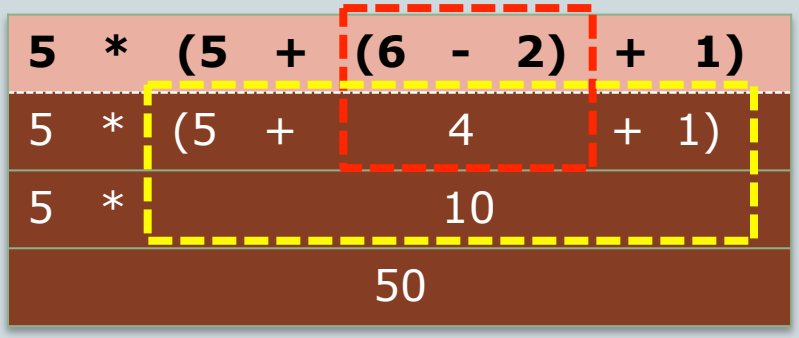




Operadores Aritméticos

- Veja as expressões abaixo, resolvidas de acordo com a prioridade.





- Sempre que abrir um parênteses feche-o, pois senão será gerado um erro de compilação. E cuide, pois a utilização dos parênteses de forma incorreta modificam todo o valor da expressão.

Operadores Relacionais

- Utilizados em comparações, gerando um resultado verdadeiro ou falso.
- Todos possuem a mesma prioridade.

Operador	Significado	Exemplo
==	Igual a	a == b
!=	Diferente de	a != b
>	Maior que	a > b
<	Menor que	a < b
>=	Maior ou igual que	a >= b
<=	Menor ou igual a que	a <= b

- A associatividade é da esquerda para a direita.

Operadores Lógicos

- Se utilizam de expressões para retornar valores verdadeiros (qualquer número inteiro diferente de 0) ou falso (0).
- São utilizados para unir condições, usadas em desvios condicionais e laços de repetição.

Operador	Operação Lógica	Exemplo	Prioridade
! (Negação)	! lógica	!(x >= y)	1
&& / and (e)	op1 && op2	m < n && i > j	2
|| / or (ou)	op1 || op2	m == 5 || n != 10	3

Operadores Lógicos

- O operador ! inverte o valor lógico do operando.

Operando a	!a (not a)
true (1)	false (0)
false (0)	true (1)

Operadores Lógicos

- Ao utilizar o operador && (and), ambos os operandos precisam ser verdadeiros para o resultado ser verdadeiro. Caso algum dos operandos seja falso, o resultado será falso.

Operandos		a && b (a and b)
a	b	
true (1)	true (1)	true (1)
true (1)	false (0)	false (0)
false (0)	true (1)	false (0)
false (0)	false (0)	false (0)

Operadores Lógicos

- Ao utilizar o operador || (or), se qualquer um dos operandos for verdadeiro a sentença será verdadeira. Só será falsa quando os dois operandos forem falsos.

Operandos		a || b (a or b)
a	b	
true (1)	true (1)	true (1)
true (1)	false (0)	true (1)
false (0)	true (1)	true (1)
false (0)	false (0)	false (0)

Relembrando

- Prioridade entre os operadores :
  - 1 – Parênteses.
  - 2 – Operadores Aritméticos.
  - 3 – Operadores Relacionais.
  - 4 – Operadores Lógicos.
- Condições como 1 < x < 10 não existirão mais. Será necessário utilizar os operadores lógicos para interligar as mesmas. Ficariam (1 < x) and (x < 10).
- Quando as duas condições precisam ser verdadeiras para executar o código, utiliza-se and (&&). Caso uma ou outra possam ser verdadeiras para executá-lo, usa-se or (||).