

# C# Essencial



Exercícios - Fase 2

Fundamentos

## Exercícios fase 2

1- Declare as variáveis **nome**, **idade** e **nota** atribuindo os valores “**Paulo**”, **17** e **7.5** e exiba a saída no formato : **Aluno** <**nome**> **tem** <**idade**> **anos** e **nota** <**nota**> usando a *concatenação e a interpolação de strings*.

2- Para o exercício anterior exiba **o nome a idade e a nota** em linhas separadas usando as sequências de escapes.

## Exercícios fase 2

3- Para <sup>qual</sup> qual tipo de dados você pode converter um **float** *implicitamente* ?

- ☐ int
- ☒ double
- ☐ long
- ☒ decimal

4- Em <sup>qual</sup> qual conversão numérica você precisaria realizar o casting (*convesão forçada*) ?

- ☐ int para long
- ☒ double para long
- ☒ double para float
- ☒ decimal para float
- ☒ long para int
- ☒ double para decimal

## Exercícios fase 2

5- Escreva um programa que recebe **3 letras via teclado** e as exiba na **ordem reversa** usando *a concatenação e também a interpolação de strings*

## Exercícios fase 2

6- Marque verdadeiro(V) ou falso(F) para os códigos abaixo:

- ( F ) long resultado = 1.32;
- ( V ) var nome = "Maria";
- ( V ) string resultado = 100.ToString();
- ( V ) A sequência de escape `\n` inclui uma nova linha
- ( F ) float f = 5.45;
- ( V ) decimal valor = (decimal) 10.99f;
- ( F ) var status = null;
- ( V ) object o = 12.45m;
- ( V ) string titulo = true.ToString();
- ( F ) A sequência `\t` inclui uma tabulação vertical

## Exercícios fase 2

7- Escreva um programa para receber **dois valores** via teclado do tipo **double** e a seguir realize as operações *de soma, subtração, multiplicação, exponenciação, divisão e módulo* exibindo o resultado:

## Exercícios fase 2

8 - Faça um programa para calcular o resultado da **fórmula de baskara** dados os valores **de a b e c**.

$$x = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

Valores : **a = 1 , b = 12 e c = -13**

Fórmula do delta

$$\text{delta} \Rightarrow \Delta = b^2 - 4ac$$

$$x_1 = \frac{-b + \sqrt{\Delta}}{2.a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2.a}$$

## Exercícios fase 2

9- Escreva um programa que receba *um nome e uma senha via teclado*. *Nome é uma string e Senha é um inteiro*. Se o nome for igual a **'admin'** ou **'maria'** e a senha for igual a **'123'** então exiba a mensagem *'Login feito com sucesso'* caso contrário exiba a mensagem *'Login inválido'*: **(use o operador condicional ternário)**



## Exercícios fase 2

10- Indique verdadeiro(V) ou falso (F) para as seguintes declarações de variáveis considerando a nomenclatura usada:

- ( V ) **string? nome;** é um exemplo de *nullable reference type*;
- ( F ) para x igual a 0 , **Console.WriteLine(x++);** imprime o valor 1
- ( F ) A ordem de precedência dos operadores lógicos é : **!, || e && (NOT, OR e AND)**
- ( V ) para y igual a 0 , **Console.WriteLine(++y);** imprime o valor 1
- ( F ) **(10 % 2 == 0) ? "Par" : "Impar";** Vai retornar **"Impar"**
- ( F ) Para **x=25 e y=5** , a expressão ( <sup>Falso</sup> **y >= x** ) && ( <sup>Verdadeiro</sup> **y <= x** ); retorna **true**;
- ( V ) Os **nullable reference types** emitem um alerta em tempo de compilação para uso do **null**
- ( F ) Para **x = 10** , o código **Console.WriteLine(x+=x);** imprime o valor 10;
- ( F ) Para **x= 5** , a expressão <sup>Falso</sup> **!(9 >= x)** && <sup>Verdadeiro</sup> **x <= 7** || <sup>Falso</sup> **x == 5**); retorna **false**;
- ( V ) O operador **?.** permite verificar se um objeto é **null** e evitar o **NullReferenceException**
- ( V ) Qualquer tipo de dado pode ser declarado como **anulável** com a ajuda do operador **"?"**

## Exercícios fase 2

11- Escreva um programa que recebe via teclado **dois números inteiros x e y** e imprima no console se x é par ou não e se y é par ou não. Use o ***operador condicional ternário (? :)***

## Exercícios fase 2

12- Crie um programa que receba um numero inteiro x via teclado e calcule e imprima no console o resultado das seguintes operações : **( $x^2 \rightarrow x$  ao quadrado)** (**pi = 3.1415**)

$$-6 + x * 5$$

$$(13-2) * x$$

$$(x + -2) * (20 / x)$$

$$(12 + x) / (x - 4)$$

$$3 x^2 + x + 10$$

$$\text{pi} * x^2$$


## Exercícios fase 2

13- Considere o seguinte trecho de código:

```
int y = 5 ;
```

```
y = (y++) + y + (++y);      ou      y = y++ + y + ++y;
```

```
Console.WriteLine(y);
```

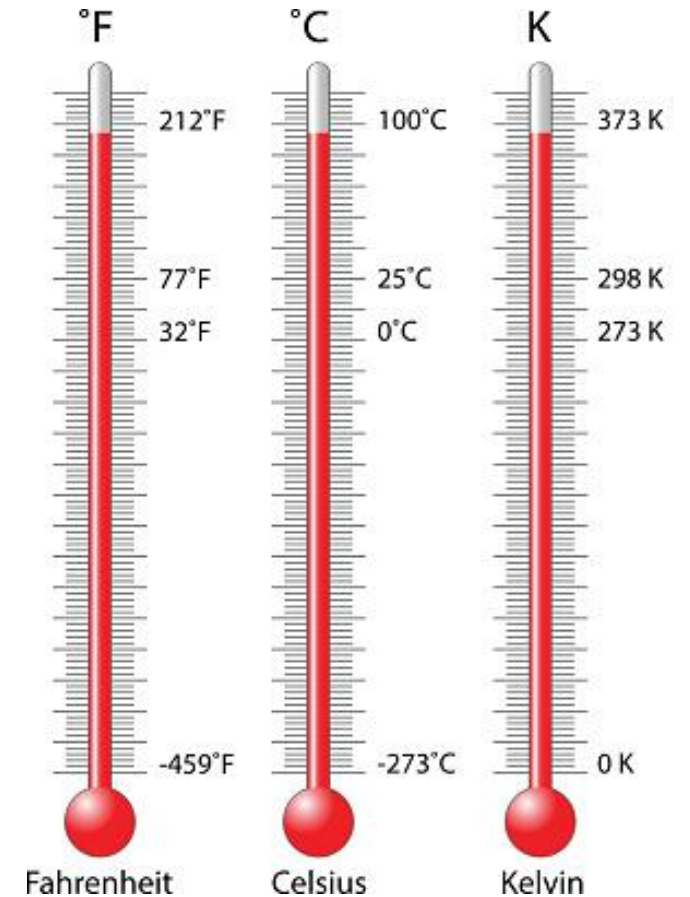
 y++ : 5 e então incrementará para 6 ;  
++y : vai incrementar o para 7 :

Logo:  $y = (5 + 6 + 7) = 18$

## Exercícios fase 2

14- Escreva um programa que solicite a temperatura em graus **Celsius** e converta para **Kelvin** e **Farhenheit** usando as fórmulas a seguir:

- Converter para Kelvin =>  **$K = C + 273$**  ;
- Converter para Farhenheit =>  **$F = (C * 9) / 5 + 32$**  ;



# Exercícios fase 2

15 - Escolha a opção que representa a exibição do resultado para o código usando os *operadores de decremento e incremento (pré e pós)* :

```
var numero = 5;
Console.WriteLine(numero++);
numero = 1;
Console.WriteLine(++numero);
numero = 2;
Console.WriteLine(numero--);
numero = 3;
Console.WriteLine(--numero);

Console.ReadKey();
```

resolva primeiro a expressão  
resolva primeiro o decremento

( + )	( )	( )	( )
5	6	5	6
2	3	2	2
2	2	1	1
2	2	2	3